Probability in the Engineering and Informational Sciences, **27**, 2013, 1–23.
doi:10.1017/S0269964812000277

SINGLE VEHICLE ROUTING PROBLEMS WITH A PREDEFINED CUSTOMER ORDER, UNIFIED LOAD AND STOCHASTIC DISCRETE DEMANDS

Dimitrios G. Pandelis
Department of Mechanical Engineering
University of Thessaly
Volos, Greece
E-mail: d_pandelis@mie.uth.gr

Constantinos C. Karamatsoukis
Department of Financial and Management Engineering
University of the Aegean
Chios, Greece
E-mail: k.karamatsoukis@fme.aegean.gr

Epaminondas G. Kyriakidis
Department of Statistics
Athens University of Economics and Business
Athens, Greece
E-mail: ekyriak@aueb.gr

We consider the problem of finding the optimal routing of a single vehicle that delivers K different products to N customers that are served according to a particular order. It is assumed that the demands of the customers for each product are discrete random variables, and the total demand of each customer for all products cannot exceed the vehicle capacity. The joint probability mass function of the demands of each customer is known. It is assumed that all products are stored together in the vehicle's single compartment. The policy that serves all customers with the minimum total expected cost is found by implementing a suitable dynamic programming algorithm. We prove that this policy has a specific threshold-type structure. Furthermore, we study a corresponding infinite-time horizon problem in which the service of the customers is not completed when the last customer has been serviced but it continues periodically with the same customer order. The demands of each customer for the products have the same distributions at different periods. The discounted cost optimal policy and the average-cost optimal policy have the same structure as the optimal policy in the finite-horizon problem. Numerical results are given that illustrate the structural results.

1. INTRODUCTION

The problems concerning the optimal distribution of goods between depot and users (customers) are generally known as vehicle routing problems (VRPs). The VRP is a

 0269-9648/13 $25.00

combinatorial optimization problem and in many cases it can be solved using integer programming algorithms. It is a very important problem in the fields of transportation, distribution, and logistics. It can be considered as a generalization of the classical traveling salesman problem that involves finding an optimal route for visiting n cities and returning to the point of origin. Dantzig and Ramser [4] introduced a first version of the VRP and proposed a mathematical programming formulation and algorithmic approach for its solution. In that paper the authors described, as a real-world application, the problem concerning the delivery of gasoline to gas stations. Clarke and Wright [2] improved the Dantzig–Ramser approach by developing an effective greedy heuristic. Following these two seminal papers, during the last 40 years a great number of mathematical models were introduced for various versions of the VRP, and many exact and heuristic algorithms were implemented for their optimal and approximate solution. Surveys covering the most important results on the VRPs were presented by Laporte [6,7], Toth and Vigo [17], Simchi-Levi, Chen, and Bramel [13], Cordeau et al. [3], and Liong et al. [8].

The VRP refers to a fleet of vehicles that originate from one or several depots and deliver or collect products from N geographically scattered customers. Each vehicle starts its route from a depot, visits a subset of customers, delivers new products or collects expired products from each customer, and finally returns to the depot. If the demand of a customer for new products exceeds the amount of new products carried by the vehicle or the amount of expired products of a customer exceeds the empty space of the vehicle, the vehicle must interrupt its route and go to the depot for restocking new products or unloading expired products. The cost structure includes the travel costs from one customer to another and the travel costs from a customer back to the depot for replenishment or unloading. The objective is usually to minimize the total travel cost for the service of all customers. It is possible to consider other optimization criteria as the minimization of the number of vehicles (or drivers) required to serve all customers and the minimization of the penalties associated with partial service of the customers. In addition, in many cases it is necessary to consider stochastic versions of the VRP, that is, problems for which, a priori, there is only partial knowledge of the number of the customers or the demands of the customers or the total costs associated with the arcs of the road network. Two interesting variants of the VRP that have been studied extensively in the literature are (i) the VRP with time windows in which the delivering locations have time windows within which the deliveries or the collections must be made, and (ii) the capacitated VRP (with or without time windows) in which the vehicles have limited carrying capacity. Typical applications of the VRP are the delivery of goods to supermarkets, the solid waste collection, cash collection from bank branches, street cleaning, school bus routing, dial-a-ride systems, transportation of handicapped persons, routing of salespeople, and routing of maintenance units. The VRPs are NP-hard problems and, therefore, many heuristics and metaheuristics (tabu search, simulated annealing, genetic algorithms, and colony optimization) have been proposed that search for good solutions. In addition, exact algorithms such as branch-and-bound, branch-and-cut, and branch-and-cut-and-price methods that find the global minimum for the cost function have been proposed.

In the present paper, we consider a simple and interesting capacitated VRP that was introduced by Tatarakis and Minis [15]. In this problem it is assumed that a single vehicle starts its route from a depot and delivers K different products to N customers according to a predefined customer sequence $1 \to 2 \to \cdots \to N$. The vehicle may carry any quantity of product $i \in \{1, \ldots, K\}$ provided that the total capacity of the vehicle is not exceeded. All product quantities are calculated using the same unit of measure, for example, m^3 or kg. The demands of the customers for each product are discrete random variables and the total demand for all products of each customer cannot exceed the vehicle capacity. The vehicle is allowed to return to the depot for stock replenishment. It is assumed that the travel costs between any two consecutive customers and between each customer and the depot are known. The objective is to find the policy that minimizes the expected total cost for the service of all customers. Tatarakis and Minis [15] selected as decision epochs for this problem the epochs at which the service of each customer has been completed. They presented a dynamic programming formulation for this problem for the case in which $K = 2$ and they state the following structural result that is proved in Chapter 6 of Tatarakis's [14] thesis: If z_i, $i = 1, 2$, is the load of product i carried by the vehicle after the service of customer $j \in \{1, \ldots, N-1\}$ has been completed, there exists a critical number $s_j(z_1) \geq 0$ such that the optimal decision is to proceed to customer $j+1$ if $z_2 > s_j(z_1)$. If $z_2 \leq s_j(z_1)$, the optimal decision is to return to the depot to load $\theta \in \{0, 1, \ldots, Q\}$ units of product 1 and $Q - \theta$ units of product 2, and then go to customer $j+1$. Note that, if the second decision is chosen, it is possible that a second return to the depot is needed if the demand of customer $j+1$ for product 1 or 2 is greater than θ or $Q - \theta$, respectively. In the present paper, we choose as decision epochs of the problem the epochs at which the vehicle visits for the first time each customer and has satisfied as much of the customer's demand as possible. This enables us (i) to prove, in a much simpler way, for each customer $j \in \{1, \ldots, N-1\}$ a similar threshold-type structural result as the one presented in Tatarakis and Minis [15] for any positive integer K and (ii) to prove for each customer $j \in \{1, \ldots, N-1\}$ that it is optimal for the vehicle to return twice to the depot if a very simple condition holds. We also study a corresponding infinite time-horizon problem in which the service of the customers does not stop when the demands of the last customer N are satisfied but it continues periodically with the same customer order. It is assumed that at different periods the demands of customers for each product are identically distributed. Using well-known results of Markov decision theory we prove that the policy that minimizes the total expected discounted cost and the policy that minimizes the long-run expected average cost per unit time have the same threshold-type structure as the optimal policy in the initial finite-horizon problem.

As mentioned by Tatarakis and Minis [15], a practical application of the considered problem could be the so called ex-van sales. In ex-van sales the driver of the vehicle acts as a salesman. He visits his customers (retail outlets, supermarkets, kiosks, etc.) typically according to a predefined sequence. The demands of each customer for the products are not known in advance but they are revealed upon arrival. If a customer's demand for a product exceeds the quantity that is loaded in the vehicle, the driver has to go to the depot for replenishment. Another example could be the routing of a self-propelled vehicle in a manufacturing shop that transfers discrete parts to workcenters in a predefined sequence (see Rembold, Blume, and Dillmann [10]). Note that in addition to the main pathway connecting the workcenters, there are spurs connecting each workcenter with the material warehouse, allowing the return and the reloading of the vehicle. The demand for discrete parts of each workcenter may be stochastic due to failures.

Note that the problem studied by Tatarakis and Minis [15] is a generalization of the problem introduced by Yang, Mathur, and Ballou [19] in which it is assumed that the vehicle delivers to the customers only one product, that is, $K = 1$. They proved that for each customer $j \in \{1, \ldots, N-1\}$ there exists a critical number s_j such that the optimal decision, after serving customer j, is to continue to customer $j+1$ if the remaining quantity in the vehicle is greater than or equal to s_j, or return to the depot for replenishment if it is less than s_j. Kyriakidis and Dimitrakos [5] proved an analogous result if the demands of the customers are continuous random variables and developed a suitable dynamic programming algorithm for the determination of the critical numbers s_j, $j = 1, \ldots, N-1$. Tatarakis and Minis [15] also studied the case of multiple product deliveries when the demand of each customer for product $i \in \{1, \ldots, K\}$ is a discrete random variable and each product type is stored in its dedicated compartment in the vehicle. They proved that the optimal policy has a specific threshold-type structure for $K = 2$. Pandelis, Kyriakidis, and Dimitrakos [9] proved the structure of the optimal policy for this problem for any positive K. Tsirimbas et al. [18] studied the same problem when the demand of each customer for product $i \in \{1, \ldots, K\}$ is not a random variable but a constant number. They also assumed that the vehicle visits each customer only once and they designed for three versions of the problem (compartmentalized load, unified load, and pickup and delivery) suitable dynamic programming algorithms for the determination of the optimal policy. Finally, Secomandi and Margot [12] developed a dynamic programming algorithm for the problem in which a single vehicle delivers one product to N customers, the demands of the customers are discrete random variables and the customers are not necessarily serviced according to a particular sequence.

The rest of the paper is organized as follows. In the next section we present the finite-horizon unified load problem, we define the decision epochs and we give the corresponding dynamic programming equations. We show that the optimal policy has a specific threshold-type structure. In Section 3, the infinite-horizon problem is presented and it is shown that the discounted cost optimal policy and the average-cost optimal policy have the same structure as the finite-horizon optimal policy. The theoretical results of Sections 2 and 3 are illustrated by numerical results. The conclusions of the paper are summarized in the last section.

2. THE FINITE-HORIZON PROBLEM

We consider a set of nodes $V = \{0, 1, \ldots, N\}$ with node 0 denoting the depot and nodes $1, \ldots, N$ corresponding to customers. There are K different products to be delivered to the customers. The items of all products are of the same size. It is assumed that the depot contains enough items of all products to satisfy the demands of all customers. The customers are serviced in the order $1, 2, \ldots, N$ by a vehicle, which may carry any quantity of product $i \in \{1, \ldots, K\}$ provided that its total capacity Q is not exceeded. The vehicle starts its route from the depot with a total load of Q products of all kinds and after servicing all customers it returns to the depot. The road network is depicted in Figure 1.

