International Journal of Systems Science: Operations & Logistics

ISSN: 2330-2674 (Print) 2330-2682 (Online) Journal homepage: https://www.tandfonline.com/loi/tsyb20

Two-compartment stochastic single vehicle routing problems with simultaneous pickups and deliveries from *N* ordered customers

Constantinos C. Karamatsoukis, Epaminondas G. Kyriakidis & Theodosis D. Dimitrakos

To cite this article: Constantinos C. Karamatsoukis, Epaminondas G. Kyriakidis & Theodosis D. Dimitrakos (2020): Two-compartment stochastic single vehicle routing problems with simultaneous pickups and deliveries from *N* ordered customers, International Journal of Systems Science: Operations & Logistics, DOI: 10.1080/23302674.2020.1720034

To link to this article: https://doi.org/10.1080/23302674.2020.1720034

Published online: 30 Jan 2020.

INTERNATIONAL JOURNAL OF SYSTEMS SCIENCE: OPERATIONS & LOGISTICS
https://doi.org/10.1080/23302674.2020.1720034

Two-compartment stochastic single vehicle routing problems with simultaneous pickups and deliveries from *N* ordered customers

Constantinos C. Karamatsoukis[a], Epaminondas G. Kyriakidis[b] and Theodosis D. Dimitrakos[c]

[a]Department of Military Sciences, Hellenic Military Academy, Attica, Greece; [b]Department of Statistics, Athens University of Economics and Business, Athens, Greece; [c]Department of Mathematics, University of the Aegean, Samos, Greece

ABSTRACT

We consider a vehicle with two compartments that starts its route from a depot and visits N ordered customers in order to deliver new (or fresh or useful) products and to collect old (or expired or useless) products. The new products are placed in Compartment 1 and the old products are placed in Compartment 2. The quantity of new products that each customer demands and the quantity of old products that each customer returns are random variables with known joint distribution. The vehicle is allowed during its route to return to the depot in order to replenish Compartment 1 with new products and to unload the old products from Compartment 2. Under a suitable cost structure, it is possible to find the optimal restocking strategy by implementing a suitable dynamic programming algorithm. We also find the optimal restocking strategy for the corresponding infinite-time horizon problem. We further consider the same problem without the assumption that the customers are ordered. Numerical experiments indicate that the optimal restocking strategy can be computed for values of N up to 9.

ARTICLE HISTORY
Received 20 August 2019
Accepted 18 January 2020

KEYWORDS
Routing problem; two-compartment vehicle; stochastic demands; dynamic programming

1. Introduction

A vehicle routing problem that has been the topic of significant research is the problem of finding the optimal routing of a single vehicle with finite capacity that starts its route from a depot and serves N ordered customers. During its route, the vehicle may return to the depot for replenishment or for unloading. Various versions of this problem have been analysed. In some versions of this problem (see Dikas, Minis, & Mamasis, 2016; Tsirimpas, Tatarakis, Minis, & Kyriakidis, 2008) the demands of the customers were assumed to be deterministic and in some other versions (see Dimitrakos & Kyriakidis, 2015; Kyriakidis & Dimitrakos, 2019; Kyriakidis, Dimitrakos, & Karamatsoukis, 2019; Pandelis, Kyriakidis, & Dimitrakos, 2012, 2013; Tatarakis & Minis, 2009; Yang, Mathur, & Ballou, 2000; Yee & Golden, 1980; Zhang, Lam, & Chen, 2016; Zhu & Sheu, 2018) the demands of the customers were assumed to be stochastic. The assumption that the customers are serviced according to a particular order makes possible the development of suitable dynamic programming formulations for these problems. In some problems (see e.g. Kyriakidis et al., 2019; Pandelis, Karamatsoukis, & Kyriakidis, 2013) the instants at which the vehicle arrives for the first time at each customer's location were chosen as decision epochs for choosing some action, while, in some other problems (see e.g. Pandelis et al., 2012; Tatarakis & Minis, 2009; Yang et al., 2000; Yee & Golden, 1980) the actions are selected as soon as a customer has been served. In most of these problems it was shown that the optimal restocking strategy, for each customer, is of threshold type, i.e. it is characterised by some critical integers. For example (see Section 3 in Yang et al., 2000), if the remaining load in the vehicle after the completion of the service of a customer exceeds a critical level, then the optimal action is to proceed to the next customer, whereas, if it is smaller than the critical level then the optimal action is to return to the depot for replenishment and then go to the next customer. By taking into account the structure of the optimal restocking strategy, in many cases (see e.g. Dimitrakos & Kyriakidis, 2015) it is possible to design a fast special-purpose dynamic programming algorithm that operates only on the restocking strategies that have the special structure and leads to the optimal restocking strategy. It is noteworthy that in the recent literature there is a significant number of papers that consider vehicle routing problems with stochastic demands and restocking policies (see e.g. Bertazzi & Secomandi, 2018a, 2018b; Florio,

CONTACT Theodosis D. Dimitrakos dimitheo@aegean.gr Department of Mathematics, University of the Aegean, Karlovassi, Samos 83200, Greece

Hartl, & Minner, 2018; Louveaux & Juan-José Salazar-González, 2018; Salavati-Khoshghal, Gendreau, Jabali, & Rei, 2019).

Special attention has been paid to vehicle routing problems with ordered customers and compartmentalised load. Tsirimpas et al. (2008) assumed that the demands of the ordered customers are deterministic and investigated the case of multiple-product deliveries if each product is stored in its own compartment in the vehicle. The optimal restocking strategy was found by implementing a suitable dynamic programming algorithm. Tatarakis and Minis (2009) investigated the above problem when the demands of the ordered customers are discrete random variables. When the vehicle has two compartments, a structural result for the optimal restocking strategy was obtained. This result was generalised by Pandelis et al. (2012) for the case in which the vehicle has $K \geq 2$ compartments. In the present paper, we consider a vehicle that has two compartments with finite capacities and visits N customers according to a predefined sequence $1 \rightarrow 2 \rightarrow \ldots \rightarrow N$. New (or fresh or useful) and old (or expired or useless) products are placed in Compartment 1 and Compartment 2, respectively. The vehicle starts its route from a depot. During its route, new products are delivered to the customers and old products are collected from them. We assume that, for each customer, the quantity that is delivered and the quantity that is collected are random variables with known joint distribution. The actual quantities that are delivered and collected are disclosed only when the vehicle arrives at the customer's location. The vehicle may interrupt its route and return to the depot to restock with new products and unload the old products. The total cost for servicing all customers consists of travel costs between consecutive customers and travel costs between customers and the depot. By selecting suitable decision epochs, it is possible to construct a dynamic programming algorithm that computes the minimum expected total cost and determines the optimal actions for the routing of the vehicle. It is possible to prove that, for each customer, the optimal restocking strategy is of threshold type, i.e. the selection of the optimal actions depends on some critical numbers. This structural property of the optimal restocking strategy permits us to design an efficient special-purpose dynamic programming algorithm that is restricted to restocking strategies with this structure.

A practical application of this vehicle routing problem could be the routing of a vehicle that visits N stores in order to deliver and collect fresh and expired milk (or ice-cream or vegetables or fruit). The vehicle is separated into two compartments. The bottles with fresh milk are placed in the first compartment in low temperature while the bottles with expired milk are placed in the second compartment in normal temperature. Another example could be the delivery and collection of two different building materials (e.g. lime and pebble) that are placed in the suitable compartments of the vehicle.

The main contribution of the present work is that an improved dynamic programming formulation is presented for the two-compartment problem in comparison with the dynamic programming formulations presented in Tatarakis and Minis (2009) and in Pandelis et al. (2012). In Tatarakis and Minis and in Pandelis et al. the possible actions (which are only two) are selected as soon as the service of a customer is completed, while in the present work the possible actions (which are six) are selected when the vehicle visits each customer for the first time and the maximum possible service has been offered. The dynamic programming formulation that we adopt in the present work leads to a detailed characterisation of the optimal restocking strategy with practical usefulness.

The rest of the paper is organised as follows. In Section 2, we give a dynamic programming formulation of the problem when the quantities that are given to and collected from each customer are discrete random variables. It is shown that the structure of the optimal restocking strategy is of threshold-type and a fast special-purpose dynamic programming algorithm is designed. In Section 3, we investigate the problem of optimal routing of the vehicle when the quantities that are given to and collected from each customer are continuous random variables. In Section 4, we give some numerical results that verify our analytical results. In Section 5, we investigate the corresponding infinite-time horizon problem when the service of the customers does not stop when the last customer has been serviced but continues indefinitely with the same customer order. In Section 6, we consider a generalisation of the problem without the assumption that the customers are ordered. The conclusions of the paper are given in the last section.

2. The problem and the optimal restocking strategy

2.1. The problem

We consider a network consisting of a set of nodes $V = \{0, 1, \ldots, N\}$ with node 0 denoting a depot and the nodes $1, 2, \ldots, N$ corresponding to ordered customers. A vehicle starts its route from the depot and serves the customers according to the sequence $1 \rightarrow 2 \rightarrow \ldots \rightarrow N$. The vehicle consists of two compartments, Compartment 1 and Compartment 2, with finite capacities Q_1 and Q_2, respectively. The vehicle delivers new (or fresh or useful) items of a product to the customers. The vehicle also

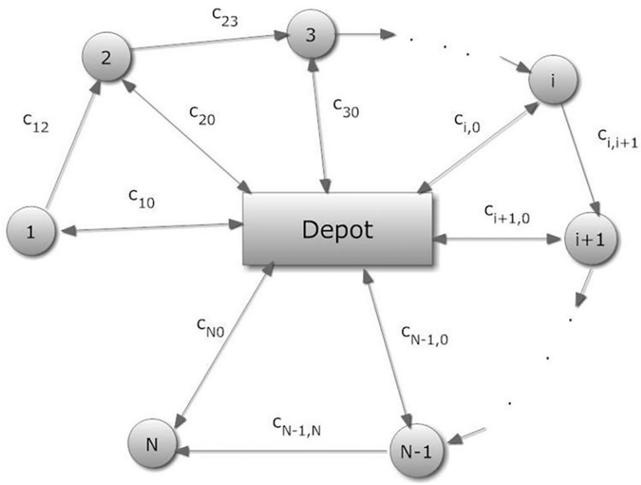

Figure 1. The customer network.

collects old (or expired or useless) items of the same or of a different product. We assume that (i) the items that are delivered are placed in Compartment 1, (ii) the items that are collected are placed in Compartment 2, (iii) the items that are delivered have the same size, (iv) the items that are collected have the same size, (v) the size of the items that are delivered may be different from the size of the items that are collected. Let $c_{j,j+1}, j = 1, \dots, N-1$, be the travel cost from customer j to customer $j+1$. Let also c_{j0} and $c_{0j}, j = 1, \dots, N$ be the travel cost from customer j to the depot and the travel cost from the depot to customer j, respectively. It is plausible that these costs satisfy the triangle inequality, i.e.

$$c_{j,j+1} \leq c_{j0} + c_{0,j+1}, \quad j = 1, \dots, N-1.$$

It is assumed that, for each customer $j = 1, \dots, N$, the quantity $\xi_j \in \{0, \dots, Q_1\}$ of new items that he/she demands and the quantity $\psi_j \in \{0, \dots, Q_2\}$ of old items that he/she returns are discrete random variables. The joint distribution of ξ_j and ψ_j is known. The actual quantity of new items that each customer demands and the actual quantity of old items that each customer returns are disclosed only when the vehicle arrives at the customer's location. The vehicle may interrupt its route and return to the depot to replenish Compartment 1 with new items and to unload the old items that are stored in Compartment 2. After servicing the last customer N, the vehicle returns to the depot and terminates its route. The customer network is presented in Figure 1.

The total cost consists of travel costs between consecutive customers and between customers and the depot. The problem is to find the restocking strategy that minimises the expected total cost for servicing all customers.

Suppose that the vehicle arrives at customer's $j \in \{1, \dots, N\}$ location. The actual demand for new items and the actual quantity of old items that are returned are revealed. Let (q, s) be the state of the process after the first visit to customer j and after the maximum possible quantity of new items has been delivered and the maximum possible quantity of old items has been collected. The number q is the quantity of new items that remain in Compartment 1 and the number s is the empty space in Compartment 2, i.e. the number of old items that can be placed in Compartment 2. Negative values of q and s denote unsatisfied demand for new items and lack of empty space in Compartment 2 for old items, respectively. There are four cases:

Case 1. $0 \leq q \leq Q_1, 0 \leq s \leq Q_2$. In this case customer j has been serviced completely, q new items remain in Compartment 1 of the vehicle and there is empty space for s old items in Compartment 2.

Case 2. $-Q_1 \leq q < 0, 0 \leq s \leq Q_2$. In this case there is unsatisfied demand for $-q$ new items and there is empty space for s old items in Compartment 2.

