Received: 1 September 2019 | Accepted: 3 September 2019
DOI: 10.1002/net.21908

SPECIAL ISSUE ARTICLE

WILEY

Complexity of inventory routing problems when routing is easy

Annelieke C. Baller[1] | **Martijn van Ee**[2] | **Maaike Hoogeboom**[1] | **Leen Stougie**[3]

[1]Department of Supply Chain Analytics, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands
[2]Faculty of Military Sciences, Netherlands Defence Academy, Den Helder, The Netherlands
[3]Centrum voor Wiskunde en Informatica (CWI), INRIA-Erable, Department of Econometrics and Operations Research, Vrije Universiteit, Amsterdam, The Netherlands

Correspondence
Annelieke C. Baller, Department of Supply Chain Analytics, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands.
Email: a.c.baller@vu.nl

Funding information
This research was supported by the Nederlandse Organisatie voor Wetenschappelijk Onderzoek, 024.002.003, 407-13-050, 612.001.215.

Abstract

In the inventory routing problem (IRP) inventory management and route optimization are combined. The traveling salesman problem (TSP) is a special case of the IRP, hence the IRP is NP-hard. We investigate how other aspects than routing influence the complexity of a variant of the IRP. We first study problem variants on a point and on the half-line. The problems differ in the number of vehicles, the number of days in the planning horizon and the service times of the customers. Our main result is a polynomial time dynamic programming algorithm for the variant on the half-line with uniform service times and a planning horizon of 2 days. Second, for nearly any problem in the class with nonfixed planning horizon, we show that the complexity is dictated by the complexity of the pinwheel scheduling problem, of which the complexity is a long-standing open research question. Third, NP-hardness is shown for problem variants with nonuniform servicing times. Finally, we prove strong NP-hardness of a Euclidean variant with uniform service times and an easily computable routing cost approximation, avoiding immediate NP-hardness via the TSP.

KEYWORDS

approximation, computational complexity, dynamic programming, inventory routing, periodic replenishment, pinwheel scheduling

1 | INTRODUCTION

This article studies the computational complexity of special cases of a variant of the inventory routing problem (IRP), in which a set of customers is supplied over a given time horizon by identical vehicles from a central depot. Each customer has a storage capacity, a fixed demand per day, a latest delivery day at the start of the planning horizon and a service time. The metrics that underlie the customer locations do not immediately imply intractability because of routing aspects. In particular, we consider the problem in which all customers are located in a single point, on a half-line and in the Euclidean plane, but the latter under a specific approximation of the tour length. On a half-line, the depot is located in the origin, that is, at one end of the half-line. The vehicles have a tour duration constraint which limits the number of time units per day (traveling plus service time). The objective is to minimize the total time spent by all vehicles over all days.

The motivation for this study stems from a business project in ATM replenishment in the Netherlands. ATMs need to be replenished regularly such that banknotes are sufficiently available to consumers. In practice, this is an involved problem in which service levels, safety regulations and uncertainty play a role. In this article, we study a stylized version of this problem.

In many vehicle routing problems (VRP) the vehicles are capacitated in terms of load, however, in the ATM replenishment problem, the time spent by a vehicle is often more binding [8]. Time is also more binding than load in, for example, online ordered package delivery and blood product distribution [23]. Therefore, we consider a tour duration constraint.

Almost any version of the IRP is NP-hard since it contains the well-known NP-hard traveling salesman problem (TSP) [26, 28] as a special case. We investigate the complexity of an IRP on metrics for which routing does not cause immediate

NP-hardness through TSP. In the first part of the article, we consider the problem on a point, that is, the depot and the customers are all at the same location, and on the half-line. Furthermore, the problem variants studied differ in the number of vehicles available, the length of the planning horizon and the type of service times. In the second part of the article, we consider the respective problem in the Euclidean plane and choose a route length approximation function that avoids hardness through TSP. Still the problem is shown to be NP-hard.