The demand of customer j, $j = 1, \ldots, N$, for product i, $i = 1, \ldots, K$, is a discrete random variable ξ_i^j. We assume that the joint probability distribution of each customer's demands is known. We denote by $c_{j,j+1}$, $j = 1, 2, \ldots, N-1$, the travel cost between customers j and $j+1$, and by c_{j0}, c_{0j}, $j = 1, 2, \ldots, N$, the travel cost between customer j and the depot and the cost between the depot and customer j, respectively. These costs can be considered as the costs of the gasoline that the vehicle needs to cover the distances between customers or the distances between customers and the depot. We naturally assume that these costs are symmetric and satisfy the triangle inequality, that is,

$$c_{i0} = c_{0i}, \quad i = 1, \ldots, N,$$

and

$$c_{i,i+1} \le c_{i0} + c_{0,i+1}, \quad i = 1, \ldots, N-1.$$

The actual demands of each customer become known upon the vehicle's arrival at the customer's site. We assume that the total demand of each customer cannot exceed the vehicle's capacity, that is, $\max_{j=1,2,\ldots,N} \sum_{i=1}^{K} \xi_i^j \le Q$. When the vehicle visits customer j for the first time it satisfies as much demand as possible. If part of the demand is not satisfied,

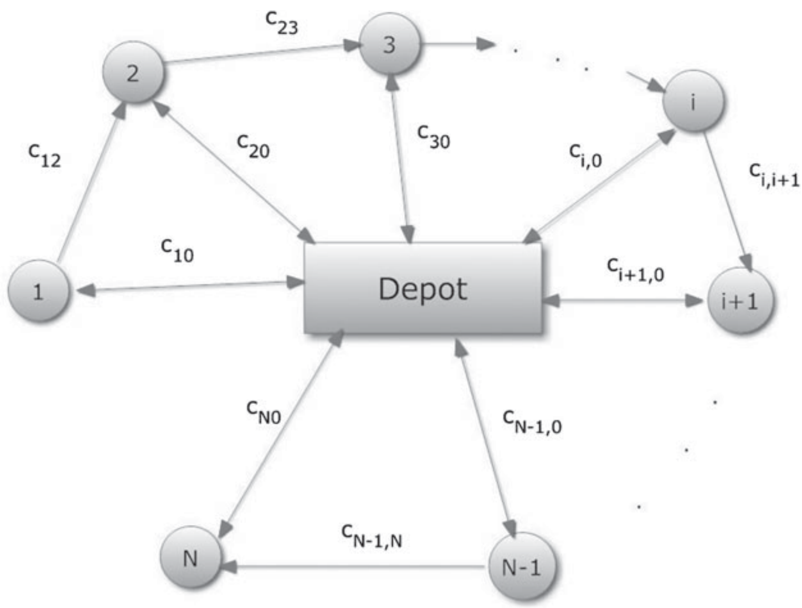

FIGURE 1. The road network for the finite-horizon problem.

the vehicle goes to the depot, restocks, and returns to satisfy the demand. It is assumed that there is no extra demand when the vehicle returns to the customer, that is, ξ_i^j, $1 \le i \le K$, remains unaltered.

Let z_i, $i = 1, 2, \ldots, K$, be the load of product i carried by the vehicle after the first visit at a customer's site; a negative value for z_i denotes the unsatisfied demand for product i. These quantities belong to the set $S = \{(z_1, \ldots, z_K) : \left|\sum_{i \in J} z_i\right| \le Q \text{ for all } J \subseteq \{1, \ldots, K\}\}$. Let $\zeta = \sum_{i=1}^{K} z_i^-$ with $z_i^- = \min\{0, z_i\}$. When $\zeta = 0$ (demand fully satisfied), the vehicle either proceeds directly to the next customer or goes to the depot, restocks with loads θ_i, $i = 1, 2, \ldots, K$, of products $1, 2, \ldots, K$, and then visits the next customer. We assume that $\sum_{i=1}^{K} \theta_i = Q$ since there is no advantage if the vehicle leaves some empty space when it goes to the depot for restocking. When $\zeta < 0$, the vehicle goes to the depot and restocks the owed quantity $-\zeta$. Then it has the following two choices: (i) it fills the remaining space with quantities θ_i of products $1, 2, \ldots, K$, where $\sum_{i=1}^{K} \theta_i = Q + \zeta$, returns to the customer, satisfies demand, and then proceeds to the next customer, (ii) returns to the customer, satisfies demand, makes a second trip to the depot where it restocks with loads θ_i of products $1, 2, \ldots, K$, where $\sum_{i=1}^{K} \theta_i = Q$, and proceeds to the next customer. Our objective is to determine a vehicle routing and replenishment strategy that minimizes the expected total cost during a visit cycle. Specifically, if after the first visit at a customer's site $\zeta = 0$, we must find if it is optimal for the vehicle to proceed to the next customer or to go to the depot for replenishment; if $\zeta < 0$ we must find if it is optimal for the vehicle to make one or two trips to the depot. In both cases we also have to determine the optimal quantities of the products that are loaded in the vehicle when it goes to the depot for replenishment. We define vectors $\overline{z} = [z_1, z_2, \ldots, z_K]$, $\overline{\xi}^j = \left[\xi_1^j, \xi_2^j, \ldots, \xi_K^j\right]$, $\overline{\theta} = [\theta_1, \theta_2, \ldots, \theta_K]$, and denote by $f_j(\overline{z})$ the minimum expected future cost when the load of product i carried by the vehicle after visiting customer j for the first time is equal to z_i. Then, an optimal routing strategy can be determined by the following dynamic programming equations (see e.g. Eq. (6.5) in Bather [1]). For $j = 1, 2, \ldots, N-1$ we have two cases.

Case 1: If $z_1, z_2, \ldots, z_K \geq 0$, then

$$f_j(\overline{z}) = \min\{H_j(\overline{z}), A_j\}. \tag{1}$$

Case 2: If $\sum_{i=1}^{K} z_i^- < 0$, then

$$f_j(\overline{z}) = 2c_{j0} + \min\left\{\widetilde{H}_j\left(\sum_{i=1}^{K} z_i^-\right), A_j\right\}, \tag{2}$$

where

$$H_j(\overline{z}) = c_{j,j+1} + Ef_{j+1}(\overline{z} - \overline{\xi}^{j+1}), \quad z_1, z_2, \ldots, z_K \geq 0, \tag{3}$$

$$\widetilde{H}_j(\zeta) = c_{j,j+1} + \min_{\overline{\theta}:\sum_{i=1}^{K}\theta_i = Q+\zeta} Ef_{j+1}(\overline{\theta} - \overline{\xi}^{j+1}), \quad \zeta < 0, \tag{4}$$

$$A_j = c_{j0} + c_{j+1,0} + \min_{\overline{\theta}:\sum_{i=1}^{K}\theta_i = Q} Ef_{j+1}(\overline{\theta} - \overline{\xi}^{j+1}). \tag{5}$$

In the boundary, we have

$$f_N(\overline{z}) = c_{N0} + 2c_{N0}\mathbf{1}\left(\sum_{i=1}^{K} z_i^- < 0\right). \tag{6}$$

Finally, the minimum total expected cost is

$$f_0 = c_{10} + \min_{\overline{\theta}:\sum_{i=1}^{K}\theta_i = Q} Ef_1(\overline{\theta} - \overline{\xi}^1). \tag{7}$$

In (3)–(5), (7), the expected values are taken with respect to the random vectors $\overline{\xi}^j$, $j = 1, \ldots, N$. The first term in the curly brackets in (1) corresponds to the action of proceeding to the next customer and the second term corresponds to the action of going to the depot for restocking before proceeding to the next customer. The first term in the curly brackets in (2) corresponds to the action of returning to the depot once before proceeding to the next customer and the second term corresponds to the action of returning to the depot twice before proceeding to the next customer. The following theorem characterizes the optimal vehicle routing strategy after it visits customer $j \in \{1, \ldots, N-1\}$ for the first time.

THEOREM 1:
(i) For each $z_1, z_2, \ldots, z_{K-1} \geq 0$ there exists integer $s_j(z_1, z_2, \ldots, z_{K-1}) \in \{0, 1, \ldots, Q + 1 - z_1 - \cdots - z_{K-1}\}$ such that it is optimal for the vehicle to proceed to the next customer iff $z_K \geq s_j(z_1, z_2, \ldots, z_{K-1})$.
(ii) $s_j(z_1, z_2, \ldots, z_{K-1})$ is non-increasing in each of its arguments.
(iii) There exists $s_j \leq 0$ such that it is optimal for the vehicle to make two trips to the depot iff $\sum_{i=1}^{K} z_i^- < s_j$.

PROOF: For parts (i) and (ii) it suffices to show that $H_j(\overline{z})$ is non-increasing in its arguments and for part (iii) that $\widetilde{H}_j(\zeta)$ is non-increasing in $\zeta < 0$. We will also need to prove that $f_j(\overline{z})$ is non-increasing in its arguments. The proof is by induction on j. First, the induction base is established by $f_N(\overline{z})$ being non-increasing (Eq. (6)). Then, assuming that $f_{j+1}(\overline{z})$ is non-increasing, we will show that $H_j(\overline{z})$, $\widetilde{H}_j(\zeta)$, and $f_j(\overline{z})$ are non-increasing.

Function $H_j(\overline{z})$ is non-increasing by the induction hypothesis and Eq. (3). Function $\widetilde{H}_j(\zeta)$ is non-increasing by Eq. (4) and the fact that $\min_{\overline{\theta}:\sum_{i=1}^K \theta_i = q} Ef_{j+1}(\overline{\theta} - \overline{\xi}^{j+1})$ is non-increasing in q. To see this, letting $\overline{\theta}_{(K-1)}$ and $\overline{\xi}^j_{(K-1)}$ be the vectors consisting of the first $K-1$ elements of $\overline{\theta}$ and $\overline{\xi}^j$, we have for $0 \le q' < q$

$$\begin{aligned}
&\min_{\overline{\theta}:\sum_{i=1}^K \theta_i = q} Ef_{j+1}(\overline{\theta} - \overline{\xi}^{j+1}) \\
&= \min_{\overline{\theta}_{(K-1)}:\sum_{i=1}^{K-1} \theta_i \le q} Ef_{j+1}\left(\overline{\theta}_{(K-1)} - \overline{\xi}^{j+1}_{(K-1)}, q - \sum_{i=1}^{K-1} \theta_i - \xi_K^{j+1}\right) \\
&\le \min_{\overline{\theta}_{(K-1)}:\sum_{i=1}^{K-1} \theta_i \le q'} Ef_{j+1}\left(\overline{\theta}_{(K-1)} - \overline{\xi}^{j+1}_{(K-1)}, q - \sum_{i=1}^{K-1} \theta_i - \xi_K^{j+1}\right) \\
&\le \min_{\overline{\theta}_{(K-1)}:\sum_{i=1}^{K-1} \theta_i \le q'} Ef_{j+1}\left(\overline{\theta}_{(K-1)} - \overline{\xi}^{j+1}_{(K-1)}, q' - \sum_{i=1}^{K-1} \theta_i - \xi_K^{j+1}\right) \\
&= \min_{\overline{\theta}:\sum_{i=1}^K \theta_i = q'} Ef_{j+1}(\overline{\theta} - \overline{\xi}^{j+1}),
\end{aligned} \tag{8}$$