Case 3. $0 \leq q \leq Q_1, -Q_2 \leq s < 0$. In this case q new items remain in Compartment 1 of the vehicle and there is lack of empty space for $-s$ old items in Compartment 2.

Case 4. $-Q_1 \leq q < 0, -Q_2 \leq s < 0$. In this case there is unsatisfied demand for $-q$ new items and there is lack of empty space for $-s$ old items in Compartment 2.

Suppose $j \in \{1, \dots, N-1\}$.

In Case 1, the possible actions are Action 1 and Action 2. Action 1 means that the vehicle proceeds to customer $j+1$ and Action 2 means that the vehicle goes to the depot to restock Compartment 1 with $Q_1 - q$ new items and unload all old items from Compartment 2 and then goes to customer $j+1$. In Case 2 the possible actions are Action 3 and Action 4. Action 3 means that the vehicle goes to the depot to restock Compartment 1 with Q_1 new items and unload all s old items from Compartment 2, returns to customer j to deliver $-q$ owed new items and then proceeds to customer $j+1$. Action 4 means that the vehicle goes to the depot to restock its Compartment 1 with $-q$ new items and to unload all s old items from Compartment 2, returns to customer j to deliver $-q$ owed new items, makes a second trip to the depot to load Compartment 1 with Q_1 new items and then proceeds to customer $j+1$. In Case 3 the possible actions are Action 5 and Action 4. Action 5 means that the vehicle goes to the depot to unload Q_2 old items from Compartment 2 and restock Compartment 1 to its full capacity, returns to customer j to load in Compartment 2 the remaining $-s$ old items in Compartment 2 and then goes to customer $j+1$. Action 4 (in this case) means that the vehicle goes to the depot to unload Q_2 old items from Compartment 2, returns to customer j to load in Compartment 2 the remaining $-s$ old items, makes a second trip to the depot to restock Compartment 1 with Q_1 new items and to empty Compartment 2 and then goes to customer $j+1$. In Case 4 the possible actions are Action 6 and Action 4. Action 6 means that the vehicle makes one trip to the depot to restock Compartment 1 with Q_1 new items and to empty Compartment 2, returns to customer j to deliver $-q$ owed new items and to collect $-s$ remaining old items and then proceeds to customer $j+1$. Action 4 (in this case) means that the vehicle goes to the depot to replenish Compartment 1 with $-q$ new items and to empty Compartment 2, returns to customer j to deliver $-q$ owed new items and to load in Compartment 2 $-s$ remaining old items, makes a second trip to the depot to load Compartment 1 to its full capacity with new items and to empty Compartment 2 and then proceeds to customer $j+1$. It is assumed that if Action $a \in \{3, 4, 5, 6\}$ is selected, there is no extra demand for new items that must be delivered or extra old items that must be collected when the vehicle returns to customer j, i.e. ξ_j and ψ_j remain unaltered.

Suppose that $j = N$.

In Case 1 the only possible action for the vehicle is to return to the depot to terminate its route. In Cases 2–4 the only possible action is to go to the depot to load the owed quantity of new items or/and to empty its Compartment 2, returns to customer N to deliver the owed quantity of new items or/and to collect the remaining quantity of old items and then returns to the depot to terminate its route.

2.2. Dynamic programming formulation

It is possible to construct a suitable dynamic programming algorithm for the determination of the optimal restocking strategy. Let $f_j(q,s)$ denote the minimum expected future cost from the first visit of the vehicle to customer $j \in \{1,\ldots,N\}$ until the end of the route, where (q,s) is the state of the process that has been defined above. For $j \in \{1,\ldots,N-1\}$ we give below the dynamic programming Equations (1–4) that correspond to Cases 1–4.

If $0 \le q \le Q_1, 0 \le s \le Q_2$, then

$$f_j(q,s) = \min\{A_j(q,s), B_j\}, \tag{1}$$

where

$$A_j(q,s) = c_{j,j+1} + Ef_{j+1}(q - \xi_{j+1}, s - \psi_{j+1}),$$
$$B_j = c_{j0} + c_{0,j+1} + Ef_{j+1}(Q_1 - \xi_{j+1}, Q_2 - \psi_{j+1}).$$

If $-Q_1 \le q < 0, 0 \le s \le Q_2$, then

$$f_j(q,s) = 2c_{j0} + \min\{C_j(q), B_j\}, \tag{2}$$

where

$$C_j(q) = c_{j,j+1} + Ef_{j+1}(Q_1 + q - \xi_{j+1}, Q_2 - \psi_{j+1}).$$

If $0 \le q \le Q_1, -Q_2 \le s < 0$, then

$$f_j(q,s) = 2c_{j0} + \min\{D_j(s), B_j\}, \tag{3}$$

where

$$D_j(s) = c_{j,j+1} + Ef_{j+1}(Q_1 - \xi_{j+1}, Q_2 + s - \psi_{j+1}).$$

If $-Q_1 \le q < 0, -Q_2 \le s < 0$, then

$$f_j(q,s) = 2c_{j0} + \min\{E_j(q,s), B_j\}, \tag{4}$$

where

$$E_j(q,s) = c_{j,j+1} + Ef_{j+1}(Q_1 + q - \xi_{j+1}, Q_2 + s - \psi_{j+1}).$$

The boundary condition is given by the following equation

$$f_N(q,s) = c_{N0} + 1(q^- + s^- < 0) \cdot 2c_{N0}, \tag{5}$$

where 1 is the indicator function, $q^- = \min\{q,0\}$ and $s^- = \min\{s,0\}$ The minimum total expected cost during a visit cycle is equal to

$$f_0 = c_{01} + Ef_1(Q_1 - \xi_1, Q_2 - \psi_1).$$

In the above equations the expected values are taken with respect to the random variables ξ_j and ψ_j, $j = 1,\ldots,N$. The terms $A_j(q,s)$ and B_j in the right-hand-side of Equation (1) correspond to Action 1 and Action 2, respectively. The terms $C_j(q)$ and B_j in the right-hand-side of Equation (2) correspond to Action 3 and Action 4, respectively. The terms $D_j(s)$ and B_j in the right-hand-side of Equation (3) correspond to Action 5 and Action 4, respectively. The terms $E_j(q,s)$ and B_j in Equation (4) correspond to Action 6 and Action 4, respectively. The structure of the optimal restocking strategy is given in Theorem 1. Lemma 1 below will be used in the proof of Theorem 1.

2.3. Structure of the optimal restocking strategy

Lemma 1: *$f_j(q,s), j = 1,\ldots,N$ is non-increasing with respect to q and s.*

Proof: The proof is by induction on j. From Equation (5) it can be seen that $f_N(q,s)$ is non-increasing in q and s. Assuming that $f_{j+1}(q,s), 2 \le j \le N$, is non-increasing in q and s we will show that $f_j(q,s)$ is non-increasing in q and s. In view of induction hypothesis and Equations (1–4), to prove that $f_j(q,s)$ is non-increasing in its arguments, it is enough to show that

$$f_j(0,s) \le f_j(-1,s), 0 \le s \le Q_2, \tag{6}$$

$$f_j(q,0) \le f_j(q,-1), 0 \le q \le Q_1, \tag{7}$$

$$f_j(0,s) \le f_j(-1,s), -Q_2 \le s \le -1, \tag{8}$$

$$f_j(q,0) \le f_j(q,-1), -Q_1 \le q \le -1. \tag{9}$$

From Equation (1) and the triangle inequality we have for $s \in \{0,\dots,Q_2\}$

$$\begin{aligned} f_j(0,s) &\le c_{j0} + c_{0,j+1} + Ef_{j+1}(Q_1 - \xi_{j+1}, Q_2 - \psi_{j+1}) \\ &\le 2c_{j0} + c_{j,j+1} + Ef_{j+1}(Q_1 - \xi_{j+1}, Q_2 - \psi_{j+1}) \\ &= H_j. \end{aligned} \tag{10}$$

Note that

$$\begin{aligned} &H_j - 2c_{j0} - C_j(-1) \\ &\quad = Ef_{j+1}(Q_1 - \xi_{j+1}, Q_2 - \psi_{j+1}) \\ &\qquad - Ef_{j+1}(Q_1 - 1 - \xi_{j+1}, Q_2 - \psi_{j+1}) \le 0, \end{aligned} \tag{11}$$

where the above inequality follows from the induction hypothesis. From (10) and (11) we have that

$$f_j(0,s) \le H_j \le 2c_{j0} + C_j(-1). \tag{12}$$

From (1) it is immediate that

$$f_j(0,s) \le B_j \le 2c_{j0} + B_j. \tag{13}$$

Relations (12) and (13) imply that

$$f_j(0,s) \le 2c_{j0} + \min\{C_j(-1), B_j\},$$

which is (6). Inequality (7) for $q = 0,\dots,Q_1$, is equivalent to

$$\min\{A_j(q,0), B_j\} \le 2c_{j0} + \min\{D_j(-1), B_j\},$$

or, equivalently,

$$\begin{aligned} &\min\{c_{j,j+1} + Ef_{j+1}(q - \xi_{j+1}, -\psi_{j+1}), B_j\} \le 2c_{j0} \\ &\quad + \min\{c_{j,j+1} + Ef_{j+1}(Q_1 - \xi_{j+1}, Q_2 - 1 - \psi_{j+1}), B_j\}. \end{aligned}$$

The above inequality holds if

$$B_j \le 2c_{j0} + c_{j,j+1} + Ef_{j+1}(Q_1 - \xi_{j+1}, Q_2 - 1 - \psi_{j+1}),$$

or, equivalently,

$$\begin{aligned} &c_{0,j+1} + Ef_{j+1}(Q_1 - \xi_{j+1}, Q_2 - \psi_{j+1}) \\ &\quad \le c_{j0} + c_{j,j+1} + Ef_{j+1}(Q_1 - \xi_{j+1}, Q_2 - 1, \psi_{j+1}), \end{aligned}$$

which, in view of the induction hypothesis and the triangle inequality, is true. Inequality (8), for $s \in \{-Q_2,\dots,-1\}$, is equivalent to

$$\begin{aligned} &\min\{c_{j,j+1} + Ef_{j+1}(Q_1 - \xi_{j+1}, Q_2 + s - \psi_{j+1}), B_j\} \\ &\quad \le \min\{c_{j,j+1} + Ef_{j+1}(Q_1 - 1 - \xi_{j+1}, \\ &\qquad Q_2 + s - \psi_{j+1}), B_j\}. \end{aligned}$$

In view of the induction hypothesis, the above inequality is true. Inequality (9), for $q \in \{-Q_1,\dots,-1\}$, is equivalent to

$$\min\{C_j(q), B_j\} \le \min\{E_j(q,-1), B_j\},$$

or, equivalently,

$$\begin{aligned} &\min\{c_{j,j+1} + Ef_{j+1}(Q_1 + q - \xi_{j+1}, Q_2 - \psi_{j+1}), B_j\} \\ &\quad \le \min\{c_{j,j+1} + Ef_{j+1}(Q_1 + q - \xi_{j+1}, \\ &\qquad Q_2 - 1 - \psi_{j+1}), B_j)\}. \end{aligned}$$

In view of the induction hypothesis, the above inequality is true.

In the following theorem we present the optimal restocking strategy after the first visit of the vehicle at customer's $j \in \{1,\dots,N-1\}$ location.

Theorem 1: *For each customer $j \in \{1,\dots,N-1\}$ the structure of the optimal restocking strategy is described in the following four cases:*

(i) *For each $q \in \{0,\dots,Q_1\}$ there exists an integer $k_{1j}(q) \ge 0$ such that it is optimal for the vehicle to proceed to customer $j+1$ if and only if $s \ge k_{1j}(q)$. Moreover, $k_{1j}(q)$ is non-increasing in q.*
(ii) *There exists $s_j \le 0$ such that it is optimal for the vehicle to make two trips to the depot when $q \in \{0,\dots,Q_1\}$ and $s < s_j$.*
(iii) *There exists $q_j \le 0$ such that it is optimal for the vehicle to make two trips to the depot when $s \in \{0,\dots,Q_2\}$ and $q < q_j$.*
(iv) *For each $s \in \{-1,\dots,-Q_2\}$ there exists an integer $k_{2j}(s) < 0$ such that it is optimal for the vehicle to make two trips to the depot if and only if $q \le k_{2j}(s)$. Moreover, $k_{2j}(s)$ is non-increasing in s.*

Proof: From Lemma 1 it follows that $A_j(q,s)$ is non-increasing in q and s, respectively. Part (i) is a direct consequence of this result. From Lemma 1 it follows that $C_j(q)$ and $D_j(s)$ are non-increasing in q and s, respectively. Part (ii) and Part (iii) are direct consequences of this result. From Lemma 1 it follows that $E_j(q,s)$ is non-increasing in q and s. Part (iv) is a direct consequence of this result. ■

Remark 1: The decision epochs for the above vehicle routing problem are the instants at which the vehicle arrives for the first time at each customer's location and the maximum possible service has been offered. It is possible to assume that the decision epochs are the instants at which each customer has been serviced completely. This choice leads to a different dynamic programming formulation with only two possible actions: (i) proceed to next customer and (ii) go to the depot to replenish Compartment 1 with new items and to unload old items from Compartment 2 and then go to next customer. Following the line of proof that is used in Pandelis et al. (2012) in stochastic vehicle routing problems with compartmentalised load and deliveries, it is possible to prove that, for a given quantity of old items in Compartment 2, it is optimal to choose the action of proceeding to next customer if the quantity of new items in Compartment 1 exceeds some critical value. However, the dynamic programming formulation that we have adopted in the present work led us to the above Theorem 1 that provides a more detailed characterisation of the structure of the optimal restocking strategy with six possible actions and with greater practical usefulness.