Some variants of the studied IRP are easily shown to be solvable in polynomial time and some are easily shown to be NP-hard (Section 3). In order to identify which aspects determine the computational complexity of the respective IRP variant, we search for borderline problem variants on a point and on the half-line, which are either maximally easy or minimally hard. A maximally easy problem variant is a variant that is polynomial time solvable, but if one feature is generalized it becomes hard or has an open complexity. Similarly, a minimally hard problem becomes easy or open if one feature is further restricted.

IRPs form a class of widely studied and still challenging problems in the Operations Research literature. An introduction to the IRP is given in the tutorials Bertazzi and Speranza [11, 12]. In their introduction to IRPs, Bertazzi et al. [10] provide examples that give insight in the influence of holding costs, inventory capacities at the customers, and continuous consumption of goods at the customers. The authors state informally that limited storage capacity causes extra complexity in IRPs because of the implied time required between two deliveries. We will formalize this statement in this article.

Several literature surveys have been published since the 1990s [2, 7, 15, 16, 20, 27, 29]. The two most recent surveys each have a different focus. Andersson et al. [2] focus on industrial aspects of inventory routing and they propose a classification based on seven aspects that concern time, demand (deterministic or stochastic), routing, inventory, and fleet aspects. The survey is split into three parts based on the time horizon: instant, finite, and infinite time horizon. Coelho et al. [15] state in their recent survey that there “does not really exist a standard version of the problem,” since many variants with changing aspects are present in the literature. They propose a classification based on seven criteria including time horizon, routing, inventory and fleet aspects, and call all variants of the IRP that fit these criteria as “basic versions.” These seven criteria do not include demand aspects, since the authors want to separate the structure of the problem from information availability (i.e., stochasticity of demand). The criterion added compared to Andersson et al. [2] is the inventory policy used in the problem (maximum level or order-up-to level). Coelho et al. [15] give an in-depth overview of models and solution methods for both “basic versions” of the IRP as well as extensions of this version. The solution methods are divided in exact methods and heuristic methods for the basic versions of the problem. Additionally, Desaulniers et al. [19] propose a structurally different problem formulation and an exact solution method for the IRP that gives promising results for the multiple-vehicle IRP. Alvarez et al. [1] and Archetti et al. [4] present the most recent heuristic solution methods for the IRP.

The variant of the IRP considered in this article fits into the classification of Coelho et al. [15] as follows. It has a finite time horizon and a one-to-many structure, which means that one vehicle can visit multiple customers in one route. Multiple homogeneous vehicles are available and each vehicle can serve multiple customers in one route. Inventory is replenished via an order-up-to-level policy, which means that inventory is filled up to capacity at each replenishment and all demand has to be satisfied, that is, back-orders or lost sales are not allowed. We assume that all demand information is available at the beginning of the planning horizon. Furthermore, suppose that in a given day all replenishments take place before demand occurs, which is a common assumption in IRP (see, e.g., Archetti et al. [3]). Besides this classification, we take the service times of the customers as given and we do not consider inventory holding costs. Hence, our objective is to minimize the total traveling and service time. Moreover, instead of a vehicle capacity constraint in terms of units of goods, we consider a tour duration constraint which limits the time spent by a vehicle per day. In Section 4.4, we briefly discuss that our results with the tour duration constraint imply similar results for the case with a vehicle capacity constraint.

In this article, we study problem variants of the IRP in which customers are located on a point or on the half-line. A similar study was executed for the VRP by Archetti et al. [5]. Specifically, the authors consider the VRP with unsplittable demand and a limited fleet on a line, a star, a tree, and a cycle. They show several hardness results using the relation with the weakly NP-hard partition problem. Here we derive similar hardness results by relating our problem variants to the strongly NP-hard bin packing problem (BPP).

Finally, we mention two papers that consider problems similar to the ones in this article. Das et al. [17] study the train delivery problem with a single time period and multiple capacitated vehicles. The customers have weights and are located on the half-line. The goal is to assign customers to vehicles such that the vehicle capacity is not violated and the total distance traveled is minimized. The authors mention the NP-hardness of this problem, since it generalizes the BPP (cf., our Section 3.2.1). The main focus is on investigating approximation algorithms. Bosman et al. [13] consider a replenishment problem on a tree and on general metrics over an arbitrary time horizon. Their main results also concern approximation algorithms, but some of their complexity results are similar to ours (cf., our Section 3.2.2).