where the last inequality follows from the induction hypothesis. It remains to show that $f_j(\overline{z})$ is non-increasing. Because of symmetry it suffices to show the monotonicity property with respect to one of the arguments, say z_K. For $\sum_{i=1}^{K-1} z_i^- < 0$ the result follows directly from the monotonicity of $\widetilde{H}_j(\zeta)$ and Eq. (2). Consider now $z_1, z_2, \ldots, z_{K-1} \ge 0$. For $0 \le z'_K < z_K$ and $z'_K < z_K < 0$ we get $f_j(\overline{z}_{(K-1)}, z_K) \le f_j(\overline{z}_{(K-1)}, z'_K)$ by the monotonicity of $H_j(\overline{z})$ and $\widetilde{H}_j(\zeta)$, respectively (Eqs (1) and (2)). To complete the proof we still need to show that $f_j(\overline{z}_{(K-1)}, 0) \le f_j(\overline{z}_{(K-1)}, -1)$. From Eq. (1) and the triangle inequality we get

$$\begin{aligned}
f_j(\overline{z}_{(K-1)}, 0) \le A_j &= c_{j0} + c_{j+1,0} + \min_{\overline{\theta}:\sum_{i=1}^K \theta_i = Q} Ef_{j+1}(\overline{\theta} - \overline{\xi}^{j+1}) \\
&\le 2c_{j0} + c_{j,j+1} + \min_{\overline{\theta}:\sum_{i=1}^K \theta_i = Q} Ef_{j+1}(\overline{\theta} - \overline{\xi}^{j+1}) = B_j.
\end{aligned} \tag{9}$$

From Eqs (4), (9), and (8) we get

$$\begin{aligned}
B_j - 2c_{j0} - \widetilde{H}_j(-1) = &\min_{\overline{\theta}:\sum_{i=1}^K \theta_i = Q} Ef_{j+1}(\overline{\theta} - \overline{\xi}^{j+1}) \\
&- \min_{\overline{\theta}:\sum_{i=1}^K \theta_i = Q-1} Ef_{j+1}(\overline{\theta} - \overline{\xi}^{j+1}) \le 0.
\end{aligned} \tag{10}$$

From (2) we have

$$f_j(\overline{z}_{(K-1)}, -1) = 2c_{j0} + \min\left\{\widetilde{H}_j(-1), A_j\right\}. \tag{11}$$

If $A_j \le \widetilde{H}_j(-1)$, we get from Eqs (9) and (11)

$$f_j(\overline{z}_{(K-1)}, 0) \le A_j < 2c_{j0} + A_j = f_j(\overline{z}_{(K-1)}, -1).$$

If $A_j > \widetilde{H}_j(-1)$, we get from Eqs (9)–(11)

$$f_j(\overline{z}_{(K-1)}, 0) \le B_j \le 2c_{j0} + \widetilde{H}_j(-1) = f_j(\overline{z}_{(K-1)}, -1),$$

which completes the proof. ■

Remark 1: If $s_j(z_1, z_2, \ldots, z_{K-1}) = Q + 1 - z_1 - \cdots - z_{K-1}$, it is optimal for the vehicle to return to the depot for all $z_K \in \{0, \ldots, Q - z_1 - \cdots - z_{K-1}\}$.

Remark 2: Tatarakis and Minis [15] chose as decision epochs for this problem the epochs at which the vehicle has completed the service of each customer and presented part (i) of the above theorem for $K = 2$. The proof is included in Tatarakis [14]. This proof involves complicated algebraic expressions and cannot be extended for $K > 2$.

Remark 3: The computational complexity of the dynamic programming algorithm based on Eqs (1)–(6) is $O(NQ^K)$, for any value of $K \geq 1$.

Remark 4: Consider a more general problem in which the total demand of each customer for all products may exceed the vehicle capacity, that is, it is possible that $\sum_{i=1}^{K} \xi_i^j$ is greater than Q for $j \in \{1, \ldots N\}$. In this case it is possible after the first visit at a customer's site that $\zeta < -Q$. If this inequality holds, the vehicle must go to the depot to restock the owed quantity more than once. It is assumed that there is no extra demand for products at the 2nd, 3rd,... visit to customer j, that is, ξ_i^j, $1 \leq i \leq K$, remains unaltered. Let $\zeta' < 0$ be the amount owed before the last required trip to the depot. The possible decisions are the same as in the original problem. Equations (1), (3)–(5) remain the same. Equation (2) becomes

$$f_j(\overline{z}) = 2c_{j0}\lceil -\zeta/Q \rceil + \min\left\{ \widetilde{H}_j(\zeta'), A_j \right\},$$

where $\lceil -\zeta/Q \rceil$ is the smallest integer that it is not smaller than $-\zeta/Q$. Equation (6) becomes

$$f_N(\overline{z}) = c_{N0} + 2c_{N0}\lceil -\zeta/Q \rceil.$$

Theorem 1 holds in this case. Its proof is similar but slightly more complicated than the proof that we presented for the original problem. We point out that our dynamic programming approach can be applied only if there is no extra demand for products at the 2nd, 3rd, ... visit to customer $j \in \{1, \ldots, N-1\}$. However, it seems that this assumption is quite restrictive in many practical applications as those that we mentioned in Section 1.

Remark 5: Consider another modification of the problem in which the demand of customer j, $j = 1, \ldots, N$, for product i, $i = 1, \ldots, K$, is a continuous random variable ξ_i^j and the joint probability density function of $(\xi_1^j, \ldots, \xi_K^j)$, $j = 1, \ldots, N$, is known. Equations (1)–(6) remain valid in this case and the result of Theorem 1 can be proved in the same way as in the case of discrete demands. However, the assumption that all products are stored together in the vehicle seems to be questionable when the demands of the products are continuous random variables. It would be more reasonable to assume that the vehicle is divided into K compartments and each product is stored in its dedicated compartment. Note that the problem with compartmentalized load and continuous stochastic demands has been studied in Pandelis et al. [9]. A realistic example that fits to some extent in the model with unified load and continuous demands could be the routing of a vehicle that delivers building materials such as lime, sand, and pebble that are stored together in the unique compartment of the vehicle but are not mixed up.

Let S^+ be the set of $(z_1, z_2, \ldots, z_{K-1})$ such that $z_1, \ldots, z_{K-1} \geq 0$, $\sum_{i=1}^{K-1} z_i \leq Q$, and let M denote its cardinality. With the elements of S^+ ordered in some arbitrary way, let Z_m, $m = 1, \ldots, M$, denote its mth element. Then, in view of Theorem 1 the optimal policy, that is, the critical numbers s_j and $s_j(z_1, \ldots, z_{K-1})$, $z_1, \ldots, z_{K-1} \geq 0$, $\sum_{i=1}^{K-1} z_i \leq Q$, for each customer $j \in \{1, \ldots, N-1\}$, can be found by the following special-purpose dynamic programming algorithm:

Algorithm
Determination of critical numbers s_j and $s_j(z_1, \ldots, z_{K-1})$,
$z_1, \ldots, z_{K-1} \geq 0$, $\sum_{i=1}^{K-1} z_i \leq Q$, for each customer $j = 1, \ldots, N-1$.

Step 0. Set $f_N(z_1, z_2, \ldots, z_K) = c_{N0} + 2c_{N0}\mathbf{1}\left(\sum_{i=1}^{K} z_i^- < 0\right)$, $(z_1, z_2, \ldots, z_K) \in S$, and $j = N-1$.

Step 1. Set $\zeta = -1$.

Step 2. (Determination of critical number s_j)

If $\widetilde{H}_j(\zeta) > A_j$, do the following:

1. Set $s_j = \zeta + 1$.
2. For $(z_1, z_2, \ldots, z_K)$ such that $s_j \leq \sum_{i=1}^{K} z_i^- < 0$ set $f_j(\overline{z}) = 2c_{j0} + \widetilde{H}_j\left(\sum_{i=1}^{K} z_i^-\right)$.
3. For $(z_1, z_2, \ldots, z_K)$ such that $\sum_{i=1}^{K} z_i^- < s_j$ set $f_j(\overline{z}) = 2c_{j0} + A_j$.
4. Go to Step 3.

Otherwise, set $\zeta = \zeta - 1$.

If $\zeta = -Q - 1$, do the following:

1. Set $s_j = -Q$.
2. For $(z_1, z_2, \ldots, z_K)$ such that $\sum_{i=1}^{K} z_i^- < 0$ set $f_j(\overline{z}) = 2c_{j0} + \widetilde{H}_j\left(\sum_{i=1}^{K} z_i^-\right)$.
3. Go to Step 3.

Otherwise, go to Step 2.

Step 3. Set $m = 1$.

Step 4. Set $(z_1, z_2, \ldots, z_{K-1}) = Z_m$ and $z_K = Q - \sum_{i=1}^{K-1} z_i$.

Step 5. (Determination of $s_j(z_1, z_2, \ldots, z_{K-1})$)

If $H_j(z_1, z_2, \ldots, z_K) > A_j$, do the following:

1. Set $s_j(z_1, z_2, \ldots, z_{K-1}) = z_K + 1$.
2. Set $f_j(\overline{z}) = A_j$ for $z_K \in \{0, \ldots, s_j(z_1, z_2, \ldots, z_{K-1}) - 1\}$.
3. Set $f_j(\overline{z}) = H_j(\overline{z})$ for $z_K \in \{s_j(z_1, z_2, \ldots, z_{K-1}), \ldots, Q - \sum_{i=1}^{K-1} z_i\}$.
4. Set $m = m + 1$. If $m \leq M$, go to step 4. Otherwise, go to step 6.

Otherwise, set $z_K = z_K - 1$.

If $z_K = -1$, do the following:

1. Set $s_j(z_1, z_2, \ldots, z_{K-1}) = 0$.
2. Set $f_j(\overline{z}) = H_j(\overline{z})$ for $z_K \in \{0, \ldots, Q - \sum_{i=1}^{K-1} z_i\}$.
3. Set $m = m + 1$. If $m \leq M$, go to Step 4. Otherwise, go to Step 6.

Otherwise, go to Step 5.

Step 6. Set $j = j - 1$. If $j \geq 1$, go to Step 1. Otherwise, stop.