2.4. Special-purpose dynamic programming algorithm

In view of the above theorem, the optimal restocking strategy, i.e. the critical integers $k_{1j}(q)$, $q = 0, \ldots, Q_1$, $k_{2j}(s)$, $s = -Q_2, \ldots, -1$, q_j and s_j for each customer $j \in \{1, \ldots, N-1\}$ can be found by the following special-purpose dynamic programming algorithm.

Algorithm for the determination of the critical integers $k_{1j}(q)$, $q = 0, \ldots, Q_1$, $k_{2j}(s)$, $s = -Q_2, \ldots, -1$, q_j and s_j for each customer $j = 1, \ldots, N-1$

Step 0 Set $f_N(q,s) = c_{N0} + 2c_{N0} \cdot 1(q^- + s^-)$, $|q| \le Q_1, |s| \le Q_2$. Set $j = N-1$.

Step 1 (Determination of the critical integers $k_{1j}(q), q = 0, \ldots, Q_1$)
- Compute B_j.
- For $q = 0, 1, \ldots, Q_1$, do the following:
- For $s = Q_2, Q_2 - 1, \ldots$ compute $A_j(q,s)$ until $A_j(q,s) > B_j$ or $s = -1$.
- Set $k_{1j}(q) = s + 1$.
- Set $f_j(q,s) = B_j$, for $s \in \{0, \ldots, k_{1j}(q) - 1\}$ and $f_j(q,s) = A_j(q,s)$, for $s \in \{k_{1j}(q), \ldots, Q_2\}$.

Step 2 (Determination of critical integers $k_{2j}(s), s = -Q_2, \ldots, -1$)
- For $s = -1, -2, \ldots, -Q_2$, do the following:
- For $q = -Q_1, -Q_1 + 1, \ldots$ compute $E_j(q,s)$ until $E_j(q,s) < B_j$ or $q = 0$.
- Set $k_{2j}(s) = q - 1$.
- Set $f_j(q,s) = 2c_{j0} + B_j, q \in \{-Q_2, \ldots, k_{2j}(s)\}$ and $f_j(q,s) = 2c_{j0} + E_j(q,s)$, $q \in \{k_{2j}(s) - 1, \ldots, -1\}$.

Step 3 (Determination of the critical integers q_j)
- For $q = -Q_1, -Q_1 + 1, \ldots$ compute $C_j(q)$ until $C_j(q) < B_j$ or $q = 0$.
- Set $q_j = q$.
- Set $f_j(q,s) = 2c_{j0} + B_j$, for $q \in \{-Q_1, \ldots, q_j - 1\}$ and $s \in \{0, \ldots, Q_2\}$.
- Set $f_j(q,s) = 2c_{j0} + C_j(q)$, for $q \in \{q_j, \ldots, -1\}$ and $s \in \{0, \ldots, Q_2\}$.

Step 4 (Determination of critical integers s_j)
- For $s = -Q_2, -Q_2 + 1, \ldots$ compute $D_j(s)$ until $D_j(s) < B_j$ or $s = 0$.
- Set $s_j = s$.
- Set $f_j(q,s) = 2c_{j0} + B_j$, for $s \in \{-Q_2, \ldots, s_j - 1\}$ and $q \in \{0, \ldots, Q_1\}$.
- Set $f_j(q,s) = 2c_{j0} + D_j(s)$, for $s \in \{s_j, \ldots, -1\}$ and $q \in \{0, \ldots, Q_1\}$.

Step 5 Set $j = j - 1$. If $j \ge 1$, go to Step 1. Otherwise, stop.

The above special-purpose dynamic programming algorithm is based on the structure of the optimal restocking strategy described in Theorem 1. The complexity of this algorithm can be calculated by considering Definition 7.1 in Sipser (2013) and is found to be $O(NQ_1^2Q_2^2)$. It is more efficient than the initial dynamic programming algorithm since it requires less computations. For example, for $j \in \{1, \ldots, N-1\}$, in Case 1, the quantities $A_j(q,s)$, for (q,s) such that $q \in \{0, \ldots, Q_1\}$ and $s \in \{0, \ldots, k_{1j}(q) - 1\}$, are not computed, while these quantities are computed in the initial dynamic programming algorithm, that is based in Equations (1–5). Thus, the special-purpose dynamic programming algorithm requires smaller computation time than the computation time required by the initial dynamic programming algorithm. In Section 4, the computation times of theses algorithms will be compared in a numerical example.

3. The problem when the quantities that are delivered and collected are continuous random variables

3.1. The optimal restocking strategy when ξ_j and $\psi_j, j = 1, \ldots, N$ are continuous random variables

We modify the problem that we introduced in Section 2 by assuming that, for each customer $j \in \{1, \ldots, N\}$ (i) the quantity ξ_j of new items that he/she demands is a continuous random variable which takes values in the interval $[0, Q_1]$ and (ii) the quantity ψ_j of old items that he/she returns is a continuous random variable which take values in the interval $[0, Q_2]$. The joint distribution of ξ_j and

ψ_j is known. A realistic situation with continuous quantities could be the delivery and collection of two different building materials (e.g. lime and pebble) that are placed in suitable compartments of the vehicle. The states (q, s) of the process, where $|q| \leq Q_1$ and $|s| \leq Q_2$, after the first visit to customer j and the Actions 1–6 are the same as those defined in Section 2. The minimum expected future cost $f_j(q, s)$ for $j = 1, \ldots, N$, satisfies the dynamic programming Equations (1)-(4) and the boundary condition (5). The structure of the optimal restocking strategy is the same as in the case of discrete quantities and is described in the theorem below.

Theorem 2: *For each customer $j \in \{1, \ldots, N-1\}$, the structure of the optimal restocking strategy is described in the following four cases:*

(i) *For each $q \in [0, Q_1]$, there exists a critical quantity $k_{1j}(q) \in [0, Q_2]$, such that it is optimal for the vehicle to proceed to customer $j+1$ if and only if $s \geq k_{1j}(q)$. Moreover, $k_{1j}(q)$ is non-increasing in q.*
(ii) *There exists a critical quantity $s_j \leq 0$, such that it is optimal for the vehicle to make two trips to the depot when $q \in [0, Q_1]$ and $s \leq s_j$.*
(iii) *There exists a critical quantity $q_j \leq 0$, such that it is optimal for the vehicle to make two trips to the depot when $s \in [0, Q_2]$ and $q \leq q_j$.*
(iv) *For each $s \in [-Q_2, 0)$, there exists a critical quantity $k_{2j}(s) < 0$, such that it is optimal for the vehicle to make two trips to the depot if and only if $q \leq k_{2j}(s)$. Moreover, $k_{2j}(s)$ is non-increasing in s.*

3.2. Discretisation of state space

The state space after the first visit to customer j and after the maximum possible quantity of new items has been delivered and the maximum possible quantity of old items has been collected is the set: $S = \{(q, s) : q \in [-Q_1, Q_1], s \in [-Q_2, Q_2]\}$. A discretisation of the state space is necessary for the implementation of the dynamic programming algorithm. Let ρ be a relatively small number (e.g. $\rho = 0.05$ or $\rho = 0.01$). We discretise S by restricting our attention only to its points that belong to the set

$$\tilde{S} = \{(k\rho, l\rho) : k = -Q_1/\rho, \ldots, Q_1/\rho,\\ l = -Q_2/\rho, \ldots, Q_2/\rho\}.$$

The minimum expected cost $f_N(k\rho, l\rho), (k\rho, l\rho) \in \tilde{S}$, is found by using (5) with $q = k\rho$, $s = l\rho$. The minimum expected cost $f_j(k\rho, l\rho)$, $(k\rho, l\rho) \in \tilde{S}$ and the corresponding optimal decisions are found, recursively, for $j = N-1, N-2, \ldots, 1$, by using the dynamic programming Equations (1–4). The expectations are computed approximately. For example, the quantity $E_j(k\rho, l\rho)$, in Case 4, for $-Q_1/\rho \leq k < 0$ and $-Q_2/\rho \leq l < 0$, is computed approximately as follows:

$$E_j(k\rho, l\rho) = c_{j,j+1} + \sum_{x=0}^{Q_1/\rho - 1} \sum_{y=0}^{Q_2/\rho - 1} f_{j+1}(Q_1 + k\rho - x\rho,\\ Q_2 + l\rho - y\rho) h_{j+1}(x\rho, y\rho)\rho^2,$$

where h_{j+1} is the joint probability density function of ξ_{j+1} and ψ_{j+1}.

As in the case of discrete quantities, the optimal restocking strategy, i.e. the critical numbers $k_{1j}(k\rho)$, $k = 0, \ldots, Q_1/\rho$, $k_{2j}(l\rho), l = 0, \ldots, Q_2/\rho$, q_j and s_j for each customer $j \in \{1, \ldots, N-1\}$ can be found by a special-purpose dynamic programming algorithm that takes into account the structure of the optimal restocking strategy as given in Theorem 2.

3.3. Special-purpose dynamic programming algorithm

Algorithm for the determination of the critical numbers $k_{1j}(k\rho)$, $k = 0, \ldots, Q_1/\rho$, $k_{2j}(l\rho), l = 0, \ldots, Q_2/\rho$, q_j and s_j for customer $j = 1, \ldots, N-1$

Step 0 Set $f_N(k\rho, l\rho) = c_{N0} + 2c_{N0} \cdot 1((k\rho)^- + (l\rho)^-)$, $|k\rho| \leq Q_1, |l\rho| \leq Q_2$. Set $j = N-1$.
Step 1 (Determination of the critical numbers $k_{1j}(k\rho)$, $k = 0, \ldots, Q_1/\rho)$
Compute B_j.
For $k = 0, 1, \ldots, Q_1/\rho$, do the following:
For $l = Q_2/\rho, Q_2/\rho - 1, \ldots, 0$ compute $A_j(k\rho, l\rho)$ until $A_j(k\rho, l\rho) > B_j$ or $l\rho = -\rho$.
Set $k_{1j}(k\rho) = l\rho + \rho$.
Set $f_j(k\rho, l\rho) = B_j$, for $l \in \{0, \ldots, k_{1j}(k\rho)/\rho - 1\}$ and $f_j(k\rho, l\rho) = A_j(k\rho, l\rho)$, for $l \in \{k_{1j}(k\rho)/\rho, \ldots, Q_2/\rho\}$.
Step 2 (Determination of critical numbers $k_{2j}(l\rho), l = -Q_2/\rho, \ldots, -1)$
For $l = -1, -2, \ldots, -Q_2/\rho$, do the following:
For $k = -Q_1/\rho, -Q_1/\rho + 1, \ldots$ compute $E_j(k\rho, l\rho)$ until $E_j(k\rho, l\rho) < B_j$ or $k\rho = 0$.
Set $k_{2j}(l\rho) = k\rho - \rho$.
Set $f_j(k\rho, l\rho) = 2c_{j0} + B_j, k \in \{-Q_2/\rho, \ldots, k_{2j}(l\rho)/\rho\}$
and $f_j(k\rho, l\rho) = 2c_{j0} + E_j(k\rho, l\rho), k \in \{k_{2j}(l\rho)/\rho - 1, \ldots, -1\}$.
Step 3 (Determination of the critical numbers q_j)
For $k = -Q_1/\rho, -Q_1/\rho + 1, \ldots$ compute $C_j(k\rho)$ until $C_j(k\rho) < B_j$ or $k\rho = 0$.
Set $q_j = k\rho$.

Set $f_j(k\rho, l\rho) = 2c_{j0} + B_j$, for $k \in \{-Q_1/\rho, \ldots, q_j/\rho - 1\}$ and $l \in \{0, \ldots, Q_2/\rho\}$.
Set $f_j(k\rho, l\rho) = 2c_{j0} + C_j(k\rho)$, for $k \in \{q_j/\rho, \ldots, -1\}$ and $l \in \{0, \ldots, Q_2/\rho\}$.