The remainder of the article is organized as follows. In Section 2 the studied IRP variant is formally described, and the relevant problem variants are presented. We also present related problems and their complexity which will play a role in the complexity analysis of our problems. In Section 3, we present complexity results of some problem variants that are easily seen

TABLE 1 Overview complexity results, where problems are either in P (dots), NP-hard (diagonal lines), strongly NP-hard (vertical lines), or PSP-hard (horizontal lines) [Color table can be viewed at wileyonlinelibrary.com]

		Point, $\alpha_2 = point$			Half-line, $\alpha_2 = half\text{-}line$		
		$\alpha_4 = 1$	$\alpha_4 = 2$	$\alpha_4 = Z$	$\alpha_4 = 1$	$\alpha_4 = 2$	$\alpha_4 = Z$
Unif. service times, $\alpha_3 = s$	$\alpha_1 = 1$			3.2.2			
	$\alpha_1 = M$		3.1.1		3.1.2	4	
Arb. service times, $\alpha_3 = s_i$	$\alpha_1 = 1$		3.2.1	3.2.1	3.2.1		
	$\alpha_1 = M$	3.2.1					

polynomially solvable (in P), or intractable. The main result on a maximally easy borderline problem, for which the complexity is not trivial, is presented in Section 4. We show that the problem can be solved in polynomial time using dynamic programming. In Section 5, we present the hardness result for a variant of the problem in the Euclidean plane. Related problems and literature for that specific variant will be discussed in Section 5. Finally, Section 6 finishes the article with concluding remarks.

2 | PROBLEM DESCRIPTION AND RELATED PROBLEMS

In the problems we study, we are given a metric space containing N customers and a depot r. Each customer has a service or replenishment time s_i, a latest delivery day at the start of the planning horizon and a period p_i, $i = 1, \ldots, N$, which is the maximum number of days between two replenishments. The periods are defined directly by the customer's storage capacity and the fixed daily demand. At the depot, M identical vehicles are present that can each spend at most L time units per day on traveling plus service time. The vehicles need to return to the depot at the end of a day. We assume that travel time is equal to travel distance, that is, vehicles travel at unit speed. The length of the planning horizon is Z days. Customers can be replenished at most once per day, that is, split deliveries are not allowed. A solution to this problem consists of an assignment of customers to vehicles, and a route for each vehicle, for each day in the planning horizon. A solution is feasible if each vehicle spends at most L time units per day and the time between two consecutive replenishments of customer i is at most p_i days. The objective is to minimize the total time spent by the vehicles.

2.1 | Problem variants

We provide a concise description of the problem variants we study in this article, in much the same spirit as done for scheduling problems in Graham et al. [22] and later for dial-a-ride problems in de Paepe et al. [18]. For all variants, a vehicle can spend at most L time units a day. Remaining features of the problems are stated in a 4-field notation $\alpha_1|\alpha_2|\alpha_3|\alpha_4$. In this notation, α_1 denotes the number of identical vehicles which is equal to 1 or a given $M > 1$, hence $\alpha_1 \in \{1, M\}$. The problem is studied on a point and on the half-line, which is indicated by α_2, $\alpha_2 \in \{half\text{-}line, point\}$. We use $\alpha_2 = point$ to indicate that all distances are zero, that is, both the depot and the customers are located in one point. Equivalently, we could say that the vehicles drive at infinite speed. We use $\alpha_2 = half\text{-}line$ to indicate that the customers are located on a half-line. On the half-line, we assume the customers are numbered in increasing distance to the depot; customer i is located at distance d_i from the depot r and $d_1 \leq \cdots \leq d_N$. The type of service times are denoted by α_3, $\alpha_3 \in \{s, s_i\}$, in which s indicates uniform service times and s_i indicates that service times can differ per customer (arbitrary service times). The planning horizon is denoted by α_4, which can be equal to 1, 2 or a given $Z > 2$ days, $\alpha_4 \in \{1, 2, Z\}$.