The above algorithm is faster than the initial dynamic programming algorithm based on Eqs (1)–(6), since it does not calculate for $j = 1, \ldots, N-1$ the quantities $\widetilde{H}_j(\zeta)$ for

TABLE 1. The Critical Numbers of the Optimal Policy

Customer j	$s_j(0)$	$s_j(1)$	$s_j(2)$	$s_j(3)$	$s_j(4)$	$s_j(5)$	$s_j(6)$	s_j
1	7,7	6,2	3,1	2,1	2,1	2,1	1,1	−2,−3
2	7,2	1,1	1,0	1,0	1,0	1,0	1,0	−4,−4
3	7,3	4,1	2,1	2,0	1,0	1,0	1,0	−2,−4
4	7,1	1,0	1,0	1,0	1,0	1,0	1,0	−4,−5
5	7,2	6,1	2,0	2,0	2,0	2,0	1,0	−2,−4
6	7,7	6,4	3,2	2,1	2,1	2,1	1,1	−1,−2
7	7,1	2,0	1,0	1,0	1,0	1,0	1,0	−3,−5
8	7,7	6,6	2,1	2,1	1,0	1,0	1,0	−2,−3
9	7,7	6,0	0,0	0,0	0,0	0,0	0,0	−4,−5

$\zeta = z_1^- + z_2^- + \cdots + z_K^- < s_j - 1$ and the quantities $H_j(z_1, z_2, \ldots, z_K)$, $z_1, \ldots, z_{K-1} \geq 0$ such that $0 \leq \sum_{i=1}^{K-1} z_i \leq Q$, for $z_K \in \{0, \ldots, s_j(z_1, \ldots, z_{K-1}) - 1\}$. We point out that the computational burden of the above algorithm increases rapidly as K increases or Q increases since the critical number $s_j(z_1, \ldots, z_{K-1})$ is determined for all $(z_1, \ldots, z_{K-1})$ in set S^+ whose cardinality is equal to $\prod_{i=1}^{K-1}(Q+i)/(K-1)!$ As illustration we present the two following examples for $K = 2$.

Example 1: Suppose that $N = 10$, $Q = 6$, $K = 2$. We give below the symmetric matrix $C = (c_{ij}), 0 \leq i, j \leq 10$ whose non-zero elements are the travel costs $c_{j,j+1}$ between customer $j \in \{1, \ldots, 9\}$ and customer $j+1$, and the travel costs c_{j0} between customer $j \in \{1, \ldots, 10\}$ and the depot. We observe that these costs satisfy the triangle inequality.

$$C = \begin{pmatrix} 0 & 6 & 7 & 8 & 7 & 5 & 4 & 8 & 6 & 5 & 8 \\ 6 & 0 & 9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & 9 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 0 & 6 & 0 & 9 & 0 & 0 & 0 & 0 & 0 & 0 \\ 7 & 0 & 0 & 9 & 0 & 5 & 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 & 5 & 0 & 7 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 & 7 & 0 & 10 & 0 & 0 & 0 \\ 8 & 0 & 0 & 0 & 0 & 0 & 10 & 0 & 9 & 0 & 0 \\ 6 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 0 & 8 & 0 \\ 5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 & 0 & 9 \\ 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 9 & 0 \end{pmatrix}$$

We assume that the demands of customer $j \in \{1, \ldots, 10\}$ for products 1 and 2 are independent and follow the binomial distribution $B(3, 0.4)$, that is, $\Pr(\xi_1^j = x) = \Pr(\xi_2^j = x) = \binom{3}{x} 0.4^x 0.6^{3-x}$, $x = 0, \ldots, 3$. This assumption implies that $0 \leq \xi_1^j + \xi_2^j \leq 6$, $j \in \{1, \ldots, 10\}$, as required. For each customer $j = 1, \ldots, 9$ we present in each cell of Table 1 the critical numbers $s_j(z_1)$, $z_1 = 0, \ldots, Q$, and s_j of the optimal policy that we computed by implementing the initial dynamic programming algorithm based on Equations (1)–(6) and our special purpose dynamic programming algorithm. The first number in each cell is the critical number if the travel costs are given by the symmetric matrix C while the second number is the critical number if the travel costs are given by the symmetric matrix $C' = (c'_{ij})$, $0 \leq i, j \leq 10$, where $c'_{ij} = c_{ij}$ for $1 \leq i, j \leq 10$, $c'_{0j} = 2c_{0j}$ for $0 \leq j \leq 10$, and $c'_{i0} = 2c_{i0}$ for $0 \leq i \leq 10$. Note that Part (ii) of Theorem 1 is confirmed numerically since $s_j(z_1)$ is non-increasing in $z_1 \in \{0, \ldots, 6\}$ for $j \in \{1, \ldots, 9\}$ if the travel costs are given by matrix C or C'. Note also that $s_j(0) = 7$, $j \in \{1, \ldots, 9\}$, if the travel costs are given by matrix C, meaning that, according to Remark 1, if after the first visit to customer $j \in \{1, \ldots, 9\}$ the vehicle has no

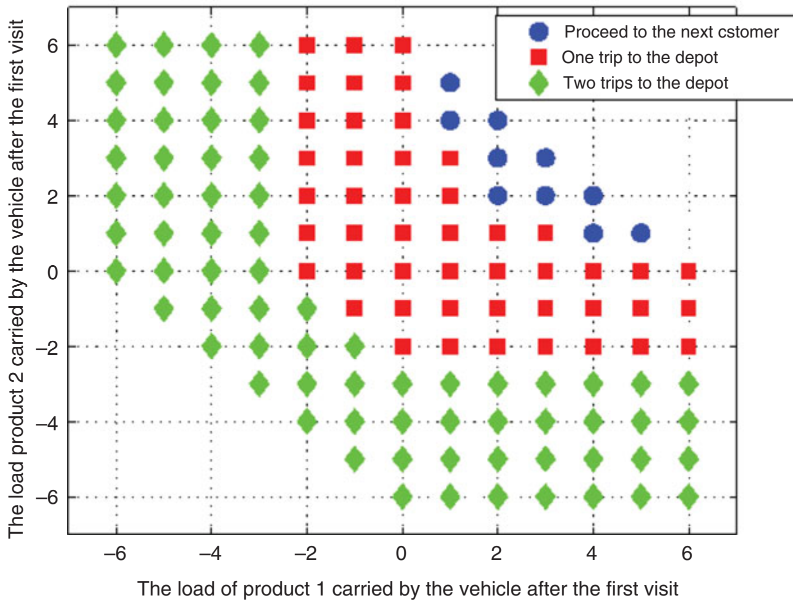

FIGURE 2. (Color online) The optimal decisions after the first visit to customer 3 if the travel costs are given by matrix C.

items of product 1, then the optimal decision is to return to the depot for replenishment even if it contains six items of product 2. We also observe from Table 1 that $s_9(z_1) = 0$, $1 \leq z_1 \leq 6$, if the travel costs are given by matrix C'. This means that if after the first visit to the ninth customer the vehicle contains at least one item of product 1, then the optimal decision is to proceed directly to the tenth customer if there is no unsatisfied demand for product 2. From Table 1 we also see that the first number in each cell is greater than or equal to the second number. This is intuitively reasonable since it seems preferable for the vehicle to avoid the trip to the depot if the travel costs between the customers and the depot are doubled. In Figures 2 and 3, we present the optimal decision for each state $(z_1, z_2) \in S$ after the first visit to the third customer, if the travel costs are given by matrices C and C', respectively. In these figures the action of proceeding to the next customer is denoted by a blue dot, the action of returning to the depot once is denoted by a red square, and the action of making two trips to the depot is denoted by a green rhomb.

We implemented the algorithms by running the corresponding Matlab programs on a personal computer equipped with an Intel Core 2 Duo, 2.5 GHz processor and 4GB of RAM. The computation time (0.316 or 0.378 s if the travel costs are given by C or C', respectively) of the special purpose algorithm is considerably smaller than the computation time (0.421 or 0.521 s if the travel costs are given by C or C', respectively) of the initial dynamic programming algorithm. The minimum expected total cost $f_0 = c_{10} + \min_{0 \leq \theta \leq 6} \sum_{x=0}^{3} \sum_{y=0}^{3} \Pr(\xi_1^1 = x) \Pr(\xi_2^1 = y) f_1(\theta - x, 6 - \theta - y)$ is found to be approximately equal to 112.7 or 131.1, if the travel costs are given by C or C', respectively. The dynamic programming algorithm equations enable us to determine the optimal quantities of products 1 and 2 that are loaded in the vehicle when it returns to the depot for replenishment. For example, if the travel costs are given by matrix C and after the first visit of the vehicle to the third customer the state is $(-2, 0)$, then the optimal decision is to go to the depot to restock two items of

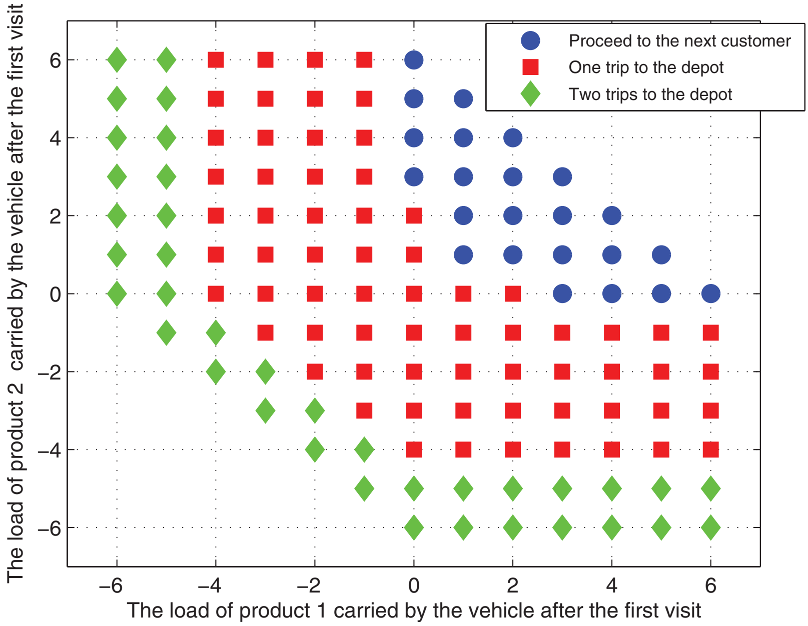

FIGURE 3. (Color online) The optimal decisions after the first visit to customer 3 if the travel costs are given by matrix C'.

product 1 and four items of product 2, return to customer 3 to deliver two items of product 1 (owed quantity), and then proceed to customer 4. If the state is $(3, -3)$, then the optimal decision is to return to the depot to load three items (unsatisfied demand) of product 2, return to customer 3 to satisfy the demand, make a second trip to the depot to load three items of product 1 and three items of product 2, and then go to customer 4.

Consider now the same example with variable Q and travel costs given by matrix C, where $Q \in \{6, 8, 10, \ldots, 40\}$ and $\Pr(\xi_1^j = x) = \Pr(\xi_2^j = x) = \binom{Q/2}{x} 0.4^x\, 0.6^{Q/2-x}$, $j = 1, \ldots, 10$. In Figure 4, we present graphs that show, as Q varies, the variation in the computation times (expressed in seconds) required by the initial dynamic programming algorithm based on Eqs (1)–(6) and the special-purpose dynamic programming algorithm. We observe that, as Q increases, the computation times for both algorithms increase non-linearly. The computation time required by the special-purpose algorithm is considerably smaller than the computation time required by our initial algorithm, especially for high values of Q. In Figure 5, we present a graph that shows the variation in the minimum total expected cost f_0 as Q varies. We see that, as Q increases, the minimum total expected cost tends to decrease. This is intuitively reasonable since an increase of the capacity of the vehicle causes an increase of the probability that it satisfies the demands of the customers when it visits them for the first time. In this case it not necessary to go to the depot for stock replenishment.

