International Journal of Production Economics 209 (2019) 194–204

Contents lists available at ScienceDirect

International Journal of Production Economics

journal homepage: www.elsevier.com/locate/ijpe

Optimal delivery of two similar products to N ordered customers with product preferences

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

[a] Department of Statistics, Athens University of Economics and Business, Patission 76, 10434, Athens, Greece
[b] Department of Mathematics, University of the Aegean, Karlovassi, 83200, Samos, Greece
[c] Department of Military Sciences, Hellenic Military Academy, Vari, 16673, Attica, Greece

ARTICLE INFO

Keywords:
Vehicle routing
Stochastic dynamic programming
Similar products

ABSTRACT

We study a mathematical model for a specific vehicle routing problem in which a vehicle starts its route from a depot loaded with items of two similar but not identical products. The vehicle must deliver the products to N customers according to a predefined sequence. It is assumed that each customer prefers either product 1 or product 2 with known probabilities and the quantity that each customer demands is a random variable with known distribution. The actual preference and demand of each customer are revealed upon the vehicle's arrival at customer's site. The demand of each customer cannot exceed the vehicle capacity and the vehicle is allowed during its route to return to the depot to restock with quantities of both products. The travel costs between consecutive customers and the travel costs between the customers and the depot are known. If there is shortage for the desired product it is permitted to deliver the other product at a reduced price. The optimal routing strategy is found by implementing a suitable stochastic dynamic programming algorithm. It is possible to prove that the optimal routing strategy has a specific threshold-type structure. Furthermore, if we consider the same problem without the assumption that the customers are ordered, numerical experiments indicate that the optimal routing strategy can be computed for $N \leq 8$.

1. Introduction

A well-known problem in Operations Research is the vehicle routing problem (VRP). The context of VRP is that of delivering goods located at a central depot to customers who are scattered in a geographical area and have ordered these goods. A vehicle or several vehicles start their routes from the depot and visit the customers in order to satisfy their orders. After servicing all customers the vehicles return to the depot. The objective is to minimize the total transportation cost for servicing the customers. The VRP has been extensively studied in the optimization literature during the last fifty-five years. Several variations of the vehicle routing problem have been considered: (i) the VRP with time windows (see e.g. Cortés et al. (2014)) in which the customers are served within predefined time windows. Time windows are defined as hard when it is not allowed to deliver outside of the time interval. Soft time windows on the other hand allow deliveries outside the time interval with a penalty cost, (ii) the capacitated VRP with or without time windows (see e.g. Laporte et al. (2002), Syrichas and Crispin (2017)) in which the vehicles have limited carrying capacity of the goods that must be delivered, (iii) the VRP with backhauls (see e.g Goetschalckx and Jacobs-Blecha (1989), Belloso et al. (2017)) in which the customers are divided into linehaul customers, who require a given quantity of product to be delivered, and backhaul customers, who require a given quantity of product to be picked up, (iv) the VRP with pickup and delivery (see e.g Tasan and Gen (2012), Zhang et al. (2012)) in which each customer is associated with two quantities representing the demands of products to be delivered and picked up, (v) the VRP with multiple trips (see e.g. Olivera and Viera (2007)) in which each vehicle can be scheduled for more than one trip, as long as it corresponds to the maximum distance allowed in the workday, (vi) open VRP (see e.g. Derigs and Reuter (2009)) in which the vehicles are not required to return to the depot after servicing the customers, (vii) VRP with multiple compartments (see e.g. Derigs et al. (2011)) in which each compartment of the vehicles is suitable for a single product. The VRP is a NP-hard problem that in some cases can be solved exactly by algorithms (for example branch-and-bound, branch-and-cut, branch-and-cut-and-price methods) that lead to the optimal routing strategy (see e.g. Gauvin et al. (2014), Dayarian et al. (2015)). Heuristics and metaheuristics (tabu search, simulated annealing, genetic

* Corresponding author.
E-mail addresses: ekyriak@aueb.gr (E.G. Kyriakidis), dimitheo@aegean.gr (T.D. Dimitrakos), ckaramatsoukis@sse.gr (C.C. Karamatsoukis).

https://doi.org/10.1016/j.ijpe.2017.12.017
Received 20 October 2016; Received in revised form 13 November 2017; Accepted 21 December 2017
Available online 27 December 2017

algorithms, colony optimization) have also been developed in many cases (see e.g. Kriticos and Ioannou (2013), Segerstedt (2014)). Although these approaches do not guarantee optimality, they yield best results in practice. Furthermore, hybrid methods use combination of exact algorithms, heuristics or metaheuristics to the VRP. It is noteworthy that a great amount of research is related to the Stochastic VRP (see e.g. Gendreau et al. (1996), Haugland et al. (2007), Nguyen et al. (2016)) that contains stochastic components, as the demands of the customers, the vehicle travel times and the service times of the customers. Recent surveys of research on VRP have been given by Pillac et al. (2013), Toth and Vigo (2014) and Psaraftis et al. (2016).

In the last seventeen years various capacitated vehicle routing problems have been studied in which a single vehicle starts its route from a depot and serves N customers according to a predefined sequence. Suitable dynamic programming formulations have been given for these problems. It was shown that the form of the optimal routing strategy is of threshold-type, i.e for each customer it is characterized by some critical numbers. We present below these studies. In Section 3 of Yang et al. (2000) the demands of the customers were assumed to be discrete random variables with known distributions. It was shown that, for each customer, the optimal routing strategy is characterized by an inventory threshold. If, after completing the service of a customer, the remaining amount of products in the vehicle is greater or equal to the threshold, then the vehicle proceeds to the next customer. Otherwise, it returns to the depot for replenishment, and then resumes its route. In Kyriakidis and Dimitrakos (2008) an analogous result was proved for the case of continuous random demands of the customers. Tsirimpas et al. (2008) assumed that the demands of the customers are deterministic and investigated (i) the case of multiple-product deliveries when each product is stored in its own compartment in the vehicle, (ii) the case of multiple-product deliveries when all products are stored together in the vehicle's single compartment, and (iii) the case in which the vehicle picks up from and delivers a single product to each customer. In each case the optimal routing strategy was found by implementing a suitable dynamic programming algorithm. Tatarakis and Minis (2009) studied cases (i), (ii) and Minis and Tatarakis (2011) studied case (iii) when the demands of the customers are discrete random numbers. In these papers structural results for the optimal routing strategies were obtained that were generalized by Pandelis et al. (2012, 2013a, 2013b). In the last three papers the corresponding infinite-time horizon problems were also studied where the service of the customers does not stop when the last customer has been serviced but it continues indefinitely with the same customer order. Kyriakidis and Dimitrakos (2013) assumed that a penalty is imposed if a customer's demand is not satisfied or if it is satisfied partially. It was shown that, for each customer, the optimal routing strategy has a specific threshold-type structure that is characterized by three critical numbers. Dimitrakos and Kyriakidis (2015) extended the results obtained by Pandelis et al. (2013b) to the case where the demands of the customers for a material are continuous random variables instead of discrete ones. In Section 4 of Zhang et al. (2016) a vehicle routing problem with ordered customers, stochastic discrete demands and time windows was investigated and the structure of the optimal routing strategy was proved to be of threshold-type. Dikas et al. (2016) considered a vehicle routing problem with a predefined customer sequence, deterministic customer demands and two load replenishment depots, which may be of limited capacity. A dynamic programming algorithm was developed for the simplest case and labeling algorithms or a partitioning heuristic were developed for more complex cases. Realistic applications of these problems are described in the above papers.

In the present work we study another problem under the assumption that the customers are ordered. We suppose that the vehicle carries two similar but not identical products that are delivered to the customers according to their preferences. If there is lack of the product that a customer prefers, it is possible to deliver to him/her the other product. In this case a penalty cost is incurred that could be due to some reduction of the price of the product that is delivered or to some loss of goodwill. We assume that a vehicle starts its route from a depot loaded to its full capacity with items of two similar but not identical products and visits N customers according to a predefined sequence $1 \rightarrow 2 \rightarrow \cdots \rightarrow N$. The probability that he/she prefers a particular product is known. The quantity that each customer demands is a random variable with known distribution. The actual preferences and demands of the customers are revealed only when a vehicle arrives at their sites. If there is lack of the preferred product, the demand of the customer may be satisfied by delivering the product that is not preferred. The vehicle may interrupt its route by returning to the depot to restock with quantities of both products. The total cost for servicing all customers consists of (i) travel costs between consecutive customers, (ii) travel costs between customers and the depot and (iii) penalty costs. The assumption that the customers are ordered and an appropriate selection of decision epochs enable us to develop a dynamic programming algorithm for the determination of the optimal routing strategy of the vehicle. It is shown that the optimal routing strategy has a specific threshold-type structure. This characterization enables us to design an efficient special-purpose dynamic programming algorithm that leads to the optimal routing strategy and requires less computations that the initial dynamic programming algorithm.

A practical application of the problem could be the delivery of two similar materials or goods to patients in a healthcare facility (see Dikas et al. (2016)). For example, the staff of the healthcare facility may distribute two similar linens or two similar medical materials or two similar meals to the patients. The service is performed according to predefined sequence, usually room after room. The staff may interrupt the route in order to return to the warehouse to reload the transportation carts. Another real-world application of the proposed model is the distribution of two similar types of ammunition (e.g. bullets, missiles) to military units. Because of safety reasons or command restrictions, the vehicle carrying military equipments may follow a predefined sequence in order to deliver the ammunition to military units. The determination of the optimal routing strategy will reduce the total transportation cost.

The rest of the paper is organized as follows. In Section 2 the problem is specified and analyzed for the case of random discrete demands of the customers. A dynamic programming formulation is given for the determination of the optimal routing strategy of the vehicle. The structure of the optimal routing strategy is shown and an efficient special-purpose dynamic programming algorithm is presented. In Section 3 similar results are obtained for the case of random continuous demands of the customers. In Section 4 the theoretical results are illustrated by numerical examples. In Section 5 we consider the more general problem without the assumption that the customers are ordered. In the last section we give a summary of the main results of the paper and a topic for future research.