Table 1 provides an overview of the complexity results obtained in Sections 3 and 4. Each entry of the table corresponds to a configuration of type of service times, whether the customers are located on a point or on the half-line, the number of vehicles and the length of the time horizon. All these problems are characterized as polynomially solvable, NP-hard, strongly NP-hard or "PSP-hard" which is defined in the next section. Table 1 contains references to the corresponding sections for all borderline problems and, additionally, for the problems in Section 3.1.

Most of the problems in Sections 3 and 4 consider a planning horizon of 2 days. For these problems, we define three types of customers: day 1-customers (D1-customers), day 2-customers (D2-customers) and period 1-customers (P1-customers). D1-customers and D2-customers need service latest on day 1 and 2, respectively. Note that D2-customers can also be served on day 1. P1-customers need service on both days 1 and 2. Similarly, define Dh-customers which need service latest on day h and Pm-customers need service every m days.

2.2 | Related problems

To assess the complexity of some variants in the set of problems, we use hardness results from the BPP and the pinwheel scheduling problem (PSP).

2.2.1 | Bin packing problem

Given are n items with weights $w_1, \ldots, w_n$ and bins with capacity B. Pack the items in a minimum number of bins such that each bin contains total item weight no more than B. BPP is strongly NP-hard [21]. The decision problem whether all items can be packed in two bins, with bins of capacity $\frac{1}{2}\sum_{j=1}^{n} w_j$, is known as the partition problem, and is a weakly NP-complete problem [26]. We use these hardness results in Section 3.2.1.

2.2.2 | Pinwheel scheduling problem

Given are n tasks with integer periods $p_1, \ldots, p_n$. Each time unit, one task can be scheduled. A schedule is feasible if the time between two consecutive moments at which task i is scheduled is at most p_i time units. The goal is to find a feasible schedule. A special feature of this problem is the following. If one would decide to schedule a task one time unit earlier, the next due date for the task is also shifted one time unit back. Hence, a decision for a given time unit influences directly the situation at a later moment in the schedule and can cause conflicts there.

The complexity of the PSP is a long-standing open question, which is mainly due to the compact input description. It was only shown to be in PSPACE by Holte et al. [24]. Recently, it was shown by Jacobs and Longo [25] that the PSP cannot be solved in pseudopolynomial time, unless there is a randomized algorithm for solving the well-known satisfiability problem in time $n^{\mathcal{O}(\log n \log\log n)}$. Since the latter is unlikely, the PSP is assumed to be intractable. Yet it is unclear if it is in NP or in co-NP. We define the class of PSP-hard problems as the problems that are at least as hard as the PSP. We use this hardness notion in Section 3.2.2.

3 | PRELIMINARY RESULTS

In this section, we consider some problems for which its computational complexity is easily established. We first discuss two easy problems that are actually not borderline easy, but we think they will provide insight into the structure of optimal solutions of the problem $M \mid half\text{-}line \mid s \mid 2$, which is discussed in Section 4. We finish this section with presenting two classes of intractable problems.

3.1 | Easy problems

The two easy problems are equivalent to special cases of problem $M \mid half\text{-}line \mid s \mid 2$. The first easy problem is the problem on a point, instead of on a half-line. In the second easy problem, the planning horizon is restricted to 1 day, instead of 2 days.

3.1.1 | Problem $M|point|s|2$

There are M vehicles, the depot and all the customers are located in one point, the customers have uniform service times ($s_i = s\ \forall\ i$) and there is a planning horizon of 2 days. Let #P1, #D1, and #D2 be the number of P1-customers, D1-customers, and D2-customers, respectively. Since there is no travel time and all customers have equal service times, it is most efficient to serve each customer (with $p_i > 1$) exactly once in the 2-day planning horizon. If the problem is feasible, this gives the optimal objective value. To check feasibility, we just need to check two things. First, whether M vehicles suffice to serve all P1-customers and D1-customers on day 1: $\#\mathrm{P1} + \#\mathrm{D1} \leq M\lfloor L/s \rfloor$. And, if so, second, whether M vehicles suffice to satisfy all the customers' service requirements in the 2 days, taking into account that a D2-customer can be served on day 1: $2\#\mathrm{P1} + \#\mathrm{D1} + \#\mathrm{D2} \leq 2M\lfloor L/s \rfloor$. Thus, the running time of the algorithm is linear in N.