Example 2: Suppose that $N = 10$, $Q = 10$, $K = 2$. The travel costs are given by matrix C of the previous example. We assume that the demands of each customer $j \in \{1, \ldots, 10\}$ for products 1 and 2 are independent and follow the binomial distribution $B(5, p)$, that is, $\Pr(\xi_1^j = x) = \Pr(\xi_2^j = x) = \binom{5}{x} p^x\, (1-p)^{5-x}$, $x = 0, \ldots, 5$. This assumption implies that $0 \leq \xi_1^j + \xi_2^j \leq 10$, $j \in \{1, \ldots, 10\}$, as required. In Table 2 we present, for each customer

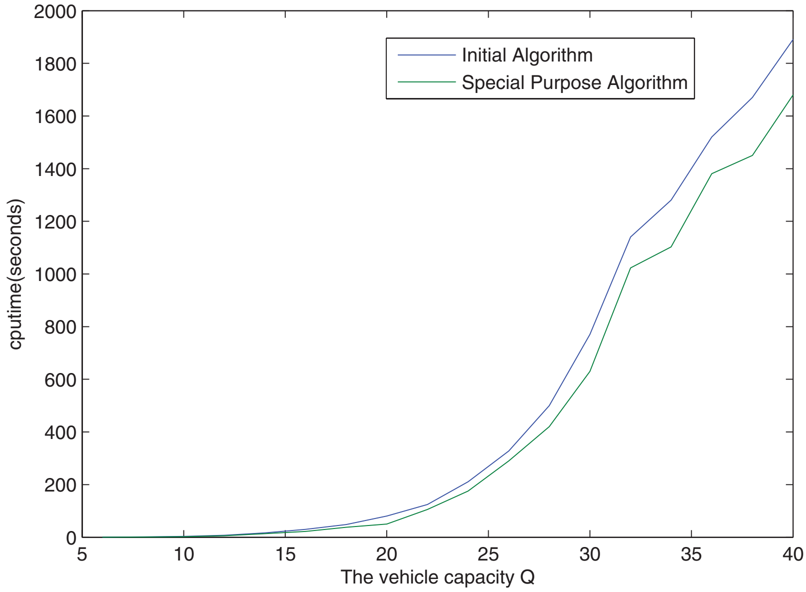

FIGURE 4. (Color online) The computation times of the algorithms as Q varies.

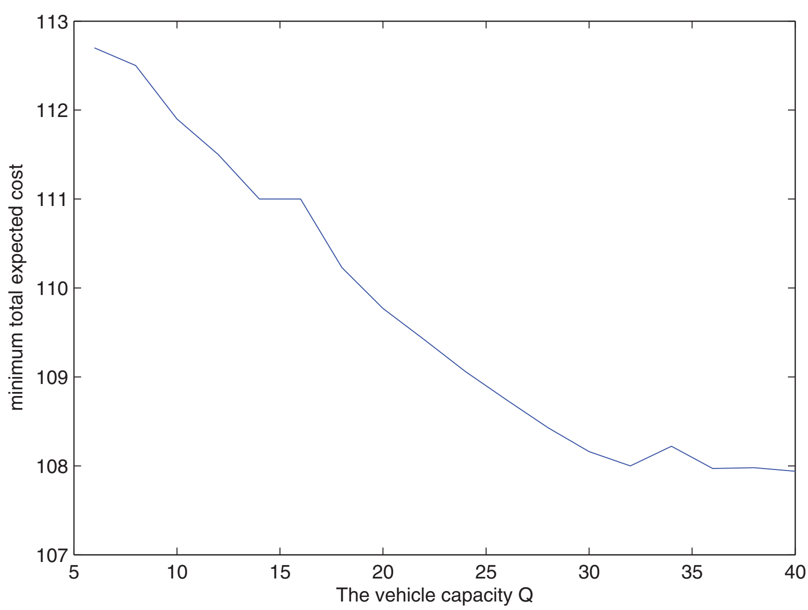

FIGURE 5. (Color online) The minimum total expected cost as Q varies.

$j = 1, \ldots, 9$ and for each value of $p \in \{0.2, 0.4, 0.6, 0.8\}$, the critical numbers $s_j(z_1)$, $z_1 = 0, \ldots, Q$, and s_j of the optimal policy that we computed by implementing the initial dynamic programming algorithm based on (1)–(6) and the special-purpose dynamic programming algorithm. In each cell of the table the first critical number corresponds to $p = 0.2$, the second critical number corresponds to $p = 0.4$, the third critical number corresponds to $p = 0.6$, and the fourth critical number corresponds to $p = 0.8$.

https://doi.org/10.1017/S0269964812000277 Published online by Cambridge University Press

Table 2. The Critical Numbers of the Optimal Policy for each Value of $p = \{0.2, 0.4, 0.6, 0.8\}$

j	$s_j(0)$	$s_j(1)$	$s_j(2)$	$s_j(3)$	$s_j(4)$	$s_j(5)$	$s_j(6)$	$s_j(7)$	$s_j(8)$	$s_j(9)$	$s_j(10)$	s_j
1	11,11	10,10	9,9	4,4	3,4	3,3	3,3	3,3	3,3	1,2	1,1	−3,−2
	11,11	10,10	10,10	4,8	4,5	3,4	3,4	3,4	3,3	2,2	1,1	−2,−1
2	11,11	2,2	1,2	1,2	1,2	1,2	1,2	1,2	2,2	1,2	1,1	−7,−6
	11,11	10,10	4,9	3,8	2,4	2,4	2,4	2,4	2,3	2,2	1,1	−4,−2
3	11,11	10,10	2,4	2,3	2,3	2,2	2,2	2,2	2,2	2,2	1,1	−6,−4
	11,11	10,10	4,9	3,8	3,4	2,4	2,4	2,4	2,3	2,2	1,1	−4,−4
4	11,11	1,3	1,2	1,1	1,1	1,1	1,1	1,1	1,1	1,1	1,1	−8,−6
	11,11	10,10	5,9	3,8	2,4	2,4	2,4	2,4	2,3	2,2	1,1	−4,−2
5	11,11	10,10	2,9	2,3	2,3	2,3	2,3	2,3	2,3	2,2	1,1	−6,−4
	11,11	10,10	9,9	8,8	4,7	4,5	4,4	4,4	3,3	2,2	1,1	−2,0
6	11,11	10,10	9,9	5,8	4,4	3,4	3,4	3,4	1,2	2,2	1,1	−2,−2
	11,11	10,10	9,9	8,8	4,7	4,5	4,4	4,4	3,3	2,2	1,1	−2,0
7	11,11	3,10	2,3	1,2	1,2	1,2	1,2	1,2	1,2	1,2	1,1	−7,−5
	11,11	10,10	9,9	4,8	4,8	3,4	3,4	3,4	3,3	2,2	1,1	−3,−1
8	11,11	10,10	9,9	2,8	1,3	1,3	1,2	1,2	1,2	1,2	1,1	−5,−3
	11,11	10,10	9,9	8,8	3,5	3,4	3,4	3,3	3,3	2,2	1,1	−3,−1
9	11,11	10,10	0,9	0,9	0,0	0,0	0,0	0,0	0,0	0,0	0,0	−8,−7
	11,11	10,10	10,10	9,9	7,7	0,0	0,0	0,0	0,0	0,0	0,0	−6,−5

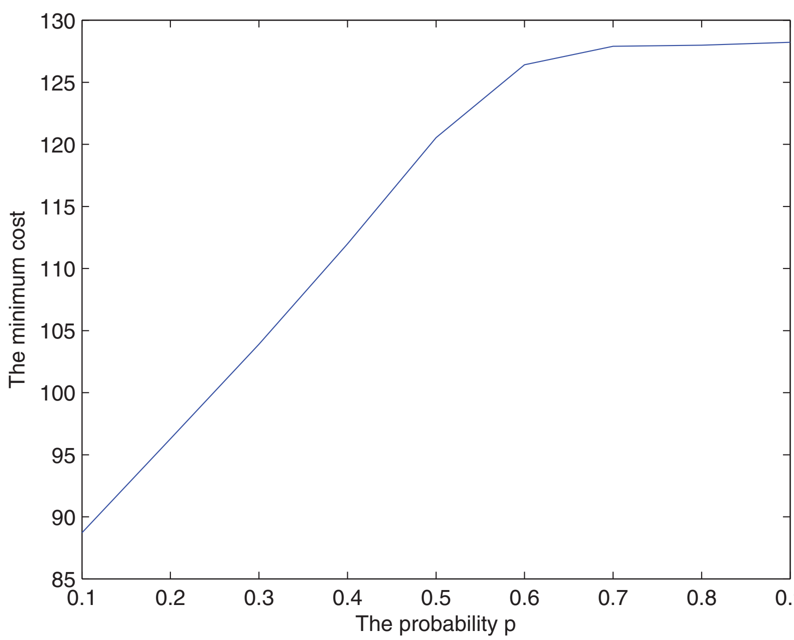

Figure 6. (Color online) The minimum total expected cost as p varies.

Note that Part (ii) of Theorem 1 is confirmed numerically since, for fixed p, $s_j(z_1)$ is non-increasing in $z_1 \in \{0, \ldots, 10\}$ for $j \in \{1, \ldots, 9\}$. We also observe that, for each customer $j \in \{1, \ldots, 9\}$, $s_j(z_1)$ and s_j are non-decreasing as p increases. This is intuitively reasonable since, as p increases, the expected values of the demands for products 1 and 2 increase and, therefore, the action of returning to the depot for restocking becomes more favourable. In Figure 6, we present a graph that shows the variation in the minimum total expected cost

f_0 as p varies. We see that as p takes values in the set $\{0.1, \ldots, 0.6\}$ the minimum total expected cost increases approximately linearly. When p takes values in the set $\{0.7, 0.8, 0.9\}$ the minimum total expected cost increases very slowly.

From a large number of examples that we tested there is strong evidence that the special-purpose dynamic programming algorithm can handle problems with quite large values of N and Q but small values of K (i.e., $N = 100$, $Q = 40$, and $K = 3$) or moderate values of N, Q, and K (i.e., $N = 50$, $Q = 20$, and $K = 4$). This is due to the fact that the computation time required by the algorithm seems to increase linearly as N increases, polynomially with degree K as Q increases, and exponentially as K increases.

3. THE INFINITE-HORIZON PROBLEM

We modify the problem that we introduced in the previous section by considering an infinite-time horizon problem in which the service of the customers does not stop when the last customer N has been serviced but it continues periodically with the same customer order. This means that after the service of customer N has been completed, the vehicle services again customer 1, customer 2, and so on. Let c_{N1} denote the travel cost from customer N to customer 1. The road network is depicted in Figure 7.