2. The problem and the optimal routing strategy

2.1. The problem

We assume that a vehicle starts its route from a depot and visits N customers in order to deliver them two similar but not identical products. We name these products, product 1 and product 2. An item of product 1 has the same size as an item of product 2. For example, an item of product 1 could be a bottle of milk A and an item of product 2 could be a bottle of milk B with the same size. The customers are serviced according to a predefined sequence $1 \rightarrow 2 \rightarrow \cdots \rightarrow N$. This means that first customer 1 must be serviced, then customer 2 must be serviced, then customer 3 must be serviced and so on. After servicing all customers, the vehicle returns to the depot. Suppose that (i) the capacity of the vehicle is finite and it is equal to Q items of product 1 or product 2, (ii) the demand of customer $j \in \{1, \ldots, N\}$ is a discrete random variable $\xi_j \in \{0, \ldots, Q\}$ with known distribution, (iii) customer $j \in \{1, \ldots, N\}$ prefers product 1 with known probability p_j or product 2 with probability $1 - p_j$, (iv) the depot contains enough items of both products to satisfy the demands of all customers according to their preferences, (v) the actual demand and preference of each customer are revealed only when the vehicle arrives at customer's $j \in \{1, \ldots, N\}$ site, (vi) if upon arrival the vehicle does not contain enough items of the product that customer $j \in \{1, \ldots, N\}$ prefers, it is permissible to deliver items of the product that he/she does not prefer; in this case a penalty cost is incurred that is equal to π_j per item that is not preferred. Let $c_{j,j+1}$, $j = 1, \ldots, N-1$, be the travel cost from customer j to customer $j+1$. Let also c_{j0} and c_{0j}, $j = 1, \ldots, N$, be the travel cost from customer j to the depot and the travel cost from the depot to customer j, respectively. These costs can be considered as the costs of the required fuel that the vehicle needs to cover the distances between consecutive customers and the distances between customers and the depot. It is reasonable to assume that they satisfy the following properties:

$$c_{j0} = c_{0j},\ j = 1, \ldots, N \text{ (symmetric property)},$$

and

$$c_{0j} + c_{j,j+1} \geq c_{0,j+1},\ j = 1, \ldots, N-1 \text{ (triangle property)}.$$

The road network is presented in Fig. 1.

Suppose that the vehicle arrives at customer's $j \in \{1, \ldots, N\}$ site. The actual demand and the preference of the customer are revealed and the maximum possible quantity of the preferred product is delivered. Let (z_1, z_2) be the state of the process after the first visit to customer j, where z_i, $i = 1, 2$, is the number of items of product i that remain in the vehicle after the first visit to customer j and after he/she has been served according to his/her preference. There are three cases:

Case 1: $0 \leq z_1 \leq Q$, $0 \leq z_2 \leq Q$, $z_1 + z_2 \leq Q$. In this case customer j has been serviced completely according to his/her preference.

Case 2: $-Q \leq z_1 < 0$, $0 \leq z_2 \leq Q$. In this case customer j prefers product 1 and the vehicle does not have $-z_1$ items to give him/her. We separate this case into Case 2a when $z_2 < -z_1$ and Case 2b when $-z_1 \leq z_2$. In Case 2a some part of the demand of the customer can be satisfied by delivering to him/her up to z_2 items of product 2. In Case 2b the whole demand of the customer can be satisfied by delivering to him/her $-z_1$ items of product 2.

Case 3: $0 \leq z_1 \leq Q$, $-Q \leq z_2 < 0$. In this case customer j prefers product 2 and the vehicle does not have $-z_2$ items of product 2 to give him/her. We separate this case into Case 3a when $z_1 < -z_2$ and Case 3b when $-z_2 \leq z_1$. In Case 3a some part of the demand of the customer can be satisfied by delivering to him/her up to z_1 items of product 2. In Case 3b the whole demand of the customer can be satisfied by delivering to him/her $-z_2$ items of product 2.

Suppose $j \in \{1, \ldots, N-1\}$:

In Case 1 the possible actions are Action 1 and Action 2_θ, $\theta \in \{0, \ldots, Q\}$. Action 1 means that the vehicle proceeds to customer $j+1$ and Action 2_θ means that it goes to the depot, restocks with θ items of product 1 and $Q - \theta$ items of product 2 and then goes to customer $j+1$. In Case 2a the possible actions are Action $3_{(\theta_1,\theta)}$, $\theta_1 \in \{0, \ldots, z_2\}$, $\theta \in \{0, \ldots, Q + z_1 + \theta_1\}$, and Action 4_θ, $\theta \in \{0, \ldots, \mathrm{Q}\}$ Action $3_{(\theta_1,\theta)}$ means that the vehicle delivers θ_1 items of product 2 (that is not preferred) to customer j, it goes to the depot to restock with $\theta - z_1 - \theta_1$ items of product 1 and $Q + z_1 + \theta_1 - \theta$ items of product 2, returns to customer j to deliver $-z_1 - \theta_1$ owed items of product 1 and then proceeds to customer $j+1$ with θ items of product 1 and $Q + z_1 + \theta_1 - \theta$ items of product 2. Note that if Action $3_{(0,\theta)}$, $\theta \in \{0, \ldots, Q + z_1\}$ is chosen, then the vehicle does not deliver to customer j any item of product 2, while if Action $3_{(z_2,\theta)}$, $\theta \in \{0, \ldots, Q + z_1 + z_2\}$ is chosen, then the vehicle delivers to customer j all remaining items of product 2. Action 4_θ means that the vehicle goes to the depot to restock with $-z_1$ items of product 1, it returns to customer j to deliver $-z_1$ owed items of product 1, it makes a second trip to the depot to restock with θ items of product 1 and $Q - \theta$ items of product 2 and then goes to customer $j+1$. In Case 2b the possible actions are Action 5, Action 6_θ, $\theta \in \{0, \ldots, Q\}$, Action $7_{(\theta_1,\theta)}$, $\theta_1 \in \{0, \ldots, -z_1 - 1\}$, $\theta \in \{0, \ldots, Q + z_1 + \theta_1\}$ and Action 4_θ, $\theta \in \{0, \ldots, Q\}$. Action 5 means that the vehicle delivers $-z_1$ items of product 2 (that is not preferred) to customer j and proceeds to customer $j+1$. Action 6_θ means that the vehicle delivers $-z_1$ items of product 2 (that is not preferred) to customer j, goes to the depot to restock with θ items of product 1 and $Q - \theta$ items of product 2 and then goes to customer $j+1$. Action $7_{(\theta_1,\theta)}$ means that the vehicle delivers θ_1 items of product 2 (that is not preferred) to customer j, it goes to the depot to restock with $\theta - z_1 - \theta_1$ items of product 1 and $Q + z_1 + \theta_1 - \theta$ items of product 2, it returns to customer j to deliver $-z_1 - \theta_1$ owed items of product 1 and then proceeds to customer $j+1$ with θ items of product 1 and $Q + z_1 + \theta_1 - \theta$ items of product 2. Note that if Action $7_{(0,\theta)}$, $\theta \in \{0, \ldots, Q + z_1\}$ is chosen, then the vehicle does not deliver to customer j any item of product 2. Actions $3_{(\theta_1,\theta)}$ and $7_{(\theta_1,\theta)}$ cause a penalty cost that is equal to $\pi_j \theta_1$ while Actions 5 and 6_θ cause a penalty cost that is equal to $-\pi_j z_1$. It is assumed that if Action $3_{(\theta_1,\theta)}$ or Action 4_θ or Action $7_{(\theta_1,\theta)}$ is selected, there is no extra demand when the vehicle returns to customer j, i.e. ξ_j remains 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 Case 2a the only possible action is Action 8 which means that the vehicle goes to the depot to restock with $-z_1$ items of product 1, returns to customer N to deliver $-z_1$ owed items of product 1 and then goes again to the depot to terminate its route. In Case 2b the possible actions are Action 8 and Action 9. Action 9 means that the vehicle delivers $-z_1$ items of product 2 (that is not preferred) to customer N and then goes to the depot to terminate its route. If Action 9 is selected a penalty cost is incurred that is equal to $-z_1 \pi_N$. It is assumed that if Action 8 is selected, there is no extra demand when the vehicle returns to customer N, i.e. ξ_N remains unaltered.

Note that in Case 3a and Case 3b for $j \in \{1, \ldots, N\}$ the possible actions are the same as in Case 2a and Case 2b by taking into account that there is shortage for items of product 2 instead of product 1. Our goal is to determine the optimal routing strategy of the vehicle that serves all customers. This routing strategy minimizes the expected total cost from the beginning of the route until its end. The total cost consists of travel costs between consecutive customers and between customers and the depot and penalty costs that incur when items of the product that is not preferred are delivered to customers. The optimal routing strategy can be found by implementing a suitable dynamic programming algorithm.

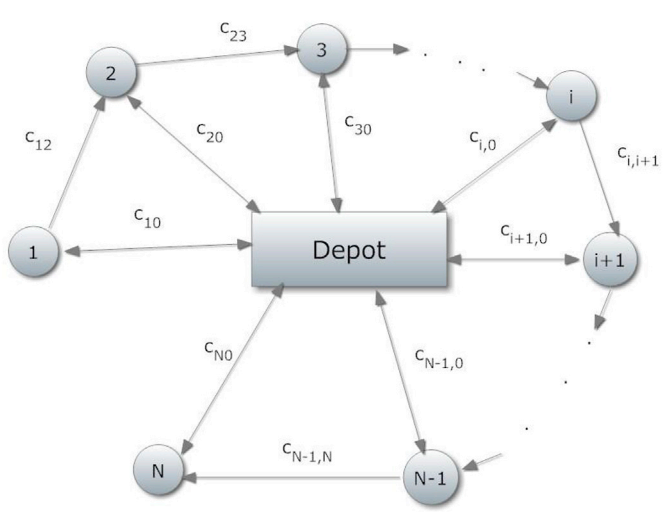

Fig. 1. The road network.

2.2. Dynamic programming equations

Let $f_j(z_1, z_2)$ 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 (z_1, z_2) 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)–(3) for Case 1, Case 2a and Case 2b. For Case 3a and Case 3b the dynamic programming equations are the same as (2) and (3) if we interchange z_1 and z_2.

If $0 \le z_1 \le Q,\ 0 \le z_2 \le Q,\ z_1 + z_2 \le Q$, then

$$f_j(z_1, z_2) = \min\{A_j(z_1, z_2),\ B_j\}, \tag{1}$$

where,

$$A_j(z_1, z_2) = c_{j,j+1} + p_{j+1} E f_{j+1}(z_1 - \xi_{j+1}, z_2) + (1 - p_{j+1}) E f_{j+1}(z_1, z_2 - \xi_{j+1}),$$

$$B_j = c_{j0} + c_{0,j+1} + \min_{0 \le \theta \le Q} [p_{j+1} E f_{j+1}(\theta - \xi_{j+1}, Q - \theta) + (1 - p_{j+1}) E f_{j+1}(\theta, Q - \theta - \xi_{j+1})].$$

If $-Q \le z_1 < 0,\ 0 \le z_2 \le Q,\ z_2 < -z_1$ then

$$f_j(z_1, z_2) = \min\{C_j(z_1, z_2),\ D_j\}, \tag{2}$$

where,

$$C_j(z_1, z_2) = 2c_{j0} + c_{j,j+1} + \min_{(\theta_1,\theta):0 \le \theta_1 \le z_2, 0 \le \theta \le Q + z_1 + \theta_1} [\pi_j \theta_1 + p_{j+1} E f_{j+1}(\theta - \xi_{j+1}, Q + z_1 + \theta_1 - \theta) + (1 - p_{j+1}) E f_{j+1}(\theta, Q + z_1 + \theta_1 - \theta - \xi_{j+1})],$$

$$D_j = 3c_{j0} + c_{0,j+1} + \min_{0 \le \theta \le Q} [p_{j+1} E f_{j+1}(\theta - \xi_{j+1}, Q - \theta) + (1 - p_{j+1}) E f_{j+1}(\theta, Q - \theta - \xi_{j+1})].$$

If $-Q \le z_1 < 0,\ 0 \le z_2 \le Q,\ -z_1 \le z_2$, then

$$f_j(z_1, z_2) = \min\{E_j(z_1, z_2),\ F_j(z_1),\ G_j(z_1),\ D_j\}, \tag{3}$$

where,

$$E_j(z_1, z_2) = -\pi_j z_1 + c_{j,j+1} + p_{j+1} E f_{j+1}(-\xi_{j+1}, z_1 + z_2) + (1 - p_{j+1}) E f_{j+1}(0, z_1 + z_2 - \xi_{j+1}),$$

$$F_j(z_1) = -\pi_j z_1 + c_{j0} + c_{0,j+1} + \min_{0 \le \theta \le Q} [p_{j+1} E f_{j+1}(\theta - \xi_{j+1}, Q - \theta) + (1 - p_{j+1}) E f_{j+1}(\theta, Q - \theta - \xi_{j+1})],$$

$$G_j(z_1) = 2c_{j0} + c_{j,j+1} + \min_{(\theta_1,\theta):0 \le \theta_1 < -z_1, 0 \le \theta \le Q + z_1 + \theta_1} [\pi_j \theta_1 + p_{j+1} E f_{j+1}(\theta - \xi_{j+1}, Q + z_1 + \theta_1 - \theta) + (1 - p_{j+1}) E f_{j+1}(\theta, Q + z_1 + \theta_1 - \theta - \xi_{j+1})]. \tag{4}$$

The boundary conditions are given below for Case 1 and for Case 2a and Case 2b. For Case 3a and Case 3b the boundary conditions are the same as for Case 2a and Case 2b if we interchange z_1 and z_2.