Step 4 (Determination of critical numbers s_j)
For $l = -Q_2/\rho, -Q_2/\rho + 1, \ldots$ compute $D_j(l\rho)$ until $D_j(l\rho) < B_j$ or $l\rho = 0$.
Set $s_j = l\rho$.
Set $f_j(k\rho, l\rho) = 2c_{j0} + B_j$, for $l \in \{-Q_2/\rho, \ldots, s_j/\rho - 1\}$ and $k \in \{0, \ldots, Q_1/\rho\}$.
Set $f_j(k\rho, l\rho) = 2c_{j0} + D_j(l\rho)$, for $l \in \{s_j/\rho, \ldots, -1\}$ and $k \in \{0, \ldots, Q_1/\rho\}$.

Step 5 Set $j = j - 1$. If $j \geq 1$, go to Step 1. Otherwise, stop.

The above special-purpose dynamic programming algorithm is based on the structure of the optimal restocking strategy described in Theorem 2. Its complexity is $O(N[Q_1/\rho]^2 \cdot [Q_2/\rho]^2)$. It requires less computations than the initial dynamic programming algorithm. For example, for $j = 1, \ldots, N - 1$, the quantities $E_j(k\rho, l\rho)$, in Case 4, for k such that $-Q_1/\rho \leq k < k_{2j}(l\rho)/\rho$ and for l such that $-1 \leq l \leq -Q_2/\rho$, are not computed, while these quantities are computed in the initial dynamic programming algorithm. In Example 2 of Section 4, the significant difference of the computation times of these algorithms is verified, especially for high values of the number of customers N.

4. Numerical results

In the numerical results that we present below, we implemented the initial dynamic programming algorithm and the special-purpose dynamic programming algorithm by running the corresponding programs in Matlab on a personal computer equipped with an Intel Core i5-3230 M, 2.6 GHz processor and 4 GB of RAM. In Example 1, we assume that the demands for new items which are stored in Compartment 1 of the vehicle and the quantities of old items that are placed in Compartment 2 of the vehicle are discrete random variables. In Example 2, we assume that the demands for new items and the quantities of old products are continuous random variables. These examples verify the structural results presented in Theorem 1 and in Theorem 2.

Example 1: Suppose that $N = 10$. The capacities of Compartment 1 and 2 are $Q_1 = 15$ and $Q_2 = 12$, respectively. The travel costs between customer j and $j + 1$, $j \in \{1, \ldots, 9\}$, are given by: $c_{12} = 35$, $c_{23} = 31$, $c_{34} = 32$, $c_{45} = 30$, $c_{56} = 28$, $c_{67} = 32$, $c_{78} = 34$, $c_{89} = 33$ and $c_{9,10} = 34$. The travel costs between customers j, $j \in \{1, \ldots, 10\}$ and the depot are given by: $c_{10} = 41$, $c_{20} = 32$, $c_{30} = 36$, $c_{40} = 38$, $c_{50} = 39$, $c_{60} = 36$, $c_{70} = 32$, $c_{80} = 35$, $c_{90} = 39$ and $c_{10,0} = 37$. These costs satisfy the triangle inequality. We also assume that, for each customer $j \in \{1, \ldots, 10\}$, the demand ξ_j for new items which are stored in Compartment 1 and the quantity ψ_j of old products that are returned and placed in Compartment 2 are independent discrete random variables and follow the Binomial distributions $Bin(Q_1, p_1)$ and $Bin(Q_2, p_2)$,respectively. Choosing $p_1 = 0.4$ and $p_2 = 0.3$, we have that

$$\Pr(\xi_j = x) = \binom{Q_1}{x} 0.4^x 0.6^{Q_1 - x}, x = 0, \ldots, Q_1$$

and

$$\Pr(\psi_j = y) = \binom{Q_2}{y} 0.3^y 0.7^{Q_2 - y}, y = 0, \ldots, Q_2.$$

In Table 1, we present, for Customers 3 and 6, the critical numbers $k_{13}(q)$, $k_{16}(q)$, $q \in \{0, \ldots, Q_1\}$, $k_{23}(s), k_{26}(s), s \in \{-1, \ldots, -Q_2\}$, q_3, s_3, q_6, s_6, that correspond to the optimal restocking strategy. Note that Parts (i) and (iv) of Theorem 1 are confirmed numerically, since, for $j \in \{3, 6\}$, the critical numbers $k_{1j}(q)$ $q \in \{0, \ldots, Q_1\}$ and $k_{2j}(s), s \in \{-1, \ldots, -Q_2\}$ are non-increasing with respect to q and s, respectively. For each customer $j \in \{1, \ldots, N\}$ and for each value of $q \in \{0, \ldots, Q_1\}$, we define the critical number $k_{1j}(q)$ equal to $Q_2 + 1$, if Action 2 is taken for all values of $s \in \{0, \ldots, Q_2\}$. For each customer $j \in \{1, \ldots, N\}$ and for each value of $s \in \{-1, \ldots, -Q_2\}$, we also define the critical number $k_{2j}(s)$ equal to zero, if Action 4 is taken for all values of $q \in \{-1, \ldots, -Q_1\}$.

In Figures 2 and 3, we present the optimal decision, for each state (q, s), $q \in \{-Q_1, \ldots, Q_1\}, s \in \{-Q_2, \ldots, Q_2\}$, after the first visit to Customers 5 and 8, respectively. Action 1 is denoted by right-point triangles, Action 2 is denoted by red squares, Action 3 is denoted by yellow circles, Action 4 is denoted by green rhombs, Action 5 is denoted by cyan pentagrams and Action 6 is denoted by magenta hexagrams.

Figures 2 and 3 confirm the structural properties of the optimal restocking strategy that are described in Theorem 1. The value of the minimum expected total cost f_0 is found to be approximately equal to 550.01. The required computation time of the special-purpose dynamic programming algorithm is approximately equal to 9.4 s. It is considerably smaller than the corresponding computation time of the initial dynamic programming algorithm which is approximately equal to 15.25 s.

In Figure 4, for $p_2 = 0.35$, we present a graph that shows the variation of the minimum expected total cost f_0 as the probability p_1 of the binomial distribution

Table 1. The critical numbers of the optimal restocking strategy for Customers 3 and 6.

Customer	$k_{1j}(q), q \in \{0, \ldots, 15\}$ $j \in \{3, 6\}$	$k_{2j}(s), s \in \{-1, \ldots, -12\}$ $j \in \{3, 6\}$	$q_j, j \in \{3, 6\}$	$s_j, j \in \{3, 6\}$
3	$k_{13}(0) = \ldots = k_{13}(4) = 13$, $k_{13}(5) = 5$, $k_{13}(6) = k_{13}(7) = 3$, $k_{13}(8) = \ldots = k_{13}(15) = 2$	$k_{23}(-1) = k_{23}(-2) = \ldots = k_{23}(-7) = -10$, $k_{23}(-8) = k_{23}(-9) = -9$, $k_{23}(-10) = -7$, $k_{23}(-11) = k_{23}(-12) = 0$	$q_3 = -10$	$s_3 = -10$
6	$k_{16}(0) = \ldots = k_{16}(6) = 13$, $k_{16}(7) = 5$, $k_{16}(8) = \ldots = k_{16}(15) = 4$	$k_{26}(-1) = \ldots = k_{26}(-7) = -8$, $k_{26}(-8) = -7$, $k_{26}(-9) = \ldots = k_{26}(-12) = 0$	$q_6 = -8$	$s_6 = -8$

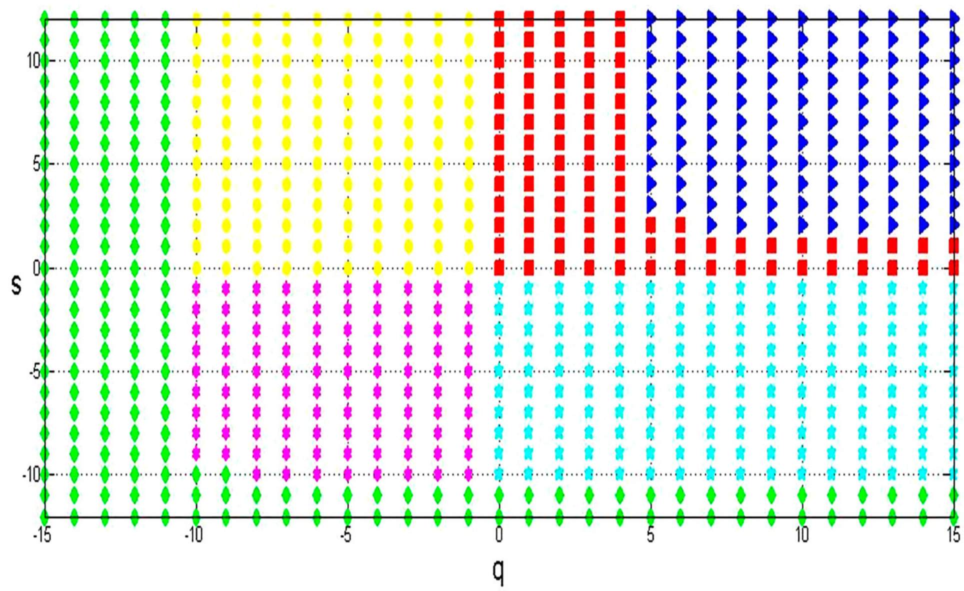

Figure 2. The optimal decisions after the first visit to Customer 5.

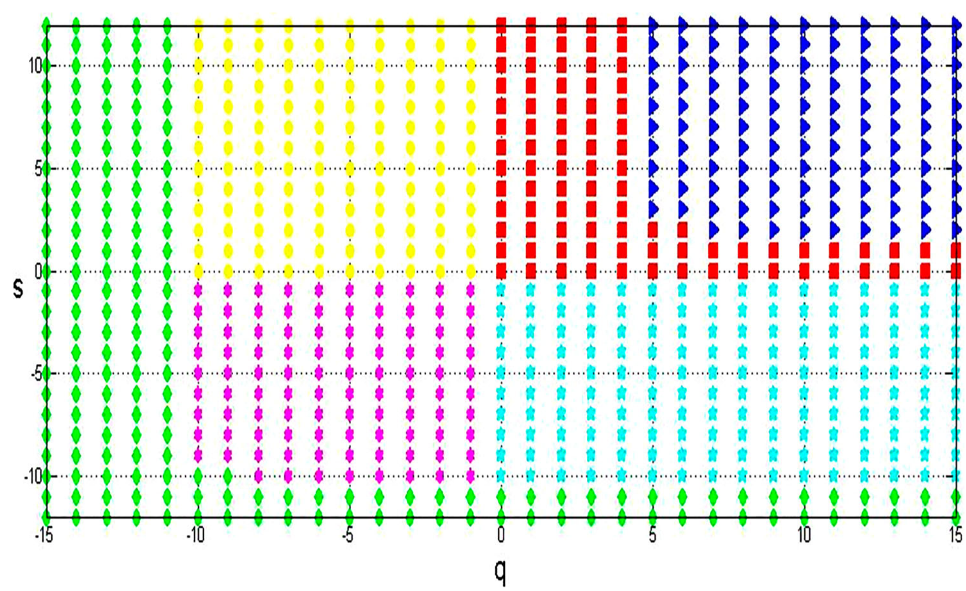

Figure 3. The optimal decisions after the first visit to Customer 8.

$Bin(Q_1, p_1)$ for the demands $\xi_j, j = 1, \ldots, 10$, takes values in the set $\{0.1, 0.15, \ldots, 0.9, 0.95\}$. We see that as p_1 takes values in the set $\{0.3, 0.35, \ldots, 0.65, 0.7\}$ the minimum expected total cost increases quickly and approximately linearly. When p_1 takes values in the set $\{0.1, 0.15, 0.2, 0.25\}$ and in the set $\{0.75, 0.8, 0.85, 0.9, 0.95\}$ the minimum expected total cost increases rather slowly.

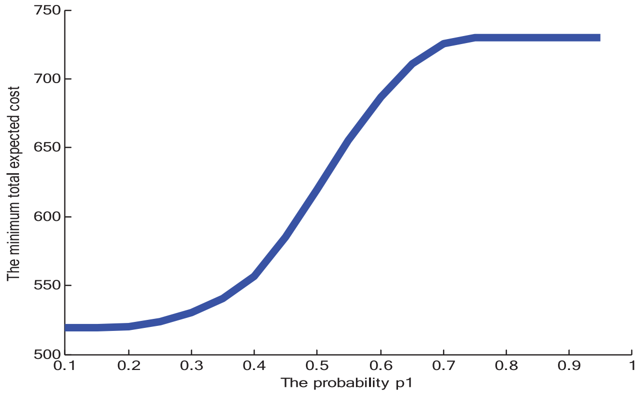

Figure 4. The minimum expected total cost as p_1 varies.