We assume that the distribution of the random vector $\overline{\xi}^j = (\xi_1^j, \ldots, \xi_K^j)$ that represents the demands of customer $j \in \{1, \ldots, N\}$ for products $1, \ldots, K$ remains the same at each cycle (period). We suppose that at each cycle the vehicle visits each customer, satisfies as much demand as possible, and chooses one decision among some possible decisions that coincide with the possible decisions in the finite-horizon problem. If the whole customer's demand is satisfied, there are two possible decisions: (i) to proceed directly to the next customer, and (ii) to go to the depot for restocking and then go to the next customer. If part of the customer's demand is not satisfied, the possible options are: (i) to go to the

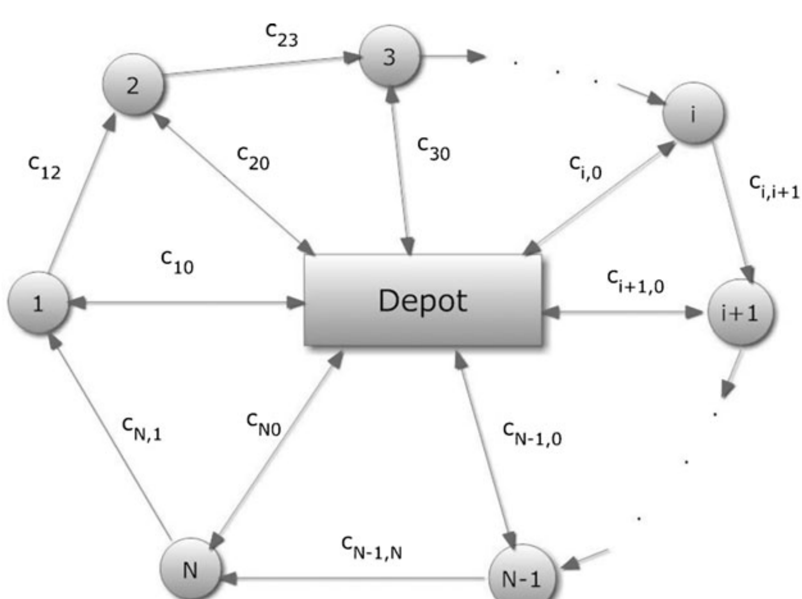

FIGURE 7. The road network for the infinite-horizon problem.

https://doi.org/10.1017/S0269964812000277 Published online by Cambridge University Press

depot for restocking the owed demand and for loading additional items of all products, return to the customer to satisfy the owed demand and then go to the next customer, and (ii) to go to the depot for restocking the owed quantity, return to the customer to deliver the owed quantity, make a second trip to the depot for replenishment and then go to the next customer. We assume that the decision epochs $\tau = 0, 1, \ldots$ are equidistant time epochs. It is also assumed that the time interval between two consecutive decision epochs is greater than the required time for the trips of the vehicle if it follows any of the above decisions. Although we impose these assumptions in order to apply well-known results from the theory of Markov decision processes, there are situations in which these assumptions may hold, as the practical applications that we mentioned in Section 1. Specifically, in the first application (ex-van sales) suppose that the vehicle supplies the customers with products (e.g., milk and fruits) that must be consumed in a short period. It seems reasonable that the supply of the customers does not stop when the last customer has been serviced but is continues with the same customer order for a long time horizon. In the second application (self propelled vehicle in a manufacturing shop) it can be assumed that the supply of the workcenters with discrete parts does not stop when the last workcenter has been supplied but it continues with the same order for a long time without interruption. For these two examples it can be assumed that the decisions are selected at equidistant time epochs (e.g., every hour).

The routing of the vehicle in the infinite-horizon setting is controlled by a policy π that is a rule for choosing decisions at epochs $\tau = 0, 1, \ldots$. The decision that is chosen by a policy at a decision epoch may depend on the history of the process or may be randomized in the sense that it is chosen by specific probabilities. An appealing class of policies is the class of stationary policies. A stationary policy chooses at each decision epoch a decision that depends only on the current state of the system. Thus, a stationary policy can be specified by a mapping from the state space of the system to the set of possible decisions.

We consider two different optimization criteria for the infinite-horizon problem. The first criterion is the minimization of the expected total discounted cost and the second criterion is the minimization of the expected long-run average cost per unit time. The expected total discounted cost of a policy π is defined as the expected total cost during an infinite-time horizon if the costs are discounted at a rate $\alpha \in (0, 1)$ per unit time given that policy π is employed. The use of the discount factor α can be explained by the economic idea that a cost to be incurred in the future is discounted in present value. The expected long-run average cost per unit time of a policy π is defined as the limit as $n \to \infty$ of the expected cost incurred until the nth decision epoch divided by n, given that policy π is employed. The discounted cost criterion becomes preferable to the average cost criterion when the time intervals between two consecutive decision epochs are sufficiently long so that the time value of money should be taken into account when adding costs in future intervals to the cost in the current interval. This assumption does not seem to hold in the two practical applications we have mentioned. Using well-known results of Markov decision processes (see Chapter 6 in Ross [11]) we will see that, under either one of these criteria, the optimal policy has the same structure as the optimal policy in the finite-horizon problem.

The state space I of the system consists of all states $(j, \overline{z})$, where $j = 1, \ldots, N$ is the customer and $\overline{z} = (z_1, \ldots, z_K)$ are all possible loads of products that remain in the vehicle after it has visited customer j and has satisfied as much demand as possible. Let $V_n^{\alpha}(j, \overline{z})$, $(j, \overline{z}) \in I$, $0 < \alpha < 1$, be the minimum $n-$step expected discounted cost if the initial state is $(j, \overline{z}) \in I$ and α is the discount factor. This quantity satisfies the following dynamic programming equations for $n = 1, 2, \ldots$.

https://doi.org/10.1017/S0269964812000277 Published online by Cambridge University Press

If $z_1, \ldots, z_K \geq 0$, then

$$V_n^\alpha(j,\overline{z}) = \min\Big\{c_{j,j+1} + \alpha E V_{n-1}^\alpha\left(j+1, \overline{z} - \overline{\xi}^{j+1}\right), \\ c_{j0} + c_{0,j+1} + \alpha \min_{\overline{\theta}:\sum_{i=1}^K \theta_i = Q} E V_{n-1}^\alpha\left(j+1, \overline{\theta} - \overline{\xi}^{j+1}\right)\Big\},$$

and if $\zeta = \sum_{i=1}^K z_i^- < 0$, then

$$V_n^\alpha(j,\overline{z}) = 2c_{j0} + \min\Big\{c_{j,j+1} + \alpha \min_{\overline{\theta}:\sum_{i=1}^K \theta_i = Q+\zeta} E V_{n-1}^\alpha\left(j+1, \overline{\theta} - \overline{\xi}^{j+1}\right), \\ c_{j0} + c_{0,j+1} + \alpha \min_{\overline{\theta}:\sum_{i=1}^K \theta_i = Q} E V_{n-1}^\alpha\left(j+1, \overline{\theta} - \overline{\xi}^{j+1}\right)\Big\}.$$

We also have that $V_0^\alpha(j,\overline{z}) = 0$, $(j,\overline{z}) \in I$. In the above equations we assume that $N+1$ is equal to 1 since the customer next to N is customer 1. It can be shown by induction on n that $V_n^\alpha(j,\overline{z})$ is non-increasing in z_i, $i = 1, \ldots, K$, in the same way we proved that $f_j(\overline{z})$ is non-increasing in its arguments in Theorem 1. Let $V^\alpha(j,\overline{z})$, $(j,\overline{z}) \in I$, denote the $\alpha-$discounted total expected cost if the initial state is $(j,\overline{z}) \in I$. This quantity is finite since the state space I is finite. It satisfies the following optimality equations:
If $z_1, \ldots, z_K \geq 0$, then

$$V^\alpha(j,\overline{z}) = \min\Big\{c_{j,j+1} + \alpha E V^\alpha\left(j+1, \overline{z} - \overline{\xi}^{j+1}\right), \\ c_{j0} + c_{0,j+1} + \alpha \min_{\overline{\theta}:\sum_{i=1}^K \theta_i = Q} E V^\alpha\left(j+1, \overline{\theta} - \overline{\xi}^{j+1}\right)\Big\},$$

and if $\sum_{i=1}^K z_i^- = \zeta < 0$, then

$$V^\alpha(j,\overline{z}) = 2c_{j0} + \min\Big\{c_{j,j+1} + \alpha \min_{\overline{\theta}:\sum_{i=1}^K \theta_i = Q+\zeta} E V^\alpha\left(j+1, \overline{\theta} - \overline{\xi}^{j+1}\right), \\ c_{j0} + c_{0,j+1} + \alpha \min_{\overline{\theta}:\sum_{i=1}^K \theta_i = Q} E V^\alpha\left(j+1, \overline{\theta} - \overline{\xi}^{j+1}\right)\Big\}.$$

It is well known (see Corollary 6.6 in Ross [11]) that, as $n \to \infty$, $V_n^\alpha(j,\overline{z}) \to V^\alpha(j,\overline{z})$. This implies that $V^\alpha(j,\overline{z})$ is non-increasing in z_i, $i = 1, \ldots, K$. Hence, the first terms in the curly brackets in the above optimality equations are non-increasing in z_i, $i = 1, \ldots, K$, and ζ, respectively. This implies that the $\alpha-$discounted cost optimal policy has the threshold-type structure described in Theorem 1.

We focus now on the minimization of the expected average cost. First we note that the state $(1,\overline{0}) \in I$ is accessible from any other state under any stationary policy. From Corollary 6.20 in Ross [11] it follows that there exist numbers g and $h(j,\overline{z})$, $(j,\overline{z}) \in I$ such that the following equations hold.
If $z_1, \ldots, z_K \geq 0$, then

$$h(j,\overline{z}) = \min\Big\{c_{j,j+1} - g + E h\left(j+1, \overline{z} - \overline{\xi}^{j+1}\right), \\ c_{j0} + c_{0,j+1} - g + \min_{\overline{\theta}:\sum_{i=1}^K \theta_i = Q} E h\left(j+1, \overline{\theta} - \overline{\xi}^{j+1}\right)\Big\},$$

https://doi.org/10.1017/S0269964812000277 Published online by Cambridge University Press

and if $\sum_{i=1}^{K} z_i = \zeta < 0$, then

$$h(j,\overline{z}) = 2c_{j0} + \min\Big\{c_{j,j+1} - g + \min_{\overline{\theta}:\sum_{i=1}^{K}\theta_i = Q+\zeta} Eh\left(j+1, \overline{\theta} - \overline{\xi}^{j+1}\right), \\ c_{j0} + c_{0,j+1} - g + \min_{\overline{\theta}:\sum_{i=1}^{K}\theta_i = Q} Eh\left(j+1, \overline{\theta} - \overline{\xi}^{j+1}\right)\Big\}.$$

The above equations are known as the average-cost optimality equations. The number g is the minimum average cost and does not depend on the initial state of the system. There also exists a sequence $\alpha_n \to 1$ (see Theorem 6.18 in Ross [11]) such that

$$h(j,\overline{z}) = \lim_{n\to\infty}\left[V^{\alpha_n}(j,\overline{z}) - V^{\alpha_n}(1,\overline{0})\right],\ (j,\overline{z}) \in I.$$

The monotonicity of $V^{\alpha_n}(j,\overline{z})$ with respect to z_i, $i = 1,\ldots,K$, implies that the first terms in the curly brackets in the above average-cost optimality equations are non-increasing with respect to z_i, $i = 1,\ldots,K$, and ζ, respectively. Hence the average cost optimal policy has the same threshold-type structure as the finite-horizon optimal policy and the discounted cost optimal policy.