If $0 \le z_1 \le Q,\ 0 \le z_2 \le Q,\ z_1 + z_2 \le Q$, then

$$f_N(z_1, z_2) = c_{N0}. \tag{5}$$

If $-Q \le z_1 < 0,\ 0 \le z_2 \le Q,\ z_2 < -z_1$ then

$$f_N(z_1, z_2) = 3c_{N0}. \tag{6}$$

If $-Q \le z_1 < 0,\ 0 \le z_2 \le Q,\ -z_1 \le z_2$, then

$$f_N(z_1, z_2) = \min\{3c_{N0}, c_{N0} - z_1 \pi_N\}. \tag{7}$$

The minimum total expected cost during a visit cycle is equal to

$$f_0 = c_{01} + \min_{0 \le z \le Q} [p_1 E f_1(z - \xi_1, Q - z) + (1 - p_1) E f_1(z, Q - z - \xi_1)].$$

In the above equations the expected values are taken with respect to the random variables $\xi_j,\ j = 1, \ldots, N$. The terms $A_j(z_1, z_2)$ and B_j in the right-hand-side of Eq. (1) correspond to Action 1 and Actions 2_θ ($\theta \in \{0, \ldots, Q\}$), respectively. The terms $C_j(z_1, z_2)$ and D_j in the right-hand-side of Eq. (2) correspond to Actions $3_{(\theta_1,\theta)}$ ($\theta_1 \in \{0, \ldots, z_2\}$, $\theta \in \{0, \ldots, Q + z_1 + \theta_1\}$) and Actions 4_θ ($\theta \in \{0, \ldots, Q\}$), respectively. The terms $E_j(z_1, z_2)$, $F_j(z_1)$, $G_j(z_1)$, D_j in the right-hand-side of Eq. (3) correspond to Action 5, Actions 6_θ ($\theta \in \{0, \ldots, Q\}$), Actions $7_{(\theta_1,\theta)}$ ($\theta_1 \in \{0, \ldots, -z_1 - 1\}$, $\theta \in \{0, \ldots, Q + z_1 + \theta_1\}$), Actions 4_θ ($\theta \in \{0, \ldots, Q\}$), respectively. The terms in the curly brackets in the right-hand-side of Eq. (7) correspond to Action 8 and Action 9, respectively. Lemma 1 below will be used in the proof of Theorem 1 that describes the structure of the optimal routing strategy.

2.3. Structure of the optimal policy

Lemma 1. $f_j(z_1, z_2)$, $j = 1, \ldots, N$, is non-increasing with respect to z_1 and z_2.

Proof. The proof is by induction on *j*. From (5), (6), (7) it can be seen that $f_N(z_1, z_2)$ is non-increasing in z_1 and z_2. Assuming that $f_{j+1}(z_1, z_2)$ is non-increasing in z_1 and z_2, we will show that $f_j(z_1, z_2)$ is non-increasing in z_1 and z_2. We will restrict ourselves to Case 1 and Case 2, since Case 3 is similar to Case 2. Let some fixed $z_1 \in \{-Q, \ldots, Q\}$. In view of the induction hypothesis, it follows from (1), (2), (3) that, to prove that $f_j(z_1, z_2)$ is non-increasing in z_2, it is enough to show that $f_j(-z_2, z_2) \le f_j(-z_2, z_2 - 1)$, $z_2 \in \{1, \ldots, Q\}$. This inequality holds if $G_j(-z_2) \le C_j(-z_2, z_2 - 1)$, $z_2 \in \{1, \ldots, Q - 1\}$, which holds as equality. Let some fixed $z_2 \in \{0, \ldots, Q\}$. In view of the induction hypothesis and Equations (1)–(3), it follows that, to prove that $f_j(z_1, z_2)$ is non-increasing in z_1, it is enough to prove that

$$G_j(z_1 + 1) \le G_j(z_1), \quad -Q \le z_1 < -1, \tag{8}$$

$$f_j(0, z_2) \le f_j(-1, z_2), \quad 1 \le z_2 \le Q, \tag{9}$$

$$f_j(0, 0) \le f_j(-1, 0), \tag{10}$$

$$f_j(z_1, -z_1) \le f_j(z_1 - 1, -z_1), \quad -Q + 1 \le z_1 \le -1. \tag{11}$$

To prove (8), in view of the induction hypothesis, it is enough to show that $H(-z_1 - 1, \theta) \ge H(-z_1 - 2, \theta)$, $\theta \in \{0, \ldots, -z_1 - 1\}$, where $H(\theta_1, \theta)$, $0 \le \theta_1 < -z_1$, $0 \le \theta \le Q + z_1 + \theta_1$, is the quantity in the square brackets in the right-hand-side of (4). It can be readily checked that the last inequality holds. Inequality (9) is equivalent to

$$\min\{A_j(0, z_2), B_j\} \le \min\{E_j(-1, z_2), F_j(-1), G_j(-1), D_j\},\ 1 \le z_2 \le Q.$$

The above inequality holds since $A_j(0, z_2) \le E_j(-1, z_2)$, $B_j \le F_j(-1)$, $B_j \le G_j(-1)$, $B_j \le D_j$. Inequality (10) is equivalent to $\min\{A_j(0,0), B_j\} \le \min\{C_j(-1,0), D_j\}$ This inequality holds since $B_j \le C_j(-1, 0)$, $B_j \le D_j$. Inequality (11) is equivalent to $\min\{E_j(z_1, -z_1), F_j(z_1), G_j(z_1), D_j\} \le \min\{C_j(z_1 - 1, -z_1), D_j\}$ This inequality holds since, in view of induction hypothesis, $G_j(z_1) \le C_j(z_1 - 1, -z_1)$.

Theorem 1. For each customer $j \in \{1, \ldots, N-1\}$ the structure of the optimal routing strategy is described in the following five cases:

(i) For $z_1 \in \{0, \ldots, Q\}$ there exists a critical integer $s_1(z_1) \ge 0$ such that if $z_2 \in \{s_1(z_1), \ldots, Q - z_1\}$ the optimal action is Action 1, while if $z_2 \in \{0, \ldots, s_1(z_1) - 1\}$ the optimal action is Action 2_θ for some $\theta \in \{0, \ldots, Q\}$. Moreover, $s_1(z_1)$ is non-increasing in z_1.

(ii) For $z_2 \in \{0, \ldots, Q - 1\}$ there exists a critical integer $s_2(z_2) \in \{-Q, \ldots, -z_2 - 1\}$ such that if $z_1 \in \{s_2(z_2), \ldots, -z_2 - 1\}$ the optimal action is Action $3_{(\theta_1,\theta)}$ for some $\theta_1 \in \{0, \ldots, z_2\}$ and some $\theta \in \{0, \ldots, Q + z_1 + \theta_1\}$, while if $z_1 \in \{-Q, \ldots, s_2(z_2) - 1\}$ the optimal action is Action 4_θ for some $\theta \in \{0, \ldots, Q\}$. Moreover, $s_2(z_2)$ is non-increasing in z_2.

(iii) For $z_2 \in \{1, \ldots, Q\}$ there exists a critical integer $s_3(z_2) \in \{-z_2, \ldots, -1\}$ such that if $z_1 \in \{s_3(z_2), \ldots, -1\}$ the optimal action is Action 5 or Action 6_θ for some $\theta \in \{0, \ldots, Q\}$ or Action $7_{(\theta_1,\theta)}$ for some $\theta_1 \in \{0, \ldots, -z_1 - 1\}$ and some $\theta \in \{0, \ldots, Q + z_1 + \theta_1\}$, while if $z_1 \in \{-z_2, \ldots, s_3(z_2) - 1\}$ the optimal action is Action 4_θ for some $\theta \in \{0, \ldots, Q\}$. Moreover, $s_3(z_2)$ is non-increasing in z_2.

(iv) For $z_1 \in \{0, \ldots, Q-1\}$ there exists a critical integer $s_4(z_1) \in \{-Q, \ldots, -z_1 - 1\}$ such that if $z_2 \in \{s_4(z_1), \ldots, -z_1 - 1\}$ the optimal action is Action $3_{(\theta_1,\theta)}$ for some $\theta_1 \in \{0, \ldots, z_1\}$ and some $\theta \in \{0, \ldots, Q + z_2 + \theta_1\}$, while if $z_2 \in \{-Q, \ldots, s_4(z_1) - 1\}$ the optimal action is Action 4_θ for some $\theta \in \{0, \ldots, Q\}$. Moreover, $s_4(z_1)$ is non-increasing in z_1.

(v) For $z_1 \in \{1, \ldots, Q\}$ there exists a critical integer $s_5(z_1) \in \{-z_1, \ldots, -1\}$ such that if $z_2 \in \{s_5(z_1), \ldots, -1\}$ the optimal action is Action 5 or Action 6_θ for some $\theta \in \{0, \ldots, Q\}$ or Action $7_{(\theta_1,\theta)}$ for some $\theta_1 \in \{0, \ldots, -z_2 - 1\}$ and some $\theta \in \{0, \ldots, Q + z_2 + \theta_1\}$, while if $z_2 \in \{-z_1, \ldots, s_5(z_1) - 1\}$ the optimal action is Action 4_θ for some $\theta \in \{0, \ldots, Q\}$. Moreover, $s_5(z_1)$ is non-increasing in z_1.