3.1.2 | Problem $M \mid half\text{-}line \mid s \mid 1$

In this problem variant on the half-line, there are M vehicles, uniform service times ($s_i = s\ \forall\ i$) and a planning horizon of 1 day. Recall that there are N customers which are numbered in order of increasing distance to the depot on the half-line. Define a "region" to be the interval in which a given set of customers is located on the half-line. Polynomial solvability follows from the following lemma, which is easily proved by a simple exchange argument, left to the reader.

Lemma 1. *In an optimal solution of the given problem, the regions of customer locations served by any two vehicles are disjoint*.

The optimal solution is established as follows. Let a vehicle serve the farthest customer N and include, on the way to the depot, as many customers as possible with the highest indices. Let the next vehicle serve the farthest customer that is not served by the first vehicle and again, let this vehicle serve as many customers as possible. Continue until all M vehicles are used or all customers are served. This results in a solution in which each vehicle serves consecutive customers on the half-line. In case all vehicles are used and there are some customers left, there is no feasible solution. Otherwise, a feasible solution is found, which is clearly optimal by construction. Hence, if the order of the customers is given, this is a linear time algorithm.

3.2 | Hard problems

The problem variants studied in this section are the problems having either arbitrary processing times s_i or a planning horizon of Z days. Both problem variants are minimally hard, that is, restricting one of the characteristics makes a problem easy to solve.

3.2.1 | Problem $M|point|s_i|1$ and problem $1|point|s_i|Z$

Consider the class of problems with customers having arbitrary service times s_i. For the problem on a point, any feasible solution has the same objective value, hence feasibility is the core problem. The problem variant that has 1 day and one vehicle ($1|point|s_i|1$) is trivially in P (which also holds for $1\,|\,half\text{-}line\,|\,s_i\,|\,1$). However, if there are multiple vehicles and/or multiple days in the planning horizon, the problems are equivalent to BPPs (cf., Das et al. [17]). For example, in the problem $M|point|s_i|1$, the vehicles are equivalent to the bins in the BPP with bin capacity equal to duration limit L. Then the problem boils down to the feasibility question whether or not the number of available bins is sufficient to be able to assign each customer to a bin. Hence, this problem and more general variants are strongly NP-hard. The problem with one vehicle and a planning horizon of Z days ($1|point|s_i|Z$) is by similar arguments also strongly NP-hard. The problem with one vehicle and a planning horizon of 2 days ($1|point|s_i|2$) is a weakly NP-hard problem because of its equivalence to the partition problem. Concluding, any problem variant with arbitrary service times s_i and multiple vehicles and/or days is an NP-hard problem. In Section 4, but also in the next subsection, hardness is avoided through bin packing by restricting to uniform service times.

3.2.2 | Problem $1|point|s|Z$

Consider the problem on a point with uniform service times ($s_i = s\,\forall\, i$), one vehicle, and an arbitrary long time horizon Z. A special case of this problem, in which a vehicle can replenish at most one customer per day, that is, $L = s$, is equivalent to the PSP. Hence, this problem variant is PSP-hard (cf., Bosman et al. [13]). This does not imply that this variant is NP-hard, but that it is unlikely that it can be solved in polynomial time. Note that the recurrence of replenishments can occur because of the longer planning horizon. We avoid analyzing PSP-hard problems in Section 4 by restricting ourselves to a planning horizon of 2 days.

4 | POLYNOMIAL TIME ALGORITHM FOR $M\,|\,half\text{-}line\,|\,s\,|\,2$

In this section, we prove that problem variant $M\,|\,half\text{-}line\,|\,s\,|\,2$ is solvable in polynomial time. In this variant on the half-line there are M vehicles, uniform service times ($s_i = s\,\forall\, i$) and a planning horizon of 2 days. All previously mentioned hardness results are avoided as follows. By restricting the problem to the half-line, we avoid hardness through TSP; by restricting to uniform service times, we avoid hardness through bin packing; by restricting to a planning horizon of 2 days, we avoid hardness through PSP.