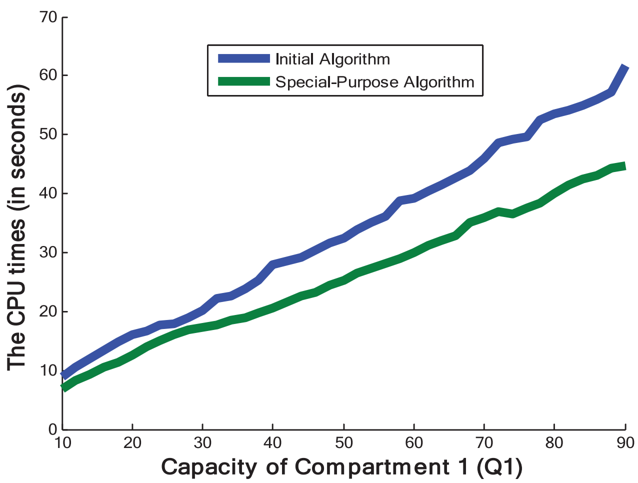

Figure 5. The computation times of the algorithms as Q_1 varies.

In Figure 5, for $\xi_j \sim Bin(Q_1, 0.4)$, $\psi_j \sim Bin(20, 0.3)$, $j = 1, \ldots, 10$, we present graphs that show, as Q_1 varies in the set {10, 12, ..., 88, 90} the variation in computation times, expressed in seconds, required by the initial dynamic programming algorithm and by the special-purpose dynamic programming algorithm.

We observe that, as Q_1 increases, the computation times for both algorithms increase rather linearly. The computation time required by the special-purpose algorithm is considerably smaller than the computation time required by the initial dynamic programming algorithm especially for high values of Q_1.

Example 2: Suppose that $N = 9$. The capacities of Compartment 1 and 2 are $Q_1 = 8$ and $Q_2 = 7$, respectively. The travel costs between customer j and $j+1$, $j \in \{1, \ldots, 8\}$, are given by: $c_{12} = 21$, $c_{23} = 16$, $c_{34} = 23$, $c_{45} = 15$, $c_{56} = 20$, $c_{67} = 16$, $c_{78} = 24$ and $c_{89} = 19$. The travel costs between customers j, $j \in \{1, \ldots, 9\}$ and the depot are given by: $c_{10} = 15$,, $c_{20} = 18$, $c_{30} = 14$, $c_{40} = 19$, $c_{50} = 17$, $c_{60} = 14$, $c_{70} = 18$, $c_{80} = 21$ and $c_{90} = 17$.

Table 2. The critical quantities q_j and s_j for Customers $j \in \{5, 6, 7, 8\}$.

Customer	Critical quantity q_j $j \in \{5, 6, 7, 8\}$	Critical quantity s_j $j \in \{5, 6, 7, 8\}$
5	−3.65	−2.8
6	−3.6	−2.7
7	−3.1	−2.4
8	−3.65	−2.65

These costs satisfy the triangle inequality. We also assume that, for each customer $j \in \{1, \ldots, 9\}$, the demand ξ_j for new products, that are delivered, and the quantity ψ_j of old products, that are returned, are independent continuous random variables which follow truncated Normal distributions in the intervals $[0, Q_1]$ and $[0, Q_2]$, respectively. For each customer $j \in \{1, \ldots, 9\}$, the probability density function $\phi_j(x)$ of the demand for new products is given by:

$$\phi_j(x) = [F(Q_1) - F(0)]^{-1} \frac{1}{\sigma_1 \sqrt{2\pi}} \exp\left\{-\frac{(x-\mu_1)^2}{2\sigma_1^2}\right\}, \quad x \in [0, Q_1], \mu_1 \in \Re, \sigma_1^2 > 0$$

and the probability density function $\theta_j(x)$ of returned old products is given by:

$$\theta_j(x) = [F(Q_2) - F(0)]^{-1} \frac{1}{\sigma_2 \sqrt{2\pi}} \exp\left\{-\frac{(x-\mu_2)^2}{2\sigma_2^2}\right\}, \quad x \in [0, Q_2], \mu_2 \in \Re, \sigma_2^2 > 0$$

where $F(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left\{-\frac{(t-\mu)^2}{2\sigma^2}\right\} dt$ is the cumulative distribution function of the Normal distribution with parameters $\mu \in \Re$ and $\sigma^2 > 0$. We choose $\mu_1 = 6, \sigma_1 = 4$ and $\mu_2 = 5, \sigma_2 = 2$. In Table 2, for customers $j \in \{5, 6, 7, 8\}$, we present the critical quantities q_j and s_j.

In Figures 6 and 7, we present the optimal decisions for Customers 4 and 7. If $q \in [0, Q_1], s \in [0, Q_2]$, Action 1 is coloured by blue and Action 2 is coloured by red. If $q \in [-Q_1, 0), s \in [0, Q_2]$, Action 3 is coloured by yellow and Action 4 is coloured by green. If $q \in [0, Q_1], s \in [-Q_2, 0)$, Action 5 is coloured by cyan and Action 4 is coloured by green. If $q \in [-Q_1, 0), s \in [-Q_2, 0)$, Action 6 is coloured by magenta and Action 4 is coloured by green. We choose $\rho = 0.05$ so that the discretised state space $\tilde{S}$ for each customer $j \in \{1, \ldots, 9\}$ is the set $\{(k * 0.05, l * 0.05) : k = -160, \ldots, 160, l = -140, \ldots, 140\}$.

Figures 6 and 7 confirm the structural results concerning the optimal restocking strategy that were given in Theorem 2. The value of the minimum expected total cost f_0 is found to be approximately equal to 317.7. The

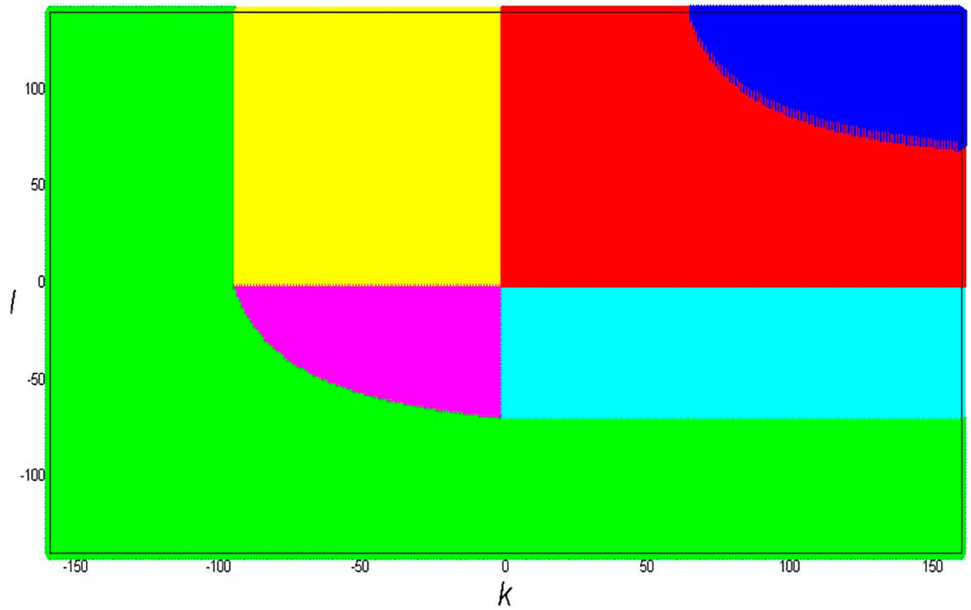

Figure 6. The optimal decisions for Customer 4.

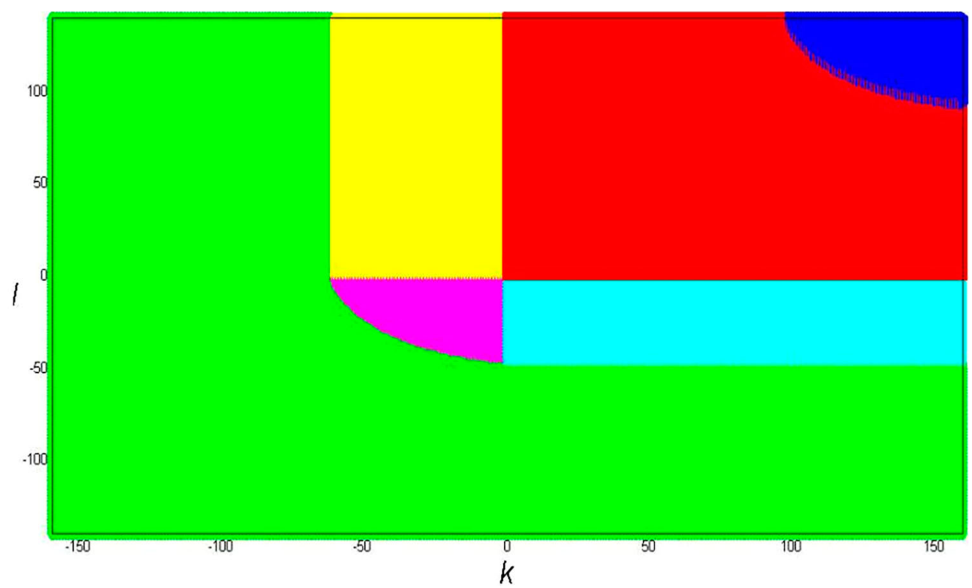

Figure 7. The optimal decisions for Customer 7.

computation time of the special-purpose dynamic programming algorithm is approximately equal to 1165 s. It is considerably smaller than the corresponding computation time of the initial dynamic programming algorithm which is approximately equal to 1729 s.

We now assume that the number of customers N takes values in the set $\{5, 6, \ldots, 15\}$. For each value of N, let $c_{i,i+1} = 16$, $i \in \{1, \ldots, N-1\}$, $c_{i0} = 18$, if i is odd and $c_{i0} = 14$, if i is even. In Figure 8, we present graphs that show, as N varies in the set $\{5, 6, \ldots, 15\}$, the variation in the computation times, expressed in seconds, required by the initial dynamic programming algorithm and by the special-purpose dynamic programming algorithm.

We observe that, as N increases, the required computation times for both algorithms increase approximately linearly. The computation time required by the special-purpose algorithm is considerably smaller than the computation time required by the initial dynamic programming algorithm for all values of N. The difference between the computation times increases as N increases.

5. The infinite-horizon problem

We modify the problem that we studied in Section 2 by considering an infinite-time horizon problem in which

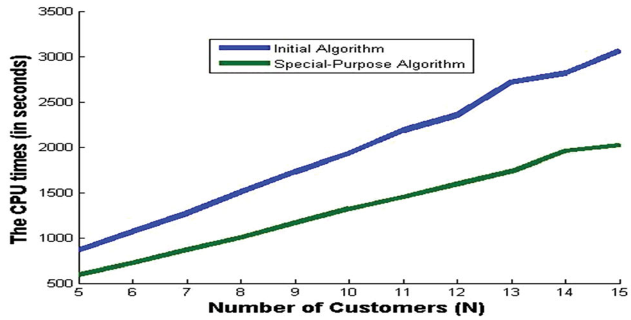

Figure 8. The computation times of the algorithms as N varies.

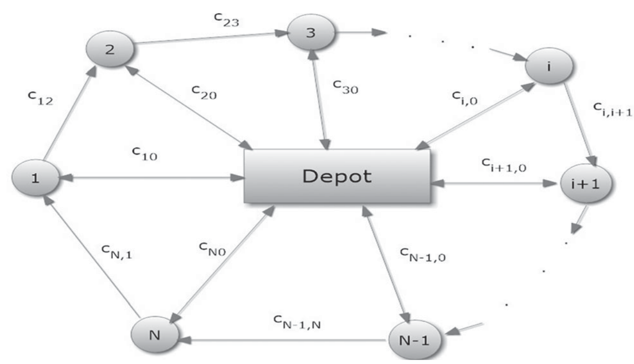

Figure 9. The customer network for the infinite-horizon problem.

the service of the customers does not stop when the service of the last customer N has been completed but it continues indefinitely with the same customer order. This means that, after the service of the last 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 customer network is presented in Figure 9.

The demands of the customers for new products, that are delivered, and the quantities of old products, that are collected, are renewed at successive tours of the vehicle. We assume that for each customer $j \in \{1, \dots, N\}$ the distribution of the quantity ξ_j of new products, that are delivered, and the distribution of the quantity ψ_j of old products, that are collected, remain the same at each tour. We suppose that, at each tour, the vehicle visits each customer, satisfies as much demand of new products as possible, collects the largest possible quantity of old products and chooses one decision among some possible decisions that coincide with the possible decisions in the finite-horizon problem.