Proof. From Lemma 1 it follows that $A_j(z_1, z_2)$ is non-increasing in z_1 and z_2. Part (i) is a direct consequence of this result. From Lemma 1 it follows that $C_j(z_1, z_2)$ is non-increasing in z_1. It can also be seen that $C_j(z_1, z_2)$ is non-increasing in z_2. Part (ii) is a direct consequence of these results. From Lemma 1 it follows that $E_j(z_1, z_2)$ is non-increasing in z_1 and z_2. In the proof of Lemma 1 it has been shown that $G_j(z_1)$ is non-increasing in z_1. It can also easily be seen that $F_j(z_1)$ is non-increasing in z_1. Part (iii) is a direct consequence of these results. Part (iv) and Part (v) can be proved in a similar way as Part (ii) and Part (iii), respectively.

2.4. Special-purpose dynamic programming algorithm

The optimal routing strategy, i.e. the critical integers $s_1(z_1)$, $z_1 \in \{0, \ldots, Q\}$, $s_2(z_2)$, $z_2 \in \{0, \ldots, Q-1\}$, $s_3(z_2)$, $z_2 \in \{1, \ldots, Q\}$, $s_4(z_1)$, $z_1 \in \{0, \ldots, Q-1\}$, $s_5(z_1)$, $z_1 \in \{1, \ldots, Q\}$, for each $j \in \{1, \ldots, N-1\}$, can be found by a special-purpose dynamic programming algorithm, that takes into account the structure of the optimal routing strategy as given in Theorem 1. The part of this algorithm that computes the critical integers $s_1(z_1)$, $z_1 \in \{0, \ldots, Q\}$, $s_2(z_2)$, $z_2 \in \{0, \ldots, Q-1\}$, $s_3(z_2)$, $z_2 \in \{1, \ldots, Q\}$ is presented below. The complete special-purpose dynamic programming algorithm includes the computation of the critical integers $s_4(z_1)$, $z_1 \in \{0, \ldots, Q-1\}$ and $s_5(z_1)$, $z_1 \in \{1, \ldots, Q\}$ that is similar to the computation of the critical integers $s_2(z_2)$, $z_2 \in \{0, \ldots, Q-1\}$ and $s_3(z_2)$, $z_2 \in \{1, \ldots, Q\}$, respectively.

Algorithm for the determination of the critical integers $s_1(z_1)$, $z_1 \in \{0, \ldots, Q\}$, $s_2(z_2)$, $z_2 \in \{0, \ldots, Q-1\}$, $s_3(z_2)$, $z_2 \in \{1, \ldots, Q\}$

Step 0 Set $f_N(z_1, z_2) = c_{N0}$ if $z_1, z_2 \in \{0, \ldots, Q\}$, $z_1 + z_2 \leq Q$, $f_N(z_1, z_2) = 3c_{N0}$ if $z_1 \in \{-Q, \ldots, -1\}$, $z_2 \in \{0, \ldots, Q\}$, $z_2 < -z_1$, $f_N(z_1, z_2) = \min\{3c_{N0}, c_{N0} - z_1\pi_N\}$ if $z_1 \in \{-Q, \ldots, -1\}$, $z_2 \in \{0, \ldots Q\}$, $-z_1 \leq z_2$. Set $j = N-1$.

Step 1 (Determination of critical integers $s_1(z_1)$, $z_1 \in \{0, \ldots, Q\}$)
Compute B_j.
For $z_1 = 0, \ldots, Q$ do the following:
For $z_2 = Q - z_1, Q - z_1 - 1, \ldots$ compute $A_j(z_1, z_2)$ until $A_j(z_1, z_2) > B_j$ or $z_2 = -1$.
Set $s_1(z_1) = z_2 + 1$.
Set $f_j(z_1, z_2) = A_j(z_1, z_2)$, $z_2 \in \{s_1(z_1), \ldots, Q - z_1\}$ and $f_j(z_1, z_2) = B_j$, $z_1 \in \{0, \ldots, s_1(z_1) - 1\}$

Step 2 (Determination of critical integers $s_2(z_2)$, $z_2 \in \{0, \ldots, Q-1\}$)
$D_j = 2c_{j0} + B_j$.
For $z_2 = 0, \ldots, Q-1$ do the following:
For $z_1 = -z_2 - 1, -z_2 - 2, \ldots$ compute $C_j(z_1, z_2)$ until $D_j < C_j(z_1, z_2)$ or $z_1 = -Q - 1$.
Set $s_2(z_2) = z_1 + 1$.
Set $f_j(z_1, z_2) = C_j(z_1, z_2)$, $z_1 \in \{s_2(z_2), \ldots, -z_2 - 1\}$ and $f_j(z_1, z_2) = D_j$, $z_1 \in \{-Q, \ldots, s_2(z_2) - 1\}$

Step 3 (Determination of critical integers $s_3(z_2)$, $z_2 \in \{1, \ldots, Q\}$)
For $z_2 = 1, \ldots, Q$ do the following:
For $z_1 = -1, -2, \ldots$ compute $E_j(z_1, z_2), F_j(z_1), G_j(z_1)$ until $D_j < \min\{E_j(z_1, z_2), F_j(z_1), G_j(z_1)\}$ or $z_1 = -z_2 - 1$.
Set $s_2(z_2) = z_1 + 1$.
Set $f_j(z_1, z_2) = \min\{E_j(z_1, z_2), F_j(z_1), G_j(z_1)\}$, $z_1 \in \{s_3(z_2), \ldots, -1\}$ and
$f_j(z_1, z_2) = D_j$, $z_1 \in \{-z_2, \ldots, s_3(z_2) - 1\}$

Step 4 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 routing strategy described in Theorem 1. The complexity of this algorithm can be calculated by considering Definition 7.1 in Sipser (2013) and it is found to be $O(NQ^3)$. It is more efficient than the initial dynamic programming algorithm since it requires less computations. For example, for $j = 1, \ldots, N-1$, the quantities $A_j(z_1, z_2)$, $z_2 \in \{0, \ldots, s_1(z_1) - 2\}$, for $z_1 \in \{0, \ldots, Q\}$ are not computed, while these quantities are computed in the initial dynamic programming algorithm. In Section 4 we will compare the computations times of these algorithms in a numerical example.

3. The problem when the demands are continuous random variables

3.1. The optimal routing strategy with continuous demands

We modify the problem that we introduced in Section 2 by assuming that the demands ξ_j, $j = 1, \ldots, N$, of the customers are continuous random variables and take values in the interval $[0, Q]$ with probability density function $\phi_j(x)$. A practical example with continuous demands could be the delivery of two different kinds of building materials, for example lime and pebble. The states (z_1, z_2) of the process, where $0 \leq z_1, z_2 \leq Q$, $z_1 + z_2 \leq Q$ or $-Q \leq z_1 < 0$, $0 \leq z_2 \leq Q$ or $-Q \leq z_2 < 0$, $0 \leq z_1 \leq Q$, after the first visit at a customer's site and Action 1, Actions 2_θ $(0 \leq \theta \leq Q)$, Actions $3_{(\theta_1,\theta)}$ $(0 \leq \theta_1 \leq z_2,\ 0 \leq \theta \leq Q + z_1 + \theta_1)$, Actions 4_θ $(0 \leq \theta \leq Q)$, Action 5, Actions 6_θ $(0 \leq \theta \leq Q)$, Actions $7_{(\theta_1,\theta)}$ $(0 \leq \theta_1 < -z_1,\ 0 \leq \theta \leq Q + z_1 + \theta_1)$, Action 8, Action 9 are the same as those defined in Section 2. The minimum expected future cost $f_j(z_1,\ z_2)$ for $j = 1, \ldots, N$, satisfies the dynamic programming equations (1)–(3) and the boundary conditions (5)–(7). The structure of the optimal routing strategy is the same as in the case of discrete demands and is given in the theorem below.

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

(i) For $z_1 \in [0,\ Q]$ there exists a critical number $s_1(z_1) \geq 0$ such that if $z_2 \in [s_1(z_1),\ Q - z_1\,]$ the optimal action is Action 1, while if $z_2 \in [0, s_1(z_1))$ the optimal action is Action 2_θ for some $\theta \in [0, Q]$. Moreover, $s_1(z_1)$ is non-increasing in z_1.

(ii) For $z_2 \in [0, Q)$ there exists a critical number $s_2(z_2) \in [-Q, -z_2)$ such that if $z_1 \in [s_2(z_2), -z_2)$ the optimal action is Action $3_{(\theta_1,\theta)}$ for some $\theta_1 \in [0, z_2]$ and some $\theta \in [0, Q + z_1 + \theta_1]$, while if $z_1 \in [-Q, s_2(z_2))$ the optimal action is Action 4_θ for some $\theta \in [0, Q]$. Moreover, $s_2(z_2)$ is non-increasing in z_2.

(iii) For $z_2 \in (0, Q]$ there exists a critical number $s_3(z_2) \in [-z_2, 0)$ such that if $z_1 \in [s_3(z_2), 0)$ the optimal action is Action 5 or Action 6_θ for some $\theta \in [0, Q]$ or Action $7_{(\theta_1,\theta)}$ for some $\theta_1 \in [0, -z_1)$ and some $\theta \in [0, Q + z_1 + \theta_1]$ or Action 4_θ, while if $z_1 \in [-z_2, s_3(z_2))$ the optimal action is Action 4_θ for some $\theta \in [0, Q]$. Moreover, $s_3(z_2)$ is non-increasing in z_2.

(iv) For $z_1 \in [0, Q)$ there exists a critical number $s_4(z_1) \in [-Q, -z_1)$ such that if $z_2 \in [s_4(z_1), -z_1)$ the optimal action is Action $3_{(\theta_1,\theta)}$ for some $\theta_1 \in [0, z_1]$ and some $\theta \in [0, Q + z_2 + \theta_1]$, while if $z_2 \in [-Q, s_4(z_1))$ the optimal action is Action 4_θ for some $\theta \in [0, Q]$. Moreover, $s_4(z_1)$ is non-increasing in z_1.

(v) For $z_1 \in (0, Q]$ there exists a critical number $s_5(z_1) \in [-z_1, 0)$ such that if $z_2 \in [s_5(z_1), 0)$ the optimal action is Action 5 or Action 6_θ for some $\theta \in [0,\ Q]$ or Action $7_{(\theta_1,\theta)}$ for some $\theta_1 \in [0,\ -z_2)$ and some $\theta \in [0,\ Q + z_2 + \theta_1]$, while if $z_2 \in [-z_1, s_5(z_1))$ the optimal action is Action 4_θ for some $\theta \in [0,\ Q]$ Moreover, $s_5(z_1)$ is non-increasing in z_1.