Recall that P1-customers have to be served on both days, D1-customers have to be served on day 1, whereas D2-customers can be served on either of the 2 days. To minimize total travel and service time, it must be determined how many vehicles to use every day and which customers to serve with each vehicle, such that the tour duration limit (of L time units) is not exceeded. Obviously, in any optimal solution each customer that does not need service every day is served only once. Hence total service time is always equal in any relevant solution and we disregard it from the objective from here onwards.

A dynamic programming algorithm (DP) is designed to solve this problem to optimality. To simplify explanations, in Section 4.1 we first assume that there are no customers that need service on both days (no P1-customers). In Section 4.2 this assumption will be relaxed. In Section 4.3 the running time of the DP will be analyzed, an observation to speed up the DP is made and the complexity of extensions of the problem is discussed. Finally, Section 4.4 discusses the variant of the respective IRP variant with a vehicle capacity constraint.

4.1 | Dynamic programming

Before discussing the DP in detail, we observe the following. Given the farthest customer that still needs to be assigned to a vehicle, there are only a limited number of options for the other customers that will be served by the same vehicle in an optimal

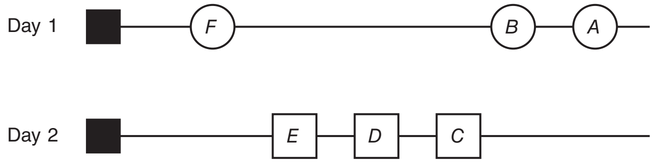

FIGURE 1 Example for observation, the depot is represented by black squares, D1-customers are represented by circles, and D2-customers by squares

solution. Consider the example in Figure 1. The figure shows two half-lines, the depot as black squares and the locations of the D1-customers and D2-customers. The D1-customers are indicated by circles and the D2-customers are indicated by squares.

Suppose customer A is the farthest customer that has not been assigned to a vehicle yet. Suppose that, because of the limitation on the time, at most three more customers can be served by the vehicle if customer A is the farthest served customer. For example, customers $\{A, B, C, D\}$ can be served by one vehicle. Now, suppose customer A is assigned to a vehicle v that will serve customers on day 1. Then, by Lemma 1, it is not optimal to have vehicle v serving customer F but not serving customer B. A similar argument holds for serving D2-customers with vehicle v. If customer D is served by vehicle v, but C is not served by vehicle v, this means that another vehicle w has to drive up to customer C. By interchanging customers C and D a better solution is constructed.

Hence, in general, in an optimal solution, any vehicle serves a combination of consecutive D1-customers and consecutive D2-customers, that is, no vehicle skips a Dh-customer to serve another Dh-customer closer to the depot. This observation is used in the DP. The idea of the DP is to start at the customer farthest from the depot and work backwards to the depot. For every customer that needs service latest on day 2, a decision must be made whether this customer is served on day 1 or 2.

As a basis for the DP, Lemma 2 first generalizes Lemma 1 to prove the observation in the example above. Then, the DP is formulated.

Lemma 2. *Starting from the farthest customer and moving toward the depot, no vehicle skips a Dh-customer to serve another Dh-customer closer to the depot, for* $h \in \{1, 2\}$.

Proof. First, consider the case of assigning customers that need service latest on day h to a vehicle v serving these customers on day h (case 1).

Case 1: the Dh-customers assigned to vehicle v will be consecutive on the half-line, otherwise the solution can be improved with the same interchange argument used in Lemma 1.

Second, consider the case of assigning customers that need service latest on any of the 2 days to a vehicle v on one of these days (such that the services are feasible). Define $S(v)$ to be the set of customers served by vehicle v. Again, distinguish two cases: the customer in $S(v)$ farthest from the depot is a D1-customer (case 2a) and a D2-customer (case 2b), respectively.