The average-cost optimal policy can be found numerically by the value-iteration algorithm, the policy-iteration algorithm, and the linear programming formulation. We refer to Chapter 3 in Tijms [16] for a detailed description of these algorithms. To implement these algorithms we must specify the one-step transition probabilities and the one-step expected costs. For simplicity we suppose that $K = 2$. Let $a \in \{0, 1_\theta\}$, $0 \le \theta \le Q$, be the action that is selected when the system at a decision epoch is at state $(j, z_1, z_2) \in I$ with $z_1, z_2 \ge 0$. We assume that action $a = 0$ means that the vehicle goes directly to the next customer, while action 1_θ means that the vehicle goes to the depot to load θ items of product 1 and $Q - \theta$ items of product 2 and then it goes to the next customer $j+1$. Let $a \in \{2_\theta, 3_{\theta'}\}$, $0 \le \theta \le Q + \zeta$, $0 \le \theta' \le Q$, be the action that is selected when the system at a decision epoch is at state $(j, z_1, z_2) \in I$ with $\zeta = z_1^- + z_2^- < 0$. We assume that action 2_θ means that the vehicle goes to the depot to load the owed quantity $-\zeta$, θ items of product 1 and $Q + \zeta - \theta$ items of product 2, returns to customer j to satisfy the demand $-\zeta$, and then proceeds to the next customer $j+1$. Action $3_{\theta'}$ means that the vehicle goes to the depot to load the owed quantity $-\zeta$, returns to customer j to satisfy the demand $-\zeta$, makes a second trip to the depot where it loads θ' items of product 1 and $Q - \theta'$ items of product 2, and then proceeds to the next customer $j+1$. Let $p_{(j,z_1,z_2)(j+1,z_1',z_2')}(a)$ be the probability that the state at the next decision epoch will be $(j+1, z_1', z_2')$ if the present state is (j, z_1, z_2) and the action $a \in \{0, 1_\theta, 2_\theta, 3_{\theta'}\}$ is selected, and let $C((j, z_1, z_2), a)$ be the corresponding one-step expected cost. We give these quantities below.

If $z_1, z_2 \ge 0$, $(z_1, z_2) \in S$, then

$$p_{(j,z_1,z_2)(j+1,z_1',z_2')}(0) = \Pr(\xi_1^{j+1} = z_1 - z_1',\ \xi_2^{j+1} = z_2 - z_2'),$$

where $z_1' = z_1, z_1 - 1, \ldots, z_1 - Q$, $z_2' = z_2, z_2 - 1, \ldots, z_2 - Q$, and $(z_1 + z_2) - (z_1' + z_2') \le Q$.
If $z_1, z_2 \ge 0$, $(z_1, z_2) \in S$, $0 \le \theta \le Q$, then

$$p_{(j,z_1,z_2)(j+1,z_1',z_2')}(1_\theta) = \Pr(\xi_1^{j+1} = \theta - z_1',\ \xi_2^{j+1} = Q - \theta - z_2'),$$

where $z_1' = \theta, \theta - 1, \ldots, \theta - Q$, $z_2' = Q - \theta, Q - \theta - 1, \ldots, -\theta$, and $z_1' + z_2' \ge 0$.
If $\zeta = z_1^- + z_2^- < 0$, $(z_1, z_2) \in S$, $0 \le \theta \le Q + \zeta$, then

$$p_{(j,z_1,z_2)(j+1,z_1',z_2')}(2_\theta) = \Pr(\xi_1^{j+1} = \theta - z_1',\ \xi_2^{j+1} = Q + \zeta - \theta - z_2'),$$

where $z_1' = \theta, \theta - 1, \ldots, \theta - Q$, $z_2' = Q + \zeta - \theta, Q + \zeta - \theta - 1, \ldots, \zeta - \theta$, and $z_1' + z_2' \ge \zeta$.

If $\zeta = z_1^- + z_2^- < 0$, $(z_1, z_2) \in S$, $0 \le \theta' \le Q$, then

$$p_{(j,z_1,z_2)(j+1,z_1',z_2')}(3_{\theta'}) = \Pr(\xi_1^{j+1} = \theta' - z_1',\ \xi_2^{j+1} = Q - \theta' - z_2'),$$

where $z_1' = \theta', \theta' - 1, \ldots, \theta' - Q$, $z_2' = Q - \theta', Q - \theta' - 1, \ldots, -\theta'$, and $z_1' + z_2' \ge 0$.
If $z_1, z_2 \ge 0$, $(z_1, z_2) \in S$,

$$C((j, z_1, z_2), 0) = c_{j,j+1},$$
$$C((j, z_1, z_2), 1_\theta) = c_{j0} + c_{0,j+1}, \quad 0 \le \theta \le Q.$$

If $\zeta = z_1^- + z_2^- < 0$, $(z_1, z_2) \in S$,

$$C((j, z_1, z_2), 2_\theta) = 2c_{j0} + c_{j,j+1}, \quad 0 \le \theta \le Q + \zeta,$$
$$C((j, z_1, z_2), 3_{\theta'}) = 3c_{j0} + c_{0,j+1}, \quad 0 \le \theta' \le Q.$$

As illustration we present the following example.

Example 3: Suppose that $N = 7$, $Q = 10$, $K = 2$. We give below the symmetric matrix $C = (c_{ij})$, $0 \le i, j \le 7$, whose non-zero elements are the travel costs $c_{j,j+1}$ between customer $j \in \{1, \ldots, 7\}$ and customer $j + 1$ and the travel costs c_{j0} between customer $j \in \{1, \ldots, 7\}$ and the depot. We observe that these costs satisfy the triangle inequality.

$$C = \begin{pmatrix} 0 & 20 & 15 & 21 & 15 & 20 & 19 & 20 \\ 20 & 0 & 15 & 0 & 0 & 0 & 0 & 31 \\ 15 & 15 & 0 & 17 & 0 & 0 & 0 & 0 \\ 21 & 0 & 17 & 0 & 15 & 0 & 0 & 0 \\ 15 & 0 & 0 & 15 & 0 & 18 & 0 & 0 \\ 20 & 0 & 0 & 0 & 18 & 0 & 17 & 0 \\ 19 & 0 & 0 & 0 & 0 & 17 & 0 & 21 \\ 20 & 31 & 0 & 0 & 0 & 0 & 21 & 0 \end{pmatrix}$$

We assume that at each cycle the probabilities $\Pr(\xi_1^j = x,\ \xi_2^j = y)$, $x, y = 0, \ldots, 10$, for each customer $j \in \{1, \ldots, 7\}$ are the elements of the following 11×11 matrix P.

$$P = \begin{bmatrix} 0.0171 & 0.0255 & 0.0193 & 0.0287 & 0.0024 & 0.0178 & 0.0097 & 0.0216 & 0.0048 & 0.0033 & 0.0026 \\ 0.0043 & 0.0076 & 0.0148 & 0.0089 & 0.0017 & 0.0147 & 0.0165 & 0.0234 & 0.0258 & 0.0301 & 0 \\ 0.0125 & 0.0047 & 0.0291 & 0.0110 & 0.0237 & 0.0166 & 0.0004 & 0.0052 & 0.0141 & 0 & 0 \\ 0.0168 & 0.0001 & 0.0081 & 0.0081 & 0.0109 & 0.0260 & 0.0236 & 0.0244 & 0 & 0 & 0 \\ 0.0105 & 0.0188 & 0.0026 & 0.0311 & 0.0242 & 0.0250 & 0.0263 & 0 & 0 & 0 & 0 \\ 0.0061 & 0.0183 & 0.0119 & 0.0292 & 0.0051 & 0.0082 & 0 & 0 & 0 & 0 & 0 \\ 0.0072 & 0.0024 & 0.0256 & 0.0135 & 0.0080 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.0079 & 0.0172 & 0.0178 & 0.0041 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.0248 & 0.0205 & 0.0286 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.0138 & 0.0272 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.0282 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

It can be seen that, as expected, the sum of the elements of matrix P is equal to 1. Note also that $\Pr(\xi_1^j + \xi_2^j > 10) = 0$, as expected. The standard value-iteration algorithm does not converge. This is due to the periodicity (with period N) of all states of the system under any stationary policy. This problem can be circumvented by a perturbation of the one-step transition probabilities so that a transition from a state to itself with non-zero probability is allowed. Specifically, we take the following new one-step probabilities:

$$\widetilde{p}_{(j,z_1,z_2)(j+1,z_1',z_2')}(a) = \tau p_{(j,z_1,z_2)(j+1,z_1',z_2')}(a), \quad \widetilde{p}_{(j,z_1,z_2)(j,z_1,z_2)}(a) = 1 - \tau,$$

where τ is a constant such that $0 < \tau < 1$. A reasonable choice for the value of τ is 0.5. The perturbed model has the same average-cost optimal policy as the original model (see Tijms

TABLE 3. The Critical Numbers of the Average-Cost Optimal Policy

j	$s_j(0)$	$s_j(1)$	$s_j(2)$	$s_j(3)$	$s_j(4)$	$s_j(5)$	$s_j(6)$	$s_j(7)$	$s_j(8)$	$s_j(9)$	$s_j(10)$	s_j
1	1,11	0,10	0,8	0,6	0,5	0,5	0,4	0,3	0,3	0,2	0,1	−9,−1
2	6,11	3,10	2,9	2,8	1,6	1,6	1,5	0,4	0,3	0,2	0,1	−6,0
3	0,11	0,10	0,7	0,5	0,4	0,3	0,3	0,3	0,2	0,2	0,1	−2,−2
4	7,11	4,10	3,9	2,8	2,7	1,6	1,5	1,4	1,3	0,2	0,1	−5,0
5	2,11	1,10	0,9	0,7	0,5	0,4	0,3	0,3	0,3	0,2	0,1	−8,−1
6	4,11	3,10	2,9	1,8	1,7	1,6	1,5	0,4	0,3	0,2	0,1	−6,0
7	11,11	10,10	7,9	5,8	4,7	3,6	3,5	2,4	2,3	2,2	1,1	−3,0

[16], p. 209). We implemented the value-iteration algorithm in the perturbed model, which, as expected, converges to the optimal policy after 56 iterations. The average cost of the optimal policy is found to be 47.35. We also implemented the value-iteration algorithm in the perturbed model if the travel costs are given by the symmetric matrix $C' = (c'_{ij})$, $0 \le i, j \le 7$, where $c'_{ij} = c_{ij}$ for $i, j \in \{1, \dots, 7\}$, $c'_{0j} = 0.5c_{0j}$ for $j \in \{0, \dots, 7\}$ and $c'_{i0} = 0.5c_{i0}$ for $i \in \{0, \dots, 7\}$. Note that the costs c'_{ij}, $i, j \in \{1, \dots, 7\}$ satisfy the triangle inequality. In this case the algorithm converges to the optimal policy after 58 iterations and the minimum average cost is found to be 40.85. In each cell of Table 3 we present, for each customer $j = 1, \dots, 7$, the critical numbers $s_j(z_1)$, $z_1 = 0, \dots, Q$, and s_j of the optimal policy. The first number in each cell is the critical number if the travel costs are given by the symmetric matrix C, while the second number is the critical number if the travel costs are given by the symmetric matrix C'. Note that Part (ii) of Theorem 1 is confirmed numerically since $s_j(z_1)$ is non-increasing in $z_1 \in \{0, \dots, 10\}$ for $j \in \{1, \dots, 7\}$, if the travel costs are given by the elements of C or C'. In Table 3, we observe that the first number in each cell is smaller or equal to the second number. This is intuitively plausible since it seems preferable for the vehicle to return to the depot for replenishment if the travel costs between the customers and the depot are halved. In Figures 8 and 9, we present the optimal decision for each state $(z_1, z_2) \in S$ after the first visit to customer 4 and to customer 6 if the travel costs are the elements of matrix C.