3.2. Discretization of the state space

The state space after the first visit of the vehicle at customer's $j \in \{1, \ldots, N\}$ site is the set $S = \{(z_1, z_2) : -Q \le z_1 < 0, 0 \le z_2 \le Q\} \cup \{(z_1, z_2) : 0 \le z_1, z_2 \le Q, z_1 + z_2 \le Q\} \cup \{(z_1, z_2) : 0 \le z_1 \le Q, \quad -Q \le z_2 < 0\}$ A discretization 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 discretize S by restricting our attention only to its points that belong to the set

$$\tilde{S} = \{(k\rho, l\rho) : k = -1, \ldots, -Q/\rho,\ l = 0, \ldots, Q/\rho\} \cup \{(k\rho, l\rho) : k, l = 0, \ldots, Q/\rho \text{ s.t. } k + l \le Q/\rho\} \cup \{(k\rho, l\rho) : k = 0, \ldots, Q/\rho,\ l = -1, \ldots, -Q/\rho\}.$$

The minimum expected cost $f_N(k\rho, l\rho),\ (k\rho, l\rho) \in \tilde{S}$, is found by using (5)–(7) with $z_1 = k\rho,\ z_2 = l\rho$. 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. The parameters θ_1 and θ in these equations take values in finite sets. For example in (4) the pair (θ_1, θ) takes values in the set $B = \{(u\rho, v\rho) : u = 0, \ldots, -k-1;\ v = 0, \ldots, Q/\rho + k + u\}$ The expectations are computed approximately. For example $G_j(k\rho)$ is computed approximately as follows:

$$G_j(k\rho) = 2c_{j0} + c_{j,j+1} + \min_{(\theta_1,\theta)\in B}\left[\pi_j\theta_1 + p_{j+1}\sum_{x=0}^{Q/\rho-1} f_{j+1}(\theta - x\rho, Q + k\rho + \theta_1 - \theta)\phi_{j+1}(x\rho)\rho + (1 - p_{j+1})\sum_{x=0}^{Q/\rho-1} f_{j+1}(\theta, Q + k\rho + \theta_1 - \theta - x\rho)\phi_{j+1}(x\rho)\rho\right].$$

3.3. Special-purpose dynamic programming algorithm

As in the case of discrete demands, the optimal routing strategy, i.e. the critical numbers $s_1(k\rho),\ k = 0, \ldots, Q/\rho,\ \ s_2(l\rho),\ l = 0, \ldots, Q/\rho - 1,\ s_3(l\rho),\ l = 1, \ldots, Q/\rho,\ s_4(k\rho),\ k = 0, \ldots, Q/\rho - 1, s_5(k\rho),\ k = 1, \ldots, Q/\rho$ 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 routing strategy as given in Theorem 2. The part of this algorithm that computes the critical numbers $s_1(k\rho),\ k = 0, \ldots, Q/\rho$, $s_2(l\rho),\ l = 0, \ldots, Q/\rho - 1,\ s_3(l\rho),\ l = 1, \ldots, Q/\rho$, is presented below. The complete special-purpose dynamic programming algorithm includes the computation of the critical numbers $s_4(k\rho),\ k = 0, \ldots, Q/\rho - 1$ and $s_5(l\rho),\ l = 1, \ldots, Q/\rho$ that is similar to the computation of the critical numbers $s_2(l\rho),\ l = 0, \ldots, Q/\rho - 1$ and $s_3(l\rho),\ l = 1, \ldots, Q/\rho$, respectively.

Algorithm for the determination of the critical numbers $s_1(k\rho),\ k = 0, \ldots, Q/\rho, s_2(l\rho),\ l = 0, \ldots, Q/\rho - 1,\ s_3(l\rho),\ l = 1, \ldots, Q/\rho$

Step 0 Set $f_N(k\rho, l\rho) = c_{N0}$ if $k, l \in \{0, \ldots, Q/\rho\}$ and $k + l \le Q/\rho$. Set $f_N(k\rho, l\rho) = 3c_{N0}$ if $k = -1, \ldots, -Q/\rho,\ l = 0, \ldots, Q/\rho,\ l < -k$. Set $f_N(k\rho, l\rho) = \min\{3c_{N0}, c_{N0} - k\rho\pi_N\}$ if $k = -1, \ldots, \quad -Q/\rho,\ l = 0, \ldots, Q/\rho,\ l \ge k$. Set $j = N - 1$.

Step 1 (Determination of critical numbers $s_1(k\rho),\ k = 0, \ldots, Q/\rho$)
Compute B_j.
For $k = 0, \ldots, Q/\rho$ do the following:
For $z_2 = Q - k\rho, Q - (k+1)\rho, \ldots$ compute $A_j(k\rho, z_2)$ until $A_j(k\rho, z_2) > B_j$ or $z_2 = -\rho$.
Set $s_1(k\rho) = z_2 + \rho$.
Set $f_j(k\rho, l\rho) = A_j(k\rho, l\rho),\ s_1(k\rho)/\rho \le l \le Q/\rho$.
Set $f_j(k\rho, l\rho) = B_j,\ 0 \le l < s_1(k\rho)/\rho$.

Step 2 (Determination of critical numbers $s_2(l\rho),\ l = 0, \ldots, Q/\rho - 1$)
$D_j = 2c_{j0} + B_j$.
For $l = 0, \ldots, Q/\rho - 1$ do the following:
For $z_1 = -l\rho - \rho,\ -l\rho - 2\rho, \ldots$ compute $C_j(z_1, l\rho)$ until $D_j < C_j(z_1, l\rho)$ or $z_1 = -Q - \rho$.
Set $s_2(l\rho) = z_1 + \rho$.
Set $f_j(k\rho, l\rho) = C_j(k\rho, l\rho),\ s_2(l\rho)/\rho \le k \le -l - 1$.
Set $f_j(k\rho, l\rho) = D_j,\ -Q/\rho \le k \le s_2(l\rho)/\rho - 1$.

Step 3 (Determination of critical numbers $s_3(l\rho),\ l = 1, \ldots, Q/\rho$)
For $l = 1, \ldots, Q/\rho$ do the following:
For $z_1 = -\rho, -2\rho, \ldots$ compute $E_j(z_1, l\rho), F_j(z_1), G_j(z_1)$ until $D_j < \min\{E_j(z_1, l\rho), F_j(z_1), G_j(z_1)\}$ or $z_1 = -l\rho - \rho$.
Set $s_3(z_2) = z_1 + \rho$.
Set $f_j(k\rho, l\rho) = \min\{E_j(k\rho, l\rho), F_j(k\rho), G_j(k\rho)\},\ s_3(l\rho)/\rho \le k \le -1$.
$f_j(k\rho, l\rho) = D_j,\ -l \le k \le s_3(l\rho)/\rho - 1$.

Step 4 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 routing strategy described in Theorem 2. Its complexity is $O(N[Q/\rho]^3)$ (see Definition 7.1 in Sipser (2013)). It requires less computations than the initial dynamic programming algorithm. For example, for $j = 1, \ldots, N-1$, the quantities $A_j(k\rho, l\rho),\ l \in \{0, \ldots, s_1(k\rho)/\rho - 2\}$, for $k \in \{0, \ldots, Q/\rho\}$ are not computed, while these quantities are computed in the initial dynamic programming algorithm. A numerical example is presented in Section 4 that shows that the difference of the computation times of these algorithms is significant especially for high values of the number of customers *N*.

4. Numerical results

In the following numerical results, we implemented the initial dynamic programming algorithm and the special-purpose dynamic programming algorithm by running the corresponding Matlab programs on a personal computer equipped with an Intel Core i5-4460, 3.2 GHz processor and 16 GB of RAM. In Example 1, we assume that the demands of the customers for the products are discrete random variables and in Example 2, we assume that the demands of the customers for the products are continuous random variables. These examples confirm the structural results presented in Theorem 1 and in Theorem 2.

Example 1. Suppose that $N = 8$ and $Q = 12$. The travel costs between customer j and $j+1,\ j \in \{1, \ldots, 7\}$, are given by: $c_{12} = 10,\ c_{23} = 12,\ c_{34} = 10,\ c_{45} = 14,\ c_{56} = 12,\ c_{67} = 15$ and $c_{78} = 12$. The travel costs between customers $j, j = 1, \ldots, 8$, and the depot are given by: $c_{10} = 12,\ c_{20} = 11,\ c_{30} = 9,\ c_{40} = 10,\ c_{50} = 13,\ c_{60} = 11,\ c_{70} = 14$ and $c_{80} = 9$. Note that these costs satisfy the triangle inequality. We assume that the penalty costs $\pi_1, \ldots, \pi_8$ incurred when an item of the non-preferred product is delivered to customers 1,…,8 instead of the preferred one are elements of the row vector $\pi = (4,\ 3,\ 6,\ 5,\ 3,\ 5,\ 4,\ 6)$. We further assume that the demand ξ_j of each customer$j \in \{1, \ldots, 8\}$ is a random variable which follows the binomial distribution $Bin(Q,\ 0.4)$, i.e.

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

Note that the binomial distribution has been used to model demands in vehicle routing problems, for example in Golden and Yee (1979) and in Haugland et al. (2007). We assume that the probabilities $p_1, \ldots, p_8$ that customers 1,…,8 prefer product 1 are elements of the row vector $p = (0.6,\ 0.7,\ 0.5,\ 0.4,\ 0.5,\ 0.6,\ 0.8,\ 0.4)$. In Figs. 2 and 3, we present the optimal decisions for customers 3 and 6. If $0 \le z_1 \le Q$, $0 \le z_2 \le Q$, $z_1 + z_2 \le Q$, the action of proceeding directly to next customer (Action 1) is denoted by blue circles and the action of going to the depot for restocking with product loads θ and $Q-\theta$ and then going to the next customer (Action 2_θ) is denoted by red squares. If $-Q \le z_1 < 0$, $0 \le z_2 \le Q$, or if $0 \le z_1 \le Q$, $-Q \le z_2 < 0$ and if $z_2 < -z_1$ or if $z_1 < -z_2$ we use cyan stars for action $3_{(\theta_1,\theta)}$ (or action $3_{(\theta_2,\theta)}$, where θ_2 is the number of items of the non-preferred product 1 that are delivered) that corresponds to quantity $C_j(z_1,z_2)$ and green diamonds for action 4_θ that corresponds to quantity D_j. If $z_2 \ge -z_1$ or $z_1 \ge -z_2$, we use yellow hexagrams for action 5, that corresponds to the quantity $E_j(z_1,z_2)$, magenta x-marks for action 6_θ that corresponds to quantity $F_j(z_1)$ (or $F_j(z_2)$) and black upper triangles for action $7_{(\theta_1,\theta)}$ (or action $7_{(\theta_2,\theta)}$, where θ_2 is the number of the non-preferred items of product 1 that are delivered) which correspond to the quantity $G_j(z_1)$ (or $G_j(z_2)$).

The value of the minimum total expected cost f_0 is found to be approximately equal to 165.61. The computation time of the special-purpose dynamic programming algorithm is 0.625 seconds. It is considerably smaller than the computation time of the initial dynamic programming algorithm which is 3.062 seconds.