It is assumed that the driver of the vehicle selects his/her decisions at equidistant time epochs $\tau = 0, 1, \dots$ (e.g. every 6 h). This means that, if, for example, the vehicle visits the fourth customer and the decision is selected at 2 am then the next decision is selected at 8 am after the first visit at fifth customer's site. Although this assumption is imposed in order to apply well-known results from the theory of Markov decision processes, there are situations in which this assumption may hold, as in the practical applications that we mentioned in Section 1. Specifically, in the first application, we may suppose that the vehicle, which visits N stores in order to deliver and collect fresh and expired milk (or ice-cream or vegetables or fruit), does not stop when the service of the last customer has been completed but it continues with the same customer order for a long time horizon. It can be assumed that the driver selects his/her decisions at equidistant time epochs (e.g. every 6 h). In the second application, the vehicle delivers and collects two different building materials. It can be assumed that the supply of building constructions with new materials and the collection of useless materials does not stop when the N-th building construction has been serviced but it continues indefinitely at equidistant time epochs with the same order.

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 $\tau = 0, 1, \dots$. The decision that is chosen by a policy at a decision epoch may depend on the history of the process or may be randomised 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. The usual optimisation criteria in the infinite-horizon problem are the minimisation of the expected total discounted cost and the minimisation of the long-run expected average cost per unit time. The expected total discounted cost under 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 long-run expected 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 $n-$th decision epoch divided by n, given that policy π is employed. Using well-known results of Markov decision processes (see e.g. Ch. 6 of Ross, 1992) we deduce that, under any one of these criteria, the optimal policy is stationary and has the same structure as the optimal policy in the finite-horizon problem. The state space of the system is the set $I = \{(j, q, s) : j = 1, \dots, N, q \in \{-Q_1, \dots, Q_1\}, s \in \{-Q_2, \dots, Q_2\}\}$, where the state (j, q, s) represents the situation at which the vehicle has completed the service of customer j, q is the quantity of new items that remain in Compartment 1 and s is the empty space for old items in Compartment 2.

Let $V_n^{\alpha}(j, q, s)$, $(j, q, s) \in I, 0 < \alpha < 1$, be the minimum $n-$step expected discounted cost if the initial state is $(j, q, s) \in I$ and α is the discount factor. The use of the discount factor $\alpha \in (0, 1)$ can be explained by the economic idea that a cost to be incurred in the future is discounted in today's money and thus we discount such cost at a rate α per unit of time. For Case 1, $V_n^\alpha(j, q, s)$ satisfies the following dynamic programming equations for $n = 1, 2, \ldots$.

If $0 \le q \le Q_1$, $0 \le s \le Q_2$, then

$$V_n^\alpha(j,q,s) = \min\{c_{j,j+1} + \alpha E V_{n-1}^\alpha(j+1, q-\xi_{j+1}, s-\psi_{j+1}), c_{j0} + c_{0,j+1} + \alpha E V_{n-1}^\alpha (j+1, Q_1 - \xi_{j+1}, Q_2 - \psi_{j+1})\}.$$

The dynamic programming equations for Cases 2–4 can be written similarly. We also have that $V_0^\alpha(j,q,s) = 0$, $(j,q,s) \in I$. In the above equations, we assume that $N+1$ is equal to 1 since the next customer after Customer N is Customer 1. It can be shown by induction on n that $V_n^\alpha(j,q,s)$ is non-increasing with respect to q and s in the same way as we proved that $f_j(q,s)$ is non-increasing in its arguments in Lemma 1. Let $V^\alpha(j,q,s)$, $(j,q,s) \in I$, denote the total $\alpha-$discounted expected cost if the initial state is $(j,q,s) \in I$. This quantity is finite since the state space I is finite. For Case 1, it satisfies the following optimality equations:

If $0 \le q \le Q_1$, $0 \le s \le Q_2$, then

$$V^\alpha(j,q,s) = \min\{c_{j,j+1} + \alpha E V^\alpha(j+1, q-\xi_{j+1}, s-\psi_{j+1}), c_{j0} + c_{0,j+1} + \alpha E V^\alpha (j+1, Q_1 - \xi_{j+1}, Q_2 - \psi_{j+1})\}.$$

The optimality equations for Cases 2–4 can be written similarly. It is well-known (see Corollary 6.6 in Ross, 1992) that, as $n \to \infty$, $V_n^\alpha(j,q,s) \to V^\alpha(j,q,s)$. Hence, $V^\alpha(j,q,s)$ is non-increasing in q and s. This result implies that the $\alpha-$ discounted cost optimal restocking strategy has the threshold-type structure described in Theorem 1.

We focus now on the minimisation of the expected average cost. First, we note that the state $(1, Q_1, Q_2)$ is accessible from any other state under any stationary policy. From Corollary 6.20 in Ross (1992), it follows that there exist numbers g and $h(j,q,s)$, $(j,q,s) \in I$, such that, if $0 \le q \le Q_1$, $0 \le s \le Q_2$, then

$$h(j,q,s) = \min\{c_{j,j+1} - g + Eh(j+1, q-\xi_{j+1}, s-\psi_{j+1}), c_{j0} + c_{0,j+1} - g + Eh(j+1, Q_1 - \xi_{j+1}, Q_2 - \psi_{j+1})\}.$$

The equations for Cases 2–4 can be written similarly. The above equations are known as the average-cost optimality equations.

The number g is the minimum average cost. It 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, 1992) such that

$$h(j,q,s) = \lim_{n\to\infty} [V^{\alpha_n}(j,q,s) - V^{\alpha_n}(1, Q_1, Q_2)], \quad (j,q,s) \in I.$$

The monotonicity of $V^{\alpha_n}(j,q,s)$ with respect to q and s implies that $h(j,q,s)$ is non-increasing with respect to q and s. This result implies that the average-cost optimal restocking strategy has the same structure as the finite-horizon optimal restocking strategy and the discounted-cost optimal policy.

The average-cost optimal restocking strategy can be found numerically by implementing the value-iteration algorithm, the policy-iteration algorithm and the linear programming formulation. We refer to Chapter 3 in Tijms (1994) 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. Let $p_{(j,q,s)(j+1,q',s')}(a)$ be the probability that the state at the next decision epoch will be the state $(j+1, q', s')$, if the present state is (j,q,s) and the action $a \in \{1, \ldots, 6\}$ is selected and let $C((j,q,s),a)$ be the corresponding expected cost. We give these quantities below.

If $0 \le q \le Q_1, 0 \le s \le Q_2$, then

$$p_{(j,q,s)(j+1,q',s')}(1) = \Pr(\xi_{j+1} = q - q', \psi_{j+1} = s - s'), \quad q' \in \{q - Q_1, \ldots, q\}, \quad s' \in \{s - Q_2, \ldots, s\},$$

$$p_{(j,q,s)(j+1,q',s')}(2) = \Pr(\xi_{j+1} = Q_1 - q', \psi_{j+1} = Q_2 - s'), \quad q' \in \{0, \ldots, Q_1\}, s' \in \{0, \ldots, Q_2\}.$$

If $-Q_1 \le q < 0, 0 \le s \le Q_2$, then

$$p_{(j,q,s)(j+1,q',s')}(3) = \Pr(\xi_{j+1} = Q_1 + q - q', \psi_{j+1} = Q_2 - s'), q' \in \{q, \ldots, q + Q_1\}, s' \in \{0, \ldots, Q_2\},$$

$$p_{(j,q,s)(j+1,q',s')}(4) = \Pr(\xi_{j+1} = Q_1 - q', \psi_{j+1} = Q_2 - s'), \quad q' \in \{0, \ldots, Q_1\}, s' \in \{0, \ldots, Q_2\}.$$

If $0 \le q \le Q_1, -Q_2 \le s < 0$, then

$$p_{(j,q,s)(j+1,q',s')}(5) = \Pr(\xi_{j+1} = Q_1 - q', \psi_{j+1} = Q_2 + s - s'), \quad q' \in \{0, \ldots, Q_1\}, s' \in \{s, \ldots, s + Q_2\},$$

$$p_{(j,q,s)(j+1,q',s')}(4) = \Pr(\xi_{j+1} = Q_1 - q', \psi_{j+1} = Q_2 - s'), \quad q' \in \{0, \ldots, Q_1\}, s' \in \{0, \ldots, Q_2\}.$$

If $-Q_1 \le q < 0, -Q_2 \le s < 0$, then

$$p_{(j,q,s)(j+1,q',s')}(6) = \Pr(\xi_{j+1} = Q_1 + q - q', \psi_{j+1} = Q_2 + s - s'), q' \in \{q, \dots, q+Q_1\}, s' \in \{s, \dots, s+Q_2\},$$

$$p_{(j,q,s)(j+1,q',s')}(4) = \Pr(\xi_{j+1} = Q_1 - q', \psi_{j+1} = Q_2 - s'), q' \in \{0, \dots, Q_1\}, s' \in \{0, \dots, Q_2\}.$$

If $0 \le q \le Q_1, 0 \le s \le Q_2$, then

$$C((j,q,s),\ 1) = c_{j,j+1},$$

$$C((j,q,s),\ 2) = c_{j0} + c_{0,j+1},$$

If $-Q_1 \le q < 0, 0 \le s \le Q_2$, then

$$C((j,q,s),\ 3) = 2c_{j0} + c_{j,j+1},$$

$$C((j,q,s),\ 4) = 3c_{j0} + c_{0,j+1},$$

If $0 \le q \le Q_1, -Q_2 \le s < 0$, then

$$C((j,q,s),\ 5) = 2c_{j0} + c_{j,j+1},$$

$$C((j,q,s),\ 4) = 3c_{j0} + c_{0,j+1},$$

If $-Q_1 \le q < 0, -Q_2 \le s < 0$, then

$$C((j,q,s),\ 6) = 2c_{j0} + c_{j,j+1},$$

$$C((j,q,s),\ 4) = 3c_{j0} + c_{0,j+1}.$$

As illustration we present the following example.

Example 3: Suppose that $N = 8$. The capacities of Compartment 1 and 2 are $Q_1 = 12$ and $Q_2 = 10$, respectively. The travel costs between customer j and $j+1$, $j \in \{1, \dots, 7\}$, are given by: $c_{12} = 25$, $c_{23} = 22$, $c_{34} = 21$, $c_{45} = 22$, $c_{56} = 20$, $c_{67} = 23$, $c_{78} = 24$ and $c_{81} = 25$. The travel costs between customers j, $j \in \{1, \dots, 8\}$ and the depot are given by: $c_{10} = 30$, $c_{20} = 28$, $c_{30} = 26$, $c_{40} = 24$, $c_{50} = 29$, $c_{60} = 26$, $c_{70} = 31$ and $c_{80} = 28$. These costs satisfy the triangle inequality. We also assume that, for each customer $j \in \{1, \dots, 8\}$, the demand ξ_j for new items, that are delivered, and the quantity of old items, that are returned, are independent discrete random variables with probability mass functions:

$$\Pr(\xi_j = x) = \left(\sum_{i=0}^{Q_1} e^{-\lambda_1} \frac{\lambda_1^i}{i!}\right)^{-1} e^{-\lambda_1} \frac{\lambda_1^x}{x!}, x = 0, \dots, Q_1$$

and

$$\Pr(\psi_j = y) = \left(\sum_{i=0}^{Q_2} e^{-\lambda_2} \frac{\lambda_2^i}{i!}\right)^{-1} e^{-\lambda_2} \frac{\lambda_2^y}{y!}, y = 0, \dots, Q_2,$$

respectively.

We set $\lambda_1 = 2$ and $\lambda_2 = 3$. The standard value-iteration does not converge in this example. 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 $\tilde{p}_{(j,q,s)(j+1,q',s')}(a) = \tau p_{(j,q,s)(j+1,q',s')}(a)$ and $\tilde{p}_{(j,q,s)(j,q,s)}(a) = 1 - \tau$, 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 e.g. p. 209 in Tijms, 1994).

In Table 3, we present, for Customers 2 and 5, the critical numbers $k_{12}(q)$, $k_{15}(q)$, $q \in \{0, \dots, Q_1\}$, $k_{22}(s), k_{25}(s), s \in \{-1, \dots, -Q_2\}$ and the critical numbers q_2, s_2 and q_5, s_5. Note that Parts (i) and (iv) of Theorem 1 are confirmed numerically, since, for $j \in \{2, 5\}$, the critical numbers $k_{1j}(q)$, $q \in \{0, \dots, Q_1\}$ and $k_{2j}(s), s \in \{-1, \dots, -Q_2\}$ are non-increasing with respect to q and to s, respectively.

In Figures 10 and 11, we present the optimal decision, for each state (q, s), $q \in \{-Q_1, \dots, Q_1\}, s \in \{-Q_2, \dots, Q_2\}$, after the first visit to Customers 3 and 6, respectively.