The value-iteration algorithm also enables us to determine the optimal quantities of products 1 and 2 that are loaded in the vehicle when it returns to the depot for replenishment. For example, if the travel costs are the elements of matrix C, at state $(2, 0, 5)$ the optimal decision is to return to the depot to load five items of product 1 and five items of product 2. At state $(2, -6, 0)$, the optimal decision is to return to the depot to load eight items of product 1 and two items of product 2, to return to customer 2 to deliver the owed six items of product 1, and then proceed to customer 3.

4. CONCLUSIONS

In this paper, we proposed a different approach for the solution of a special capacitated VRP studied by Tatarakis and Minis [15]. In this problem, it is assumed that a single vehicle starts its route from a depot and delivers K different products to N customers according to a particular order. The demands of the customers for the products are stochastic and each customer's total demand for all products is less than or equal to the vehicle capacity. The vehicle has only one compartment in which all products are stored. We selected as decision epochs the epochs at which the vehicle visits for the first time each customer and has satisfied as much of the customer's demand as possible. We proposed a suitable dynamic programming algorithm for the determination of the policy that minimizes the total

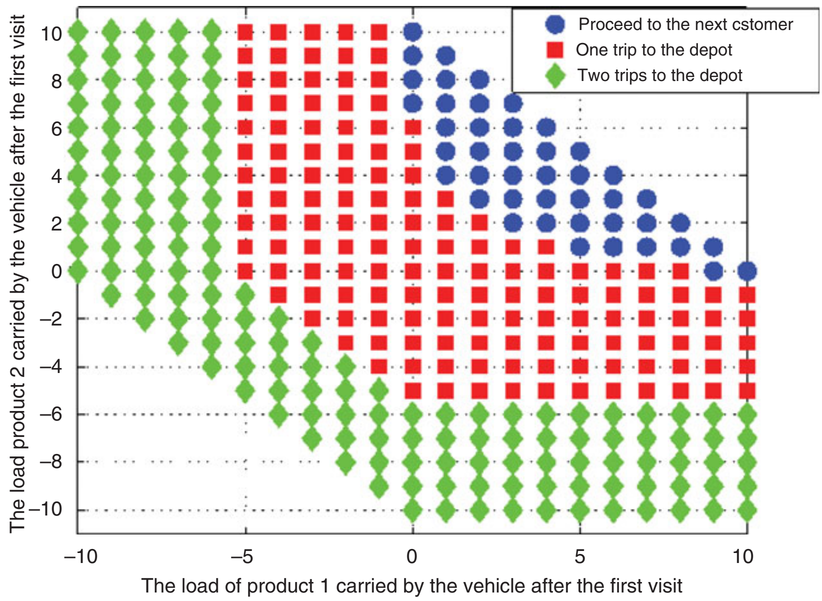

FIGURE 8. (Color online) The optimal decisions after the first visit to customer 4 if the travel costs are given by C.

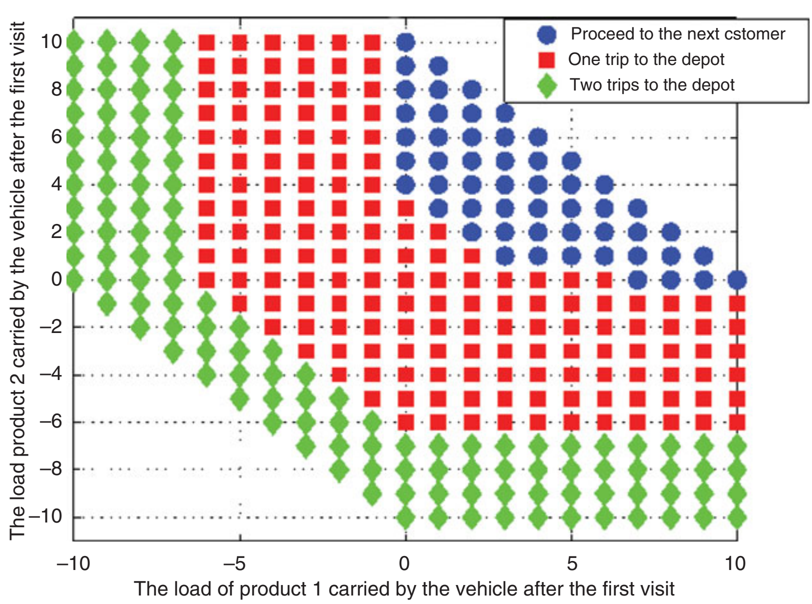

FIGURE 9. (Color online) The optimal decisions after the first visit to customer 6 if the travel costs are given by C.

expected cost for the service of all customers. It was proved that the optimal policy divides the set of all possible loads carried by the vehicle after the first visit to each customer into four disjoint subsets. If the load of the vehicle belongs to the first subset, then the optimal decision is to proceed to the next customer. If it belongs to the second subset, then it is optimal to go to the depot for restocking, and then to proceed to the next customer. If it belongs to the third subset, then it is optimal to go to the depot for restocking, to return to the customer to satisfy the owed quantity, and then to proceed to the next customer. If it belongs to the fourth subset, then it is optimal to go to the depot to restock the owed quantity, to return to the customer to deliver the owed quantity, to make a second trip to the depot for restocking, and then go to the next customer. Taking into account the structure of the optimal policy, we developed a special-purpose dynamic programming algorithm that finds the optimal policy. Numerical examples provide strong evidence that this algorithm is faster than the initial algorithm.

We also considered a corresponding infinite-horizon problem in which the vehicle after the service of the last customer continues to serve the customers periodically with the same customer order. The demands of the customers for each product are renewed in each cycle and follow the same distributions. The decision epochs are again the epochs at which the vehicle arrives at a customer and has satisfied as much of the customer's demand as possible. The times between decision epochs are assumed to be constant. It is proved that the discounted cost optimal policy and the average-cost optimal policy have the same structure as the finite-horizon optimal policy.

A subject for future research could be to investigate the structure of the optimal policy for a more general problem in which customers are not served according to a particular sequence.

Acknowledgements

We would like to thank an anonymous referee whose comments led to an improvement of the paper. We would also like to thank Christos Tarantilis for helpful discussions.

References

1. Bather, J. (2000). *Decision theory: an introduction to dynamic programming and sequential decisions*. Chichester: Wiley.
2. Clarke, G. & Wright, J.V. (1964). Scheduling of vehicles from a control depot to a number of delivery points. *Operations Research* 12: 568–581.
3. Cordeau, J.-F., Laporte, G., Savelsbergh, M.W.P., & Vigo, D. (2007). Vehicle routing. In C. Barnhart & G. Laporte (eds.), *Transportation, Handbook in Operations Research and Management Science*, Vol. 14, Amsterdam: Elsevier, pp. 367–428.
4. Dantzig, G. & Ramser, R. (1959). The truck dispatching problem. *Management Science* 6: 80–91.
5. Kyriakidis, E.G. & Dimitrakos, T.D. (2008). Single vehicle routing problem with a predefined customer sequence and stochastic continuous demands. *Mathematical Scientist* 33: 148–152.
6. Laporte, G. (1992). The vehicle routing problem: an overview of exact and approximate algorithms. *European Journal of Operational Research* 59: 345–358.
7. Laporte, G. (2009). Fifty years of vehicle routing. *Transportation Science* 43: 408–416.
8. Liong, C.Y., Wan Rosmanira, I., Khairuddin, O., & Zirour, M. (2008). Vehicle routing problem: models and solutions. *Journal of Quality Measurement and Analysis* 4: 205–218.
9. Pandelis, D.G., Kyriakidis, E.G., & Dimitrakos, T.D. (2012). Single vehicle routing problems with a predefined customer sequence, compartmentalized load and stochastic demands. *European Journal of Operational Research* 217: 324–332.
10. Rembold, U., Blume, C., & Dillmann, R. (1985). *Computer integrated manufacturing technology and systems*. New York: Marcel Dekker.
11. Ross, S.M. (1992). *Applied probability models with optimization applications*. New York: Dover.
12. Secomandi, N. & Margot, F. (2009). Reoptimization approaches for the vehicle-routing problem with stochastic demands. *Operations Research* 57: 214–230.
13. Simchi-Levi, D., Chen, X., & Bramel, J. (2005). *The logic of logistics. Theory, algorithms and applications for logistics and supply chain management*. New York: Springer.

14. Tatarakis, A. (2007). *A class of single vehicle routing problems with predefined customer sequence and depot returns*. Ph.D. thesis, Department of Financial and Management Engineering, University of the Aegean, Chios, Greece. Available from http://fidelity.fme.aegean.gr/deopsys/files/A.Tatarakis%PhD.pdf.
15. Tatarakis, A. & Minis, I. (2009). Stochastic single vehicle routing with a predefined customer sequence and multiple depot returns. *European Journal of Operational Research* 197: 557–571.
16. Tijms, H.C. (1994). *Stochastic models: an algorithmic approach*. New York: Wiley.
17. Toth, P. & Vigo, D. (Eds.) (2002). *The vehicle routing problem*. Philadelphia, PA: Siam.
18. Tsirimpas, P., Tatarakis, A., Minis, I., & Kyriakidis, E.G. (2008). Single vehicle routing with a predefined customer sequence and multiple depot returns. *European Journal of Operational Research* 187: 483–495.
19. Yang, W.-H., Mathur, K., & Ballou, R.H. (2000). Stochastic vehicle routing problem with restocking. *Transportation Science* 34: 99–112.