Both algorithms enable us to determine the optimal values of θ_1 (or θ_2) and θ when the optimal actions are the actions $3_{(\theta_1,\theta)}$ (or $3_{(\theta_2,\theta)}$) and $7_{(\theta_1,\theta)}$ (or $7_{(\theta_2,\theta)}$) and the optimal value of θ when the optimal actions are the actions 2_θ, 4_θ and 6_θ. For example, for customer 3, if the state is $(z_1,z_2)=(2,-5)$, then the optimal action for the vehicle is the action $3_{(\theta_2,\theta)}$ with $\theta_2=2$ and $\theta=7$. According to this action, the vehicle delivers $\theta_2=2$ items of non-preferred product 1 to customer 3, goes to the depot to restock with $\theta - z_2 - \theta_2 = 7-(-5)-2 = 10$ items of product 2 and $Q + z_2 + \theta_2 - \theta = 12-5+2-7 = 2$ items of product 1, returns to customer 3 to deliver $-z_2-\theta_2 = -(-5)-2 = 3$ owed items of product 2 and then proceeds to customer 4 with $\theta = 7$ items of product 2 and 2 items of product 1. If, again for customer 3, the state is $(z_1,z_2)=(-4,\ 6)$, then the optimal action for the vehicle is the action $7_{(\theta_1,\theta)}$ with $\theta_1=3$ and $\theta=8$. According to this action, the vehicle delivers $\theta_1=3$ items of non-preferred product 2 to customer 3, goes to the depot to restock with $\theta - z_1 - \theta_1 = 8-(-4)-3 = 9$ items of product 1 and $Q + z_1 + \theta_1 - \theta = 12-4+3-8 = 3$ items of product 2, returns to customer 3 to deliver $-z_1-\theta_1 = -(-4)-3 = 1$ owed item of product 1 and then proceeds to customer 4 with $\theta = 8$ items of product 1 and 3 items of product 2. For customer 6, if the state is $(z_1,z_2)=(2,\ 2)$, then the optimal action for the vehicle is the action 2_θ with $\theta=7$. According to this action, the vehicle goes to the depot, restocks with $\theta=7$ items of product 1 and $Q-\theta = 12-7 = 5$ items of product 2 and then goes to customer 7. In Table 1, for customer 6, and for some states (z_1,z_2) for which the optimal action is the action $3_{(\theta_2,\theta)}$, the optimal values of θ_2 and θ are presented.

In Table 2 for customer 6 and for some states (z_1,z_2), for which the optimal action is the action $7_{(\theta_2,\theta)}$, the optimal values of θ_2 and θ are presented.

In Fig. 4, we present a graph that shows the variation in the minimum expected total cost f_0 as the probability p of the binomial distribution $Bin(Q,p)$ for the demand ξ_j of each customer j takes values in the set $\{0.1,\ 0.2\ldots,0.8,0.9\}$

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

In Fig. 5, we present graphs that show, as Q varies in the set $\{12,14,\ldots,60\}$ the variation in computation times, expressed in seconds, required by the initial dynamic programming and the special-purpose dynamic programming algorithm.

We observe that, as Q increases, the computation times for both algorithms increase non-linearly. For the special-purpose algorithm the form of the graph verifies that the complexity of the algorithm is $O(NQ^3)$. 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.

Example 2. Suppose that $N=9$ and $Q=7$. The travel costs between customers j and $j+1$, $j \in \{1,\ldots,8\}$, are given by: $c_{12}=9$, $c_{23}=8$, $c_{34}=9$, $c_{45}=7$, $c_{56}=8$, $c_{67}=10$, $c_{78}=9$ and $c_{89}=7$. The travel costs between customers $j, j \in \{1,\ldots,9\}$, and the depot are given by: $c_{10}=7$, $c_{20}=8$, $c_{30}=8$, $c_{40}=7$, $c_{50}=6$, $c_{60}=8$, $c_{70}=6$, $c_{80}=7$ and $c_{90}=7$. Note that these costs satisfy the triangle inequality. We assume that the penalty costs $\pi_1,\ldots,\pi_9$ incurred when a unit of the non-preferred product is delivered to customers 1, …,9 instead of the preferred one are elements of the row vector $\pi = (6,\ 5,\ 6,\ 6,\ 4,\ 5,\ 6,\ 7,\ 8)$. The probabilities $p_1,\ldots,p_9$ that customers 1, …,9 prefer product 1 are elements of the row vector $p = (0.6,\ 0.5,\ 0.3,\ 0.4,\ 0.6,\ 0.3,\ 0.7,\ 0.8,\ 0.6)$. We further assume that the demand ξ_j of each customer $j \in \{1,\ldots,9\}$ is a continuous random variable that follows the Gamma distribution right-truncated in the interval $[0,\ Q]$. The probability density functions $\varphi_j(x)$ are given by:

$$\varphi_j(x) = [F(Q)]^{-1}\frac{\lambda^\alpha x^{\alpha-1}}{\Gamma(\alpha)}e^{-\lambda x}, x \in [0,\ Q],$$

where, $\alpha,\lambda > 0$, $\Gamma(\alpha) = \int_0^\infty e^{-u}u^{\alpha-1}du$ and $F(x) = [\Gamma(\alpha)]^{-1}\int_0^{\lambda x} e^{-u}u^{\alpha-1}du$, $x \ge 0$. The Gamma distribution seems to be a reasonable choice for the demand of a customer for a product since, as mentioned in p. 442 in the book of Tijms (2003), in inventory applications the Gamma distribution is often used to model demand distributions. We choose $\alpha=4$ and $\lambda=2$. We also choose $\rho=0.05$ so that the discretized state space $\tilde{S}$ for each customer $j \in \{1,\ldots,9\}$ is the set $\{(k*0.05, l*0.05) : k=-1,\ldots,-140, l=0,\ldots,140\} \cup \{(k*0.05, l*0.05),$ $k,l=0,\ldots,140 \text{ s.t. } k+l \le 140\} \cup \{(k*0.05, l*0.05) : k=0,\ldots,140, l$ $=-1,\ldots,-140\}$

In Figs. 6 and 7, we present the optimal decisions for customers 6 and 8. If $z_1 \in [0,Q]$, $z_2 \in [0,Q]$ and $z_1+z_2 \le Q$, the action of proceeding directly to next customer (Action 1) is colored by blue and the action of going to the depot for restocking with product loads θ of product 1 and $Q-\theta$ of product 2 and then going to the next customer (Action 2_θ) is colored by red. If $z_1 \in [-Q,0)$, $z_2 \in [0,Q]$ or if $z_1 \in [0,Q]$, $z_2 \in [-Q,0)$ and if $z_2 < -z_1$ or if $z_1 < -z_2$ the action $3_{(\theta_1,\theta)}$ (or the action $3_{(\theta_2,\theta)}$) which

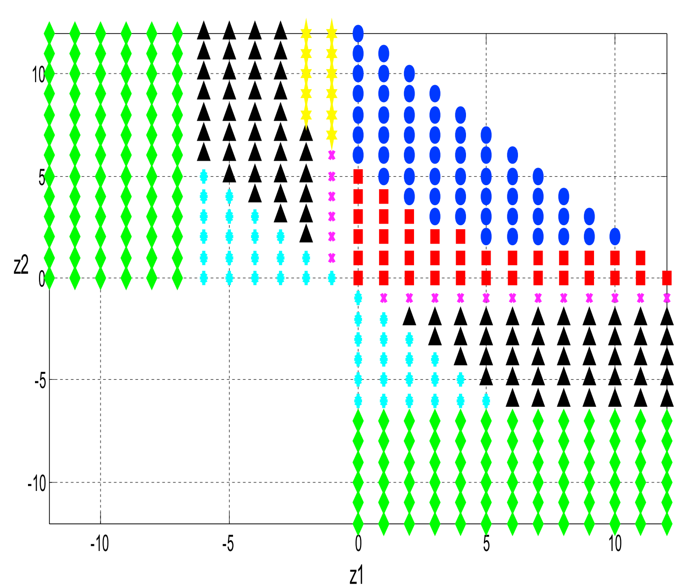

Fig. 2. The optimal decisions for customer 3.

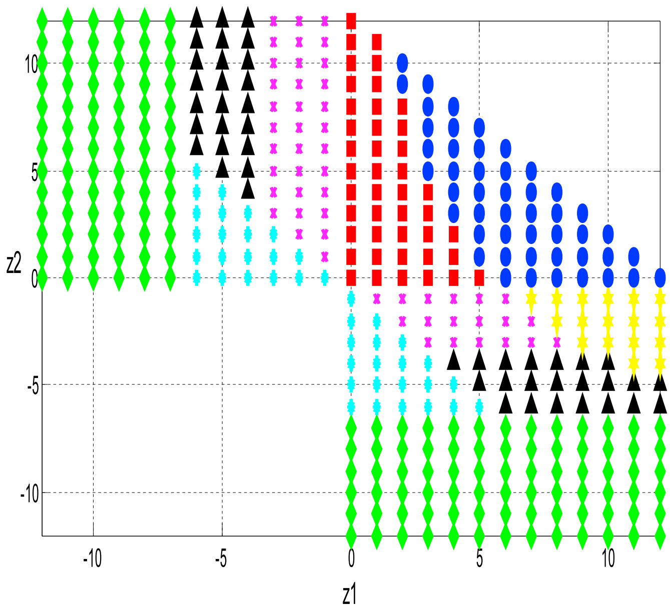

Fig. 3. The optimal decisions for customer 6.

correspond to the quantity $C_j(z_1, z_2)$ is colored by cyan and the action 4_θ which corresponds to the quantity D_j is colored by green. If $z_2 \geq -z_1$ or if $z_1 \geq -z_2$, the action 5 which corresponds to the quantity $E_j(z_1, z_2)$ is colored by yellow, the action 6_θ which correspond to the quantity $F_j(z_1)$ (or $F_j(z_2)$) is colored by magenta and the action $7_{(\theta_1,\theta)}$ (or the action $7_{(\theta_2,\theta)}$) which correspond to the quantity $G_j(z_1)$ (or $G_j(z_2)$) is colored by black.

The value of the minimum total expected cost f_0 is found to be approximately equal to 108.37. The computation time of the special-purpose dynamic programming algorithm is 22 143 s. It is considerably smaller than the computation time of the initial dynamic programming algorithm which is 23 670 seconds.

Both algorithms enable us to determine the optimal values of θ_1 (or θ_2) and θ when the optimal actions are the actions $3_{(\theta_1,\theta)}$ (or $3_{(\theta_2,\theta)}$)and $7_{(\theta_1,\theta)}$ (or $7_{(\theta_2,\theta)}$) and the optimal value of θ when the optimal actions are the actions 2_θ, 4_θ and 6_θ.