Figures 10 and 11 confirm that the average-cost optimal restocking strategy has the threshold-type structure described in Theorem 1. We implemented the value iteration algorithm in the perturbed model and we choose $\varepsilon = 10^{-3}$ as the tolerance number in the stopping criterion of the algorithm. The algorithm converged to the optimal policy after 66 iterations. The required computation time was approximately equal to 5 s. The average cost of the optimal policy was found to be approximately equal to 35.36.

It is also possible to consider the corresponding infinite-time horizon problems with continuous quantities of new products to be delivered to the customers and continuous quantities of old products to be collected from them. In this case, the state space of the system becomes:

$$I = \{(j,q,s) : j = 1, \dots, N, q \in [-Q_1, Q_1], s \in [-Q_2, Q_2]\}.$$

Using standard results of Markov decision theory (see Ch. 6 in Ross, 1992) it is possible to prove, in the same way as in the case of discrete quantities, that the infinite-horizon α−discounted cost optimal restocking strategy has the same threshold-type structure as the corresponding finite-horizon problem. It seems intuitively reasonable that the average-cost optimal restocking strategy has, in the case of continuous quantities, the same structure

Table 3. The critical numbers of the average-cost optimal restocking strategy for Customers 2 and 5

Customer	$k_{1j}(q), q \in \{0,\dots,12\}$ $j \in \{2, 5\}$	$k_{2j}(s), s \in \{-1,\dots,-10\}$ $j \in \{2, 5\}$	$q_j, j \in \{2, 5\}$	$s_j, j \in \{2, 5\}$
2	$k_{12}(0) = 11, k_{12}(1) = 6,$ $k_{12}(2) = 4,$ $k_{12}(3) = \dots = k_{12}(12) = 3$	$k_{22}(-1) = \dots = k_{22}(-4) = -10,$ $k_{22}(-5) = k_{22}(-6) = -9,$ $k_{22}(-7) = -8,$ $k_{22}(-8) = \dots = k_{22}(-10) = 0$	$q_2 = -10$	$s_2 = -8$
5	$k_{15}(0) = 11, k_{15}(1) = 4,$ $k_{15}(2) = \dots = k_{15}(12) = 3$	$k_{25}(-1) = \dots = k_{25}(-6) = -10,$ $k_{25}(-7) = -9,$ $k_{25}(-8) = \dots = k_{25}(-10) = 0$	$q_5 = -10$	$s_5 = -8$

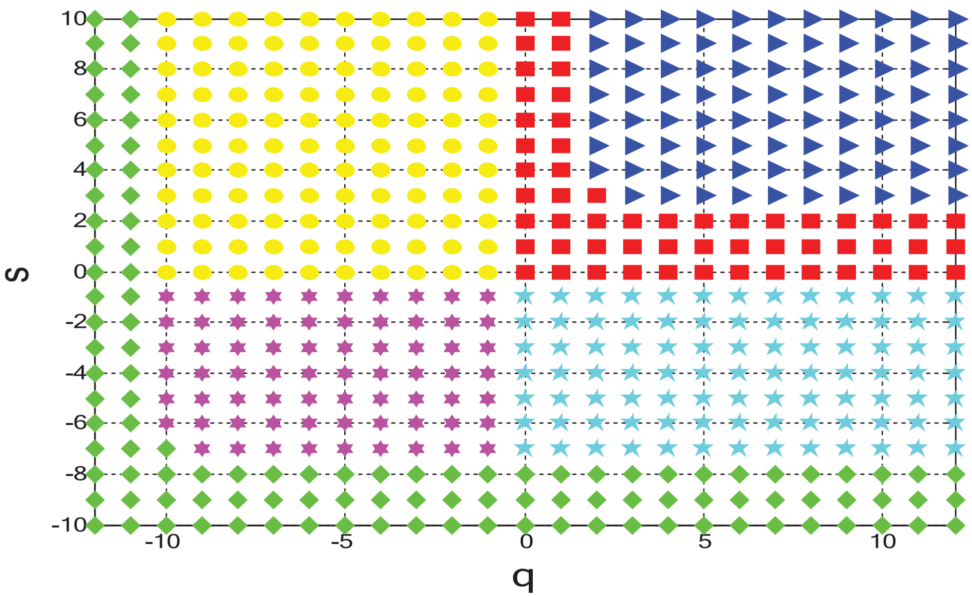

Figure 10. The optimal decisions after the first visit to Customer 3.

as in the case of discrete quantities. However, a rigorous proof seems to be difficult due to the fact that the state space I in this case is continuous, since it is not easy to prove that the minimum expected $\alpha-$discounted total cost is equicontinuous (see p. 150 in Ross, 1992). For the case of continuous quantities, we present below a numerical example in which the optimal restocking strategy is computed under the criterion of minimising the total expected $\alpha-$ discounted cost.

Example 4: Suppose that $N = 6$. The capacities of Compartment 1 and 2 are $Q_1 = 7$ and $Q_2 = 6$, respectively. The travel costs between customer j and $j+1$, $j \in \{1,\dots,6\}$, are given by: $c_{12} = 12$, $c_{23} = 11$, $c_{34} = 10$, $c_{45} = 11$, $c_{56} = 10$ and $c_{61} = 12$. The travel costs between customers j, $j \in \{1,\dots,6\}$ and the depot are given by: $c_{10} = 15, c_{20} = 14, c_{30} = 13, c_{40} = 12, c_{50} = 15$ and $c_{60} = 13$. These costs satisfy the triangle inequality. We also assume that, for each customer $j \in \{1,\dots,6\}$, the demand ξ_j for new products which are stored in Compartment 1 and the quantity ψ_j of old products that are returned from each customer and are placed in Compartment 2 are independent continuous random variables which follow Gamma distributions right-truncated in the intervals $[0, Q_1]$ and $[0, Q_2]$, respectively.

The probability density function $\varphi_j(x)$ of the random variable ξ_j is given by:

$$\varphi_j(x) = [F(Q_1)]^{-1}\frac{\lambda_1^{a_1}x^{a_1-1}}{\Gamma(a_1)}e^{-\lambda_1 x}, x \in [0, Q_1],$$

and the probability density function $\theta_j(x)$ of the random variable ψ_j is given by:

$$\theta_j(x) = [F(Q_2)]^{-1}\frac{\lambda_2^{a_2}x^{a_2-1}}{\Gamma(a_2)}e^{-\lambda_2 x}, x \in [0, Q_2],$$

where $a_1, \lambda_1 > 0$, $a_2, \lambda_2 > 0$, $\Gamma(a) = \int_0^\infty e^{-u}u^{a-1}du$, $a > 0$ and $F(x) = [\Gamma(a)]^{-1}\int_0^{\lambda x} e^{-u}u^{a-1}du, x \geq 0$. We choose $a_1 = 4, \lambda_1 = 2$ and $a_2 = 5, \lambda = 3$. We also choose $\rho = 0.05$ so that the discretised state space is $\tilde{I} = \{(j, k\rho, l\rho) : j = 1,\dots,6, -140 \leq k \leq 140, -120 \leq l \leq 120\}$. We select $\alpha = 0.8$ as the value of the discount factor. In Figure 12, we present the optimal decisions for Customer 1.

The structure of the optimal restocking strategy, as expected, is of threshold-type. We implemented the value

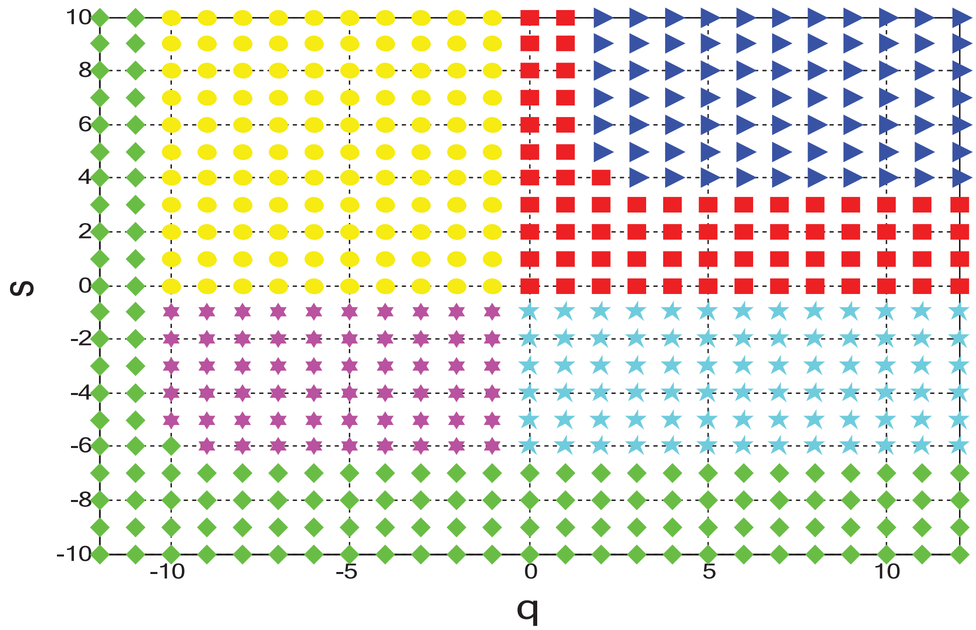

Figure 11. The optimal decisions after the first visit to Customer 6.

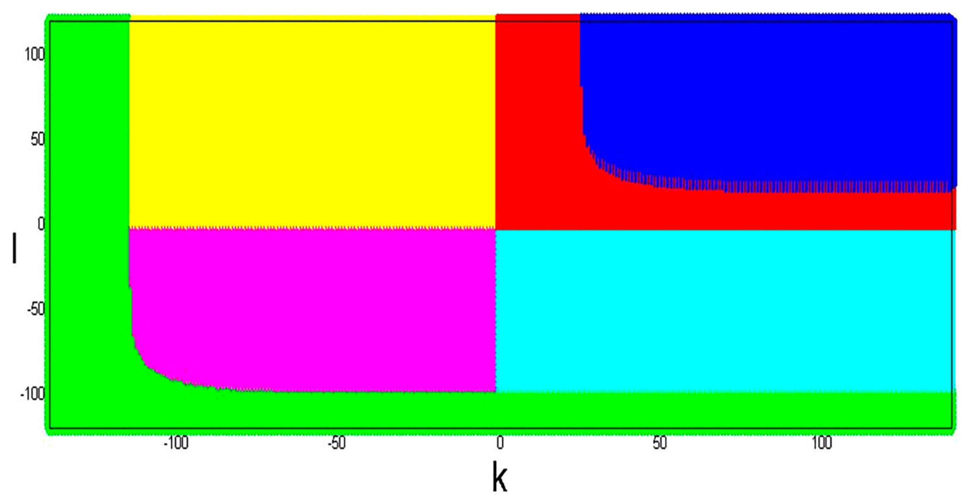

Figure 12. The optimal decisions for Customer 1.

iteration algorithm and we choose $\varepsilon = 10^{-3}$ as the tolerance number in the stopping criterion of the algorithm. The algorithm converged to the optimal restocking strategy after 39 iterations. The required computation time was approximately equal to 8466 s. The 0.8− discounted total expected cost of the optimal restocking strategy was found to be approximately equal to 27.65.

6. The problem when the customers are not ordered

We generalise the problem that we introduced in Section 2 by assuming that the customers are not serviced according to a predefined order. In this case, there are $N!$ different customer sequences that the vehicle may follow. For each sequence using the suitable dynamic programming algorithm, we can find the optimal restocking strategy and the corresponding minimum expected total cost and then by comparing these minimum costs we can determine the optimal customer sequence that achieves the overall minimum cost. Numerical experiments indicate that, if the demands of the customers for new products, that are delivered, and the quantities of old products, that are returned, are discrete random variables, it is possible to find the optimal customer sequence for values of N up to 9. If the demands of the customers for new products and the quantities of old products are continuous random variables, it is also possible to find the optimal customer sequence for values of N up to 5. As illustrations, we present below two numerical examples. In Example 5, the

Table 4. The optimal customer sequence for $N = 3, 4, 5, 6, 7, 8, 9$.

N	$N!$	Minimum cost	Optimal sequence	Time 1	Time 2
3	6	152.44	2,1,3	0.0156	0.0123
4	24	189.47	2,4,1,3	0.8298	0.5214
5	120	236.28	2,4,5,1,3	2.0115	1.5881
6	720	280.38	2,4,5,1,6,3	16.2027	6.1533
7	5040	322.91	2,4,5,1,7,6,3	210.6245	122.3542
8	40,320	365.42	5,4,2,8,1,7,6,3	1123	509.1178
9	362,880	416.51	9,5,4,2,8,1,7,6,3	11,936	3845

demands for new products and the quantities of returned old products are discrete random variables and in Example 6, the demands for new products and the quantities of returned old products are continuous random variables.