Case 2a: by case 1 it has been shown that D1-customers served by vehicle v are consecutive on the half-line. It remains to show that the D2-customers served by vehicle v are also consecutive on the half-line. Let A be the farthest D1-customer, B the farthest D2-customer and C the second farthest D2-customer from the depot such that $d_C < d_B$. Suppose there is an optimal assignment in which customer B is not assigned to vehicle v on day 1, but customers A and C are: ($A,C \in S(v)$ and $B \notin S(v)$). Then, for customer B it has to be decided on which day it is served and by which vehicle. If customer B is served on day 1 with vehicle w, this gives overlap between the delivery regions of vehicles v and w on day 1, which cannot be optimal by Lemma 1. Hence, customer B must be served on day 2 by vehicle u ($A,C \in S(v)$ and $B \in S(u)$) and the cost of vehicles v and u is $d_A + d_B$. By interchanging customers B and C, the solution is still feasible, but the cost is $d_A + d_C$ which is less than $d_A + d_B$ since customer C is closer to the depot than customer B which contradicts the assumption of optimality.

Case 2b: by case 1 it is known that D2-customers served by vehicle v on day 2 are consecutive on the half-line. Hence, if no D1-customers are served by vehicle v the lemma has been shown. If any D1-customer is served by v, the same arguments used in case 2a prove the statement also for this case, and therefore complete the proof. ■

It is easy to see that this lemma can be extended to $h \geq 3$. We will use Lemma 2 to find an optimal solution for problem $M \mid half\text{-}line \mid s \mid 2$ in polynomial time. Again, assume that the customers on the half-line are numbered 1,2, …, N in increasing distance from the depot.

In the DP, in every state the customer farthest from the depot that has not been assigned to a vehicle yet is considered. The crucial observation is that given this farthest unassigned customer, n, and its latest service day h, the number of customers with latest service day 1 that is served by the same vehicle v as customer n defines the next state. Define $C(\ell)$ as the number of customers that can be served by a vehicle within its time limit L given that customer ℓ is the farthest customer served by the vehicle. Define k as the number of D1-customers that are served by a vehicle (including the farthest customer if it is also a D1-customer). The value of k ranges from 0 (if n is not a D1-customer) to $C(n)$. Given a value of k, the optimal route for vehicle v is easy to find: select the customers farthest from the depot including exactly k D1-customers. If $k \geq 1$ vehicle v has to ride on day 1. Note that servicing only D2-customers on day 1 is also an option. Therefore, if $k = 0$, it is yet to be decided in the DP on which of the 2 days vehicle v will ride, given that there are still vehicles left for both days. Concluding, the value of k, the availability of vehicles, and the decision on the delivery day, determine both the set of customers served by the same vehicle as customer n and the day on which these customers are served. Hence, to find the optimal solution it is sufficient to consider for each customer n all possible values of k and the decision whether to serve a set of only D2-customers on day 1 or 2.

Let the indices of the D1-customers, in increasing order, be denoted by $1_1, 2_1, \ldots, I_1$, with I the total number of D1-customers. Similarly, D2-customers have indices $1_2, 2_2, \ldots, J_2$, with J the total number of D2-customers. Hence, customer 1 (which is closest to the depot) is either 1_1 or 1_2 and similarly, $N = I_1$ or $N = J_2$. Let n be the index of the customer farthest from the depot that still needs assignment to a vehicle, and let i_1 and j_2 be the indices of the D1-customer and D2-customer farthest from the depot that still need assignment. Hence, $n = i_1$ or $n = j_2$. Let M be the total number of vehicles available per day and let m_1 and m_2 be the number of remaining available vehicles for day 1 and 2, respectively. Define the value 0 for the indices i_1 and j_2 for the case that no D1-customer and D2-customer, respectively, exists or none is still to be assigned and let $x^+ = \max\{x, 0\}$.