For example, for customer 6, if the state is $(z_1, z_2) = (-3.55,\ 2.45)$, then the optimal action for the vehicle is the action $3_{(\theta_1,\theta)}$. According to this action, the vehicle delivers quantity equal to $\theta_1 = 2.45$ of non-preferred product 2 to customer 6, goes to the depot to restock with quantity equal to $\theta - z_1 - \theta_1 = 3.45 - (-3.55) - 2.45 = 4.55$ of product 1 and quantity equal to $Q + z_1 + \theta_1 - \theta = 7 - 3.55 + 2.45 - 3.45 = 2.45$ of product 2, returns to customer 6 to deliver the owed quantity equal to $-z_1 - \theta_1 = -(-3.55) - 2.45 = 1.1$ of product 1 and then proceeds to customer 7 with quantity equal to $\theta = 3.45$ of product 1 and quantity equal to 2.45 of product 2. If, again for customer 6, the state is $(z_1, z_2) = (-2.9,\ 6.05)$, then the optimal action for the vehicle is the action $7_{(\theta_1,\theta)}$. According to this action, the vehicle delivers quantity equal to $\theta_1 = 2.85$ of non-preferred product 2 to customer 6, goes to the depot to restock with quantity equal to $\theta - z_1 - \theta_1 = 4.1 - (-2.9) - 2.85 = 4.15$ of product 1 and quantity equal to $Q + z_1 + \theta_1 - \theta = 7 - 2.9 + 2.85 - 4.1 = 2.85$ of product 2, returns to customer 6 to deliver the owed quantity equal to $-z_1 - \theta_1 = -(-2.9) - 2.85 = 0.05$ of product 1 and then proceeds to customer 7 with quantity equal to $\theta = 4.1$ of product 1 and quantity equal to 2.85 of product 2. For customer 6, if the state is $(z_1, z_2) = (1.15,\ 0.45)$, then the optimal action for the vehicle is the action 2_θ with $\theta = 4.4$. According to this action, the vehicle goes to the depot, restocks with quantity of product 1 equal to $\theta = 4.4$ and with quantity of product 2 equal to $Q - \theta = 2.6$ and then goes to customer 7. In Table 3, for customer 8 and for some states (z_1, z_2) for which the

Table 1
The optimal values of θ_2 and θ for customer 6 when action $3_{(\theta_2,\theta)}$ is optimal.

States $(z_1,\ z_2)$	Optimal value of θ_2	Optimal value of θ
$(1,\ -4)$	1	8
$(2,\ -3)$	2	9
$(3,\ -5)$	3	7
$(3,\ -4)$	3	8
$(4,\ -5)$	4	7

Table 2
The optimal values of θ_2 and θ for customer 6 when the action $7_{(\theta_2,\theta)}$ is optimal.

States $(z_1,\ z_2)$	Optimal value of θ_2	Optimal value of θ
$(7,\ -6)$	5	6
$(7,\ -5)$	4	7
$(8,\ -4)$	3	8
$(9,\ -6)$	5	6
$(10,\ -6)$	5	6

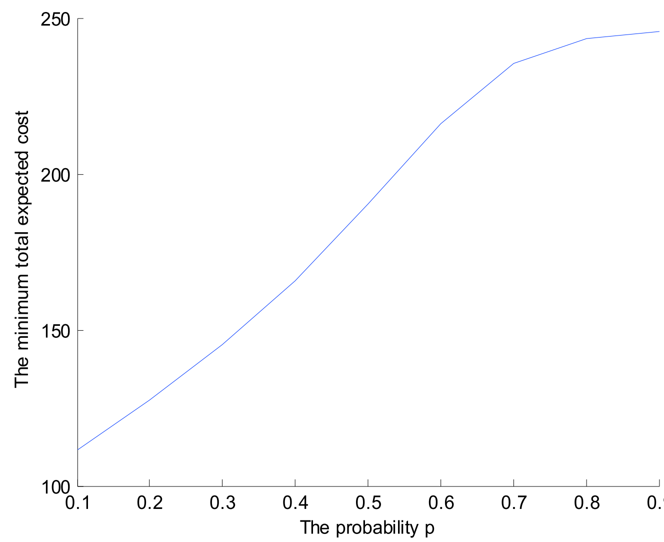

Fig. 4. The minimum expected total cost as p varies.

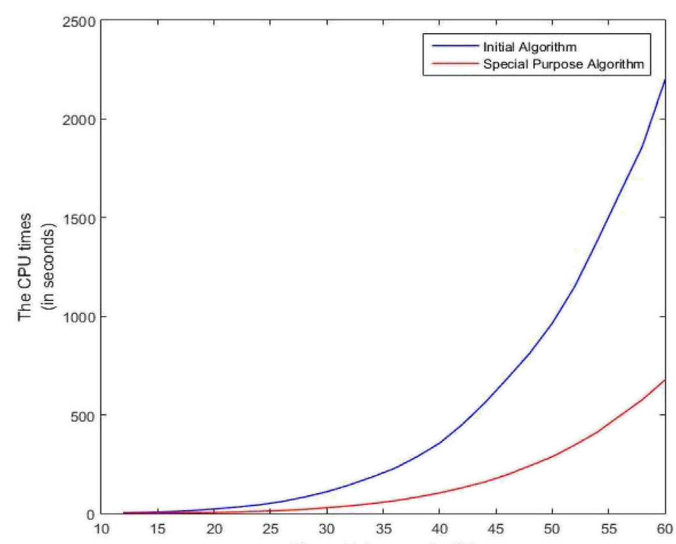

Fig. 5. The computation times of the algorithms as Q varies.

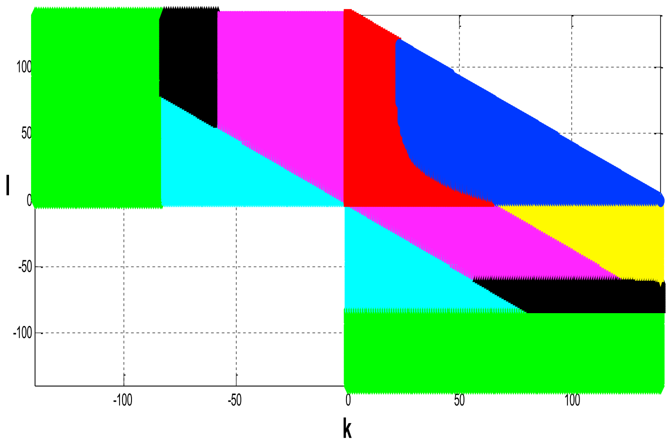

Fig. 6. The optimal decisions for customer 6.

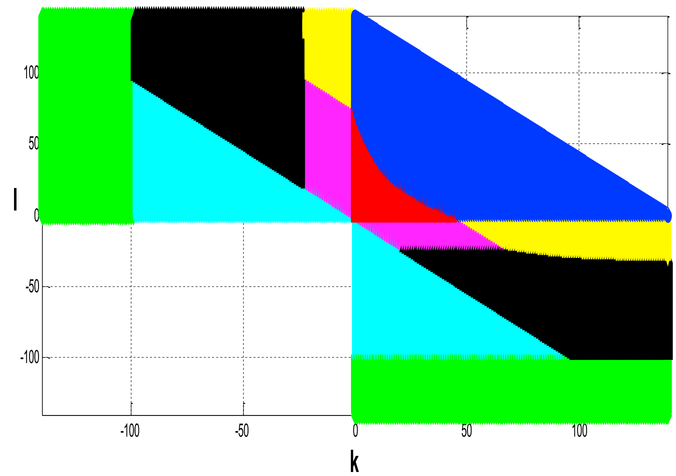

Fig. 7. The optimal decisions for customer 8.

optimal action is the action $3_{(\theta_1,\theta)}$, the optimal values of θ_1 and θ are presented.

In Table 4, again for customer 8 and for some states (z_1, z_2) for which the optimal action is the action $7_{(\theta_2,\theta)}$, the optimal values of θ_2 and θ are presented.

Suppose that $Q = 3$ and that the number of customers N takes values in the set $\{7, 8, \ldots, 15\}$ For each value of N, let $c_{i,i+1} = 27, i \in \{1, \ldots, N-1\}$, $c_{i0} = 24$, if i is odd and $c_{i0} = 22$, if i is even. For each customer $j \in \{1, \ldots, N\}$ we assume that the penalty cost π_j per unit of quantity of the non-preferred product that is delivered instead of the preferred one is equal to 3 and the probability of preference of product 1 is equal to 0.5.

In Fig. 8, we present graphs that show, as N varies in the set $\{7, 8, \ldots, 15\}$, 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 computation times for both algorithms increase approximately linearly. The form of the graph confirms that the complexity of the special-purpose algorithm $(O(N[Q/\rho]^3)$ is a linear function with respect to N. 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 problem when the customers are not ordered

We modify the problem that we introduced in Section 2 by assuming that the customers are not serviced according to a predefined sequence. In this case there are $N!$ different customer sequences that the vehicle may follow. For each sequence using the dynamic programming algorithm we can find the optimal routing 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 are discrete random variables, it is possible to find the optimal customer sequence for values of N up to 8. As illustration we give below a numerical example.

Example 3. Suppose that $Q = 6$. We assume that the number of customers N takes values in the set $\{3, 4, 5, 6, 7, 8\}$ The travel costs c_{ij} between customers $i, j \in \{1, \ldots, 8\}$ and the travel costs c_{i0} between each customer $i \in \{1, \ldots, 8\}$ and the depot are given by the following symmetric matrix $C = (c_{ij}),\ i, j = 0, \ldots, 8$.

Table 3
The optimal values of θ_1 and θ for customer 8 when the action $3_{(\theta_1,\theta)}$ is optimal.

States (z_1, z_2)	Optimal value of θ_1	Optimal value of θ
(−4.6, 4.55)	4.55	2.4
(−4.9, 4.6)	4.6	2.1
(−4.85, 4.75)	4.75	2.15
(−4.9, 4.85)	4.85	2.1
(−4.7, 4.6)	4.6	2.3

$$C = \begin{pmatrix} 0 & 18 & 21 & 15 & 14 & 22 & 17 & 13 & 13 \\ 18 & 0 & 12 & 10 & 9 & 9 & 10 & 11 & 12 \\ 21 & 12 & 0 & 12 & 11 & 12 & 10 & 11 & 10 \\ 15 & 10 & 12 & 0 & 10 & 11 & 12 & 13 & 12 \\ 14 & 9 & 11 & 10 & 0 & 12 & 15 & 17 & 10 \\ 22 & 9 & 12 & 11 & 12 & 0 & 13 & 12 & 10 \\ 17 & 10 & 10 & 12 & 15 & 13 & 0 & 11 & 13 \\ 13 & 11 & 11 & 13 & 17 & 12 & 11 & 0 & 12 \\ 13 & 12 & 10 & 12 & 10 & 10 & 13 & 12 & 0 \end{pmatrix}$$

These costs satisfy the triangle inequality. We assume that the penalty costs $\pi_1, \ldots, \pi_8$ incurred when an item of the non-preferred product is delivered to customers $1, \ldots, 8$ instead of the preferred one are elements of the row vector $\pi = (3,\ 4,\ 3,\ 10,\ 4,\ 10,\ 1,\ 2)$. We further assume that the demand ξ_j of each customer $j \in \{1, \ldots, 8\}$ is a random variable which follows the binomial distribution $Bin(Q,0.4)$ and that the probabilities $p_1, \ldots, p_8$ that customers 1,…,8 prefer product 1 are elements of the row vector $p = (0.4,\ 0.2,\ 0.3,\ 0.3,\ 0.4,\ 0.5,\ 0.6,\ 0.5)$. For $N \in \{3, \ldots, 8\}$ we consider the network consisting of customers 1,…,N. In Table 5 we present for $N \in \{3, \ldots, 8\}$ 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) 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 Fig. 9, we present the graphs that show, as N takes values in the set $\{3, \ldots, 8\}$, the variation in required computations times (expressed in seconds) if the initial dynamic programming algorithm and the special-purpose dynamic programming algorithm are used.

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.