Example 5: Suppose that the capacities of Compartment 1 and 2 are $Q_1 = 10$ and $Q_2 = 8$, respectively. We assume that the number of customers N takes values in the set $\{3, 4, \dots, 9\}$. The travel costs c_{ij} between customers $i, j \in \{1, \dots, 9\}$ and the travel costs c_{i0} between each customer $i \in \{1, \dots, 9\}$ and the depot are given by the following symmetric matrix $C = (c_{ij})$, $i, j = 0, \dots, 9$.

$$C = \begin{pmatrix} 0 & 28 & 30 & 23 & 22 & 27 & 25 & 24 & 26 & 29 \\ 28 & 0 & 23 & 21 & 20 & 21 & 22 & 20 & 18 & 22 \\ 30 & 23 & 0 & 24 & 18 & 20 & 23 & 21 & 17 & 20 \\ 23 & 21 & 24 & 0 & 19 & 21 & 17 & 20 & 18 & 22 \\ 22 & 20 & 18 & 19 & 0 & 17 & 21 & 18 & 19 & 20 \\ 27 & 21 & 20 & 21 & 17 & 0 & 23 & 21 & 19 & 20 \\ 25 & 22 & 23 & 17 & 21 & 23 & 0 & 20 & 22 & 23 \\ 24 & 20 & 21 & 20 & 18 & 21 & 20 & 0 & 20 & 21 \\ 26 & 18 & 17 & 18 & 19 & 19 & 22 & 20 & 0 & 21 \\ 29 & 22 & 20 & 22 & 20 & 20 & 23 & 21 & 21 & 0 \end{pmatrix}$$

These costs satisfy the triangle inequality. For each customer j, we assume that the demand ξ_j for new products and the quantity ψ_j of returned old products are independent random variables which follow the discrete Uniform distribution in the set $\{0, \dots, Q_1\}$ and in the set $\{0, \dots, Q_2\}$, respectively, i.e. $\Pr(\xi_j = x) = (Q_1 + 1)^{-1}, x = 0, \dots, Q_1$ and $\Pr(\psi_j = y) = (Q_2 + 1)^{-1}$, $y = 0, \dots, Q_2$. For $N \in \{3, \dots, 9\}$ we consider the network consisting of customers $1, \dots, N$. In Table 4, we present for $N \in \{3, \dots, 9\}$ the number $N!$ of all possible customer sequences, the minimum expected cost among all customer sequences, the optimal customer sequence, the required computation time for the determination of the optimal customer sequence in seconds (Time 1) if the initial dynamic programming algorithm is used and the required computation time in seconds (Time 2) if the special-purpose dynamic programming algorithm is used.

In Figure 13, we present the graphs that show, as N takes values in the set $\{3, \dots, 9\}$, the variation in required computation times, expressed in seconds, for the determination of the optimal customer sequence, if the initial dynamic programming algorithm and if the special-purpose dynamic programming algorithm are used.

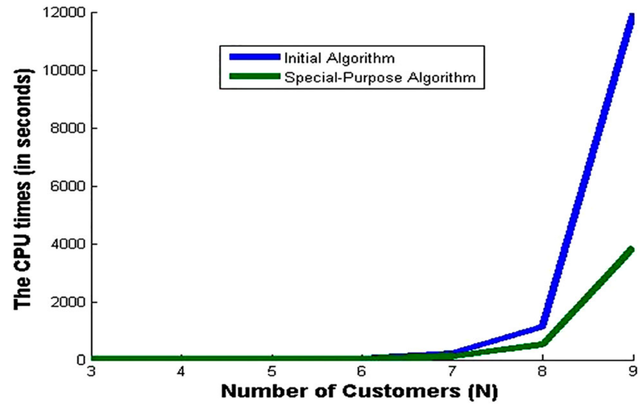

Figure 13. The computation times of the algorithms as N varies.

We observe that, as N increases, both computation times seem to increase exponentially. The required computation time if the special-purpose dynamic programming algorithm is used is considerably smaller than the required computation time if the initial dynamic programming algorithm is used.

Example 6: Suppose that the capacities of Compartment 1 and 2 are $Q_1 = 7$ and $Q_2 = 6$, respectively. We assume that the number of customers N takes values in the set $\{3, 4, 5\}$. The travel costs c_{ij} between customers $i, j \in \{1, \dots, 5\}$ and the travel costs c_{i0} between each customer $i \in \{1, \dots, 5\}$ and the depot are given by the following symmetric matrix $C = (c_{ij})$, $i, j = 0, \dots, 5$.

$$C = \begin{pmatrix} 0 & 20 & 22 & 15 & 17 & 19 \\ 20 & 0 & 18 & 16 & 15 & 16 \\ 22 & 18 & 0 & 16 & 15 & 12 \\ 15 & 16 & 16 & 0 & 17 & 15 \\ 17 & 15 & 15 & 17 & 0 & 17 \\ 19 & 16 & 12 & 15 & 17 & 0 \end{pmatrix}$$

These costs satisfy the triangle inequality. For each customer j, we assume that the demand ξ_j for new products and the quantity ψ_j of old products are independent random variables which follow the continuous Uniform distribution in the interval $[0, Q_1]$ and in the interval $[0, Q_2]$, respectively, i.e. the probability density function of ξ_j is given by: $\varphi_j(x) = {Q_1}^{-1}, x \in [0, Q_1]$ and the probability density function of ψ_j is given by: $\theta_j(x) = {Q_2}^{-1}, x \in [0, Q_2]$. We choose $\rho = 0.05$ so that the discretised state space $\tilde{S}$ for each customer is the set: $\{k * 0.05, l * 0.05) : k = -140, \dots, 140, l =$

Table 5. The optimal customer sequence for $N = 3, 4, 5$.

N	$N!$	Minimum cost	Optimal sequence	Time 1	Time 2
3	6	111.73	1,2,3	923.86	475.07
4	24	143.15	2,4,1,3	5796	2729
5	120	177.75	1,4,2,5,3	35,349	19,129

$-120, \dots, 120\}$. For $N \in \{3, 4, 5\}$ we consider the network consisting of customers $1, \dots, N$. In Table 5, we present for $N \in \{3, 4, 5\}$ the number $N!$ of all possible customer sequences, the minimum expected cost among all customer sequences, the optimal customer sequence, the required computation time in seconds (Time 1) for the determination of the optimal customer sequence if the initial dynamic programming algorithm is used and the required computation time in seconds (Time 2) if the special-purpose dynamic programming algorithm is used.

It can be seen that, as N increases, both computation times seem to increase exponentially. The required computation time if the special-purpose dynamic programming algorithm is used is considerably smaller than the required computation time if the initial dynamic programming algorithm is used.

7. Conclusions and topics for future research

In this work, a capacitated and compartmentalised stochastic vehicle routing problem was studied. It was assumed that (i) the customers are serviced according to a predefined sequence, (ii) the vehicle delivers to each customer a quantity of new (or fresh or useful) products and collects a quantity of old (or expired or useless) products, (iii) the quantities that are delivered and collected are stochastic; the actual quantities are disclosed when the vehicle visits each customer, (iv) the new and old products are placed in two different compartments of the vehicle. We defined six different actions that can be selected when the vehicle arrives at each customer's site. The cost structure of the problem includes travel costs between consecutive customers and travel costs between customers and the depot. A stochastic dynamic programming algorithm was given that leads to the restocking strategy that minimises the expected total cost for servicing all customers. We proved that, for each customer, the optimal restocking strategy is of threshold-type. According to this structural result, the set of all possible states that correspond to each customer is divided in eight disjoint subsets. The optimal restocking strategy prescribes the same action at all states of each subset. If the above Assumption (i) does not hold, it is possible to compute the optimal restocking strategy for moderate values of the number of customers. We also investigated the corresponding infinite-time horizon problem. We showed that the discounted-cost optimal restocking strategy and the average-cost optimal restocking strategy have the same structure as the optimal restocking strategy in the finite-time horizon problem.

A topic for future research could be the determination of the optimal restocking strategy for a more general problem in which (i) the customers are not serviced according to a particular sequence and (ii) the number of the customers is large.

Acknowledgements

We would like to thank two anonymous reviewers whose comments improved the presentation of the paper.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

The author Epaminondas G. Kyriakidis has been financed by the research program EP-3042-01 (RC/AUEB).

Notes on contributors

Constantinos C. Karamatsoukis is Assistant Professor in Stochastic Operations Research in the Department of Military Sciences at Hellenic Military Academy. His research interests are stochastic models in Operations Research, Markov decision processes and their applications, stochastic dynamic programming, approximate dynamic programming and logistics. His work has been published in European Journal of Operational Research, Probability in the Engineering and Informational Science, Methodology and Computing in Applied Probability and others.

Epaminondas G. Kyriakidis is Professor in Applied Probability and in Stochastic Operations Research in the Department of Statistics at Athens University of Economics and Business. His research interests are Markov decision processes and their applications, maintenance models, optimal control of biological populations and logistics. His work has been published in Journal of Applied Probability, European Journal of Operational Research, Computers & Operations Research, Probability in the Engineering and Informational Science, Methodology and Computing in Applied Probability and others.

Theodosis D. Dimitrakos is Assistant Professor in Applied Probability and in Stochastic Operations Research in the Department of Mathematics at University of the Aegean. His research interests are stochastic dynamic programming, Markov decision processes and their applications, maintenance models and logistics. His work has been published in Methodology and Computing in Applied Probability, International Journal of Production Economics, European Journal of Operational Research and others.

References

Bertazzi, L., & Secomandi, N. (2018a). Worst-case benefit of restocking for the vehicle routing problem with stochastic demands. Available at SSRN 3246339.

Bertazzi, L., & Secomandi, N. (2018b). Faster rollout search for the vehicle routing problem with stochastic demands and restocking. *European Journal of Operational Research*, *270*, 487–497.

Dikas, G., Minis, I., & Mamasis, K. (2016). Single vehicle routing with predefined client sequence and multiple warehouse returns: The case of two warehouses. *Central European Journal of Operations Research*, *24*, 709–730.

Dimitrakos, T. D., & Kyriakidis, E. G. (2015). A single vehicle routing problem with pickups and deliveries, continuous random demands and predefined customer order. *European Journal of Operational Research*, *244*, 990–993.

Florio, A. M., Hartl, R. F., & Minner, S. (2018). Optimal a priori tour and restocking policy for the single-vehicle routing problem with stochastic demands. *European Journal of Operational Research*. doi:10.1016/j.ejor.2018.10.045

Kyriakidis, E. G., & Dimitrakos, T. D. (2019). Stochastic single vehicle routing problem with ordered customers and partial fulfillment of demands. *International Journal of Systems Science: Operations & Logistics*, *6*(3), 285–299.

Kyriakidis, E. G., Dimitrakos, T. D., & Karamatsoukis, C. C. (2019). Optimal delivery of two similar products to *N* ordered customers with product preferences. *International Journal of Production Economics*, *209*, 194–204.

Louveaux, F. V., & Juan-José Salazar-González, J.-J. (2018). Exact approach for the vehicle routing problem with stochastic demands and preventive returns. *Transportation Science*, *52*(6), 1–16. Articles in Advance.

Pandelis, D. G., Karamatsoukis, C. C., & Kyriakidis, E. G. (2013). Single vehicle routing problems with a predefined customer order, unified load and stochastic discrete demands. *Probability in the Engineering and Informational Sciences*, *27*(1), 1–23.

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.

Ross, S. M. (1992). *Applied probability models with optimization applications*. New York, NY: Dover.

Salavati-Khoshghal, M. S., Gendreau, M., Jabali, O., & Rei, W. (2019). An exact algorithm to solve the vehicle routing problem with stochastic demands under an optimal restocking policy. *European Journal of Operational Research*, *273*, 175–189.

Sipser, M. (2013). *Introduction to the theory of computation* (3rd ed.). Boston, MA: Cengage Learning.

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.

Tijms, H. C. (1994). *Stochastic models: An algorithmic approach*. New York, NY: Wiley.

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.

Yang, W.-H., Mathur, K., & Ballou, R. H. (2000). Stochastic vehicle routing problem with restocking. *Transportation Science*, *34*, 99–112.

Yee, J. R., & Golden, B. I. (1980). A note on determining operating strategies for probabilistic vehicle routing. *Naval Research Logistics Quarterly*, *27*, 159–163.

Zhang, J., Lam, W. H. K., & Chen, B. Y. (2016). On-time delivery probabilistic models for the vehicle routing problem with stochastic demands and time windows. *European Journal of Operational Research*, *249*, 144–154.

Zhu, L., & Sheu, J.-B. (2018). Failure-specific cooperative recourse strategy for simultaneous pickup and delivery problem with stochastic demands. *European Journal of Operational Research*, *271*, 896–912.