6. Summary of results and a topic for future research

In this paper a capacitated stochastic single vehicle routing problem

Table 4
The optimal values of θ_2 and θ for customer 8 when the action $7_{(\theta_2,\theta)}$ is optimal.

States (z_1, z_2)	Optimal value of θ_2	Optimal value of θ
$(6.8, -4.25)$	4.2	2.75
$(6.85, -4.55)$	4.5	2.45
$(6.9, -1.5)$	1.45	5.5
$(6.95, -2.1)$	2.05	4.9
$(7, -4.05)$	4	2.95

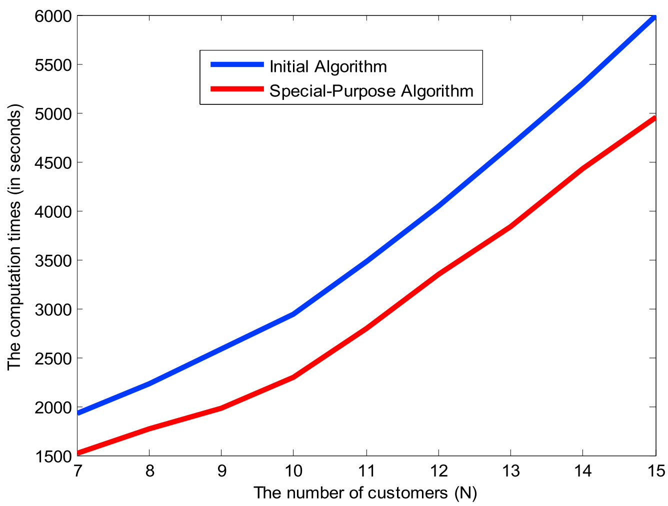

Fig. 8. The computation times of the algorithms as N varies.

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

N	$N!$	Minimum Cost	Optimal Sequence	Time 1	Time 2
3	6	80.50	2,1,3	0.56	0.23
4	24	100.64	4,1,2,3	2.83	0.84
5	120	127.53	1,5,3,2,4	18.44	5.05
6	720	152.48	6,2,3,5,1,4	139.15	38.23
7	5040	169.25	6,2,7,5,1,4,3	1173.90	324.83
8	40 320	187.93	6,2,8,5,3,4,1,7	10 825.91	617.20

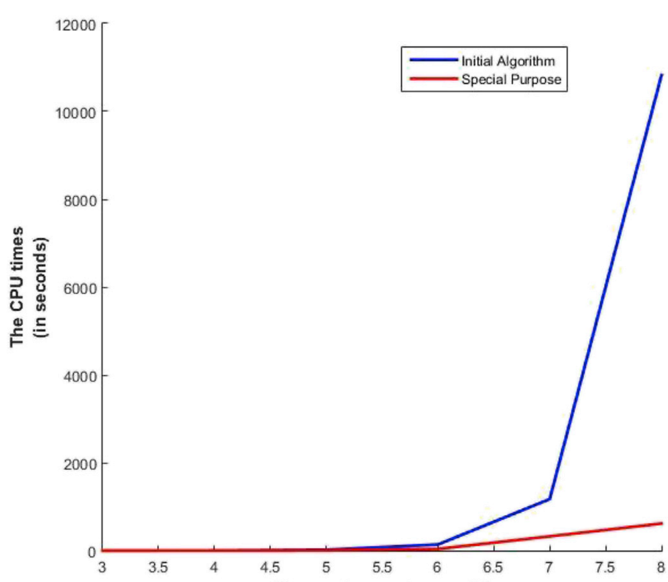

Fig. 9. The computation times of the algorithms as N varies.

was studied in which (i) the customers are served according to a predefined sequence, (ii) the vehicle delivers to the customers two similar but not identical products, (iii) the product preference and the demand of each customer are stochastic, (iv) the actual preference and the actual demand of each customer are revealed as soon as the vehicle arrives at the customer's site. The cost structure includes travel costs between consecutive customers, travel costs between customers and the depot and penalty costs that are incurred when a product that is not preferred by a customer is delivered to him/her. We selected as decision epochs for the routing of the vehicle, the epochs at which the vehicle visits for the first time each customer and the maximum possible quantity of the preferred product has been delivered. This selection of decision epochs makes possible a dynamic programming formulation for the determination of the routing strategy that minimizes the expected total cost for servicing all customers. The optimal routing strategy has a specific threshold-type structure. This result enables us to design a special-purpose dynamic programming algorithm that is considerably more efficient than the initial one. If the above Assumption (i) does not hold, it is possible to compute the optimal routing strategy for moderate values of the number of customers.

A possible topic for future research could be the study of a more general problem where the vehicle delivers $K \geq 2$ similar but not identical products.

Acknowledgments

We would like to thank an anonymous referee whose valuable suggestions improved the content and the presentation of the paper. Special thanks are also due to Petros Boutselis, Dimitris Pandelis, Dimitris Triantafyllou for useful discussions.

References

Belloso, J., Juan, A.A., Martinez, E., Faulin, J., 2017. A biased-randomized metaheuristic for the vehicle routing problem with clustered and mixed backhauls. Networks 69, 241–255.
Cortés, C.E., Gendreau, M., Rousseau, R.L., Souyris, S., Weintraub, A., 2014. Branch-and-price and constraint programming for solving a real-life technician dispatching problem. Eur. J. Oper. Res. 238, 300–312.
Dayarian, I., Crainic, T.G., Gendreau, M., Rei, W., 2015. A branch-and-price approach for a multi-period vehicle routing problem. Comput. Oper. Res. 55, 167–184.
Derigs, U., Reuter, K., 2009. A simple and efficient tabu search heuristic for solving the open vehicle routing problem. J. Oper. Res. Soc. 60, 1658–1669.
Derigs, U., Gottlieb, J., Kalkoff, J., Piesche, M., Rothlauf, F., Vogel, U., 2011. Vehicle routing with compartments: applications, modelling and heuristics. OR Spectrum 33, 885–914.
Dikas, G., Minis, I., Mamasis, K., 2016. Single vehicle routing with predefined client sequence and multiple warehouse returns: the case of two warehouses. Cent. Eur. J. Oper. Res. 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. Eur. J. Oper. Res. 244, 990–993.
Gauvin, C., Desaulniers, G., Gendreau, M., 2014. A branch-cut-and-price algorithm for the vehicle routing problem with stochastic demands. Comput. Oper. Res. 50, 141–153.
Gendreau, M., Laporte, G., Seguin, R., 1996. Stochastic vehicle routing. Eur. J. Oper. Res. 88, 3–12.
Goetschalckx, M., Jacobs-Blecha, C., 1989. The vehicle routing problem with backhauls. Eur. J. Oper. Res. 42, 39–51.
Golden, B.L., Yee, J.R., 1979. A framework for probabilistic vehicle routing. AIIE Trans. 11, 109–112.
Haugland, D., Ho, S.C., Laporte, G., 2007. Designing delivery districts for the vehicle routing problem with stochastic demands. Eur. J. Oper. Res. 180, 997–1010.
Kriticos, M.N., Ioannou, G., 2013. The heterogeneous fleet vehicle routing problem with overloads and time windows. Int. J. Prod. Econ. 144, 68–75.
Kyriakidis, E.G., Dimitrakos, T.D., 2008. Single vehicle routing problem with a predefined customer sequence and stochastic continuous demands. Math. Sci. 33, 148–152.
Kyriakidis, E.G., Dimitrakos, T.D., 2013. A vehicle routing problem with a predefined customer sequence, stochastic demands and penalties for unsatisfied demands. In: Proceedings of 5th International Conference on Applied Operational Research. Lecture Notes in Management Science, vol. 5, pp. 10–17.
Laporte, G., Louveaux, F.V., Van Hamme, L., 2002. An integer L-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. Oper. Res. 50, 415–423.
Minis, I., Tatarakis, A., 2011. Stochastic single vehicle routing problem with delivery and pickup and a predefined customer sequence. Eur. J. Oper. Res. 213, 37–51.

Nguyen, V.A., Jiang, J., Ng, K.M., Teo, K.M., 2016. Satisfying measure approach for vehicle routing problem with time windows under uncertainty. Eur. J. Oper. Res. 248, 404–414.
Olivera, A., Viera, O., 2007. Adaptive memory programming for the vehicle routing problem with multiple trips. Comput. Oper. Res. 34, 28–47.
Pandelis, D.G., Kyriakidis, E.G., Dimitrakos, T.D., 2012. Single vehicle routing problems with a predefined customer sequence, compartmentalized load and stochastic demands. Eur. J. Oper. Res. 217, 324–332.
Pandelis, D.G., Karamatsoukis, C.C., Kyriakidis, E.G., 2013a. Single vehicle routing problems with a predefined customer order, unified load and stochastic discrete demands. Probab. Eng. Inf. Sci. 27 (1), 1–23.
Pandelis, D.G., Karamatsoukis, C.C., Kyriakidis, E.G., 2013b. Finite and infinite-horizon single vehicle routing problems with a predefined customer sequence and pickup and delivery. Eur. J. Oper. Res. 231, 577–586.
Pillac, V., Gendreau, M., Gueret, C., Megaglia, A., 2013. A review of dynamic vehicle routing problems. Eur. J. Oper. Res. 225, 1–11.
Psaraftis, H.N., Wen, M., Kontovas, C.A., 2016. Dynamic vehicle routing problems: three decades and counting. Networks 67, 3–31.
Segerstedt, A., 2014. A simple heuristic for vehicle routing-A variant of Clarke and Wright's saving method. Int. J. Prod. Econ. 157, 74–79.
Sipser, M., 2013. Introduction to the Theory of Computation, third ed. Cengage Learning, Boston.
Syrichas, A., Crispin, A., 2017. Large-scale vehicle routing problems: quantum Annealing, tunings and results. Comput. Oper. Res. 87, 52–62.
Tasan, A.S., Gen, M.A., 2012. A genetic algorithm based approach to vehicle routing problem with simultaneous pick-up and deliveries. Comput. Ind. Eng. 62, 755–761.
Tatarakis, A., Minis, I., 2009. Stochastic single vehicle routing with a predefined customer sequence and multiple depot returns. Eur. J. Oper. Res. 197, 557–571.
Tijms, H.C., 2003. A First Course in Stochastic Models. Wiley, Chichester.
Toth, P., Vigo, D. (Eds.), 2014. The Vehicle Routing Problem. Problems, Methods, and Applications, second ed. MOS-SIAM, Philadelphia, PA.
Tsirimpas, P., Tatarakis, A., Minis, I., Kyriakidis, E.G., 2008. Single vehicle routing with a predefined customer sequence and multiple depot returns. Eur. J. Oper. Res. 187, 483–495.
Yang, W.-H., Mathur, K., Ballou, R.H., 2000. Stochastic vehicle routing problem with restocking. Transport. Sci. 34, 99–112.
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. Eur. J. Oper. Res. 249, 144–154.
Zhang, T., Chaovalitwongse, W.A., Zhang, Y., 2012. Scatter search for the stochastic travel-time vehicle routing problem with simultaneous pick-ups and deliveries. Comput. Oper. Res. 39 (10), 2277–2290.