A state of the DP is denoted by $<i_1, j_2, m_1, m_2>$. Let $f(i_1, j_2, m_1, m_2)$ be the minimal cost (time) of serving all customers in this state. The total minimal costs of an instance is given by $f(I_1, J_2, M, M)$. We present two recursion formulas: for $i_1 > j_2$, in which a D1-customer is the farthest unassigned customer:

$$f(i_1, j_2, m_1, m_2) = \min_{k=1,\ldots,\min\{C(i_1),i\}} \{2d_{i_1} + f((i-k)_1, ((j - C(i_1) + k)^+)_2, m_1 - 1, m_2)\}$$

and for the opposite case $j_2 > i_1$:

$$f(i_1, j_2, m_1, m_2) = \min \left\{ \begin{array}{l} \min\limits_{k=0,\ldots,\min\{C(j_2),i\}} \{2d_{j_2} + f((i-k)_1, ((j - C(j_2) + k)^+)_2, m_1 - 1, m_2)\}, \\ 2d_{j_2} + f(i_1, ((j - C(j_2))^+)_2, m_1, m_2 - 1) \end{array} \right\}.$$

The second recursion (implicitly) compares three situations: at least one D1-customer included, only D2-customers with service on day 1, only D2-customers with service on day 2. By the restriction of the choice of k in the recursion it is avoided that $i - k$ could become less than 0. But $j - C(n) + k < 0$ may occur (though never optimal in combination with $i - k > 0$). Define the following starting conditions:

$$\begin{array}{ll} f(0, 0, m_1, m_2) = 0 & \forall m_1, m_2 \geq 0 \\ f(i_1, j_2, x, m_2) = \infty & \forall i_1 > 0, m_2 \geq 0, x \leq 0 \\ f(i_1, j_2, x, y) = \infty & \forall \max\{i_1, j_2\} > 0, x, y \leq 0. \end{array}$$

4.2 | Including customers with period 1

It remains to extend the proof to the general case in which customers with period 1 (service is required on both days) can be present. First, we argue that the same reasoning as in Lemma 2 still provides an optimal solution if there are customers with period 1. Second, the DP is adjusted to cover for these customers.

First, interpret P1-customers as two customers, where the first is an additional D1-customer and the second customer must be served on day 2. This observation leads to three sets of customers. The first set of customers $\mathcal{A}$, with indices $1_1, 2_1, \ldots, I_1$, has to be served on day 1. Note that set $\mathcal{A}$ contains both the original D1-customers and the converted D1-customers. The second set $\mathcal{B}$, with indices $1_2, 2_2, \ldots, J_2$, contains the D2-customers. Finally, the third set of customers $\mathcal{C}$, with indices $1_3, 2_3, \ldots, \Lambda_3$, must be served on day 2.

The need for having three sets is illustrated in Figure 2. Suppose that A_1 has already been assigned to a vehicle, so that A_2 is the farthest unassigned customer. Further, assume that $C(B) = 3$ and $C(A_2) = 2$. Now, it is optimal to serve A_2 and D_2 together on day 2 and to serve B, C and D_1 together on day 1. Observe that in the optimal solution there is a vehicle that does not serve consecutive customers of $\mathcal{B} \cup \mathcal{C}$. However, it does serve consecutive customers of every set as defined above, which holds for any optimal solution and is formalized in Lemma 3.

Lemma 3. *Starting from the farthest customer and moving toward the depot, no vehicle skips a customer in set $\mathcal{Z}$ to serve another customer in set $\mathcal{Z}$ closer to the depot, for $\mathcal{Z} \in \{\mathcal{A}, \mathcal{B}, \mathcal{C}\}$.*

The proof of Lemma 3 is similar the proof of Lemma 2 and is omitted for reasons of conciseness.

To describe the dynamic programming algorithm we define the following notation. Again, let n be the index of the customer farthest from the depot that still needs assignment to a vehicle, and let i_1, j_2 and λ_3 be the indices of the farthest customers in

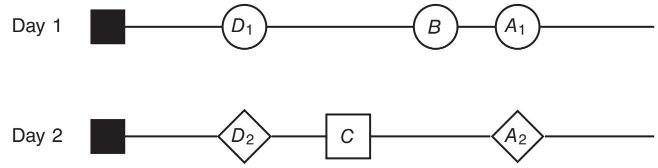

FIGURE 2 Example in which elements of $\mathcal{A}$ are represented by circles, elements of $\mathcal{B}$ by squares, and elements of $\mathcal{C}$ by diamonds

sets $\mathcal{A}$, $\mathcal{B}$ or $\mathcal{C}$, respectively, that still needs assignment. Hence, $n = \max\{i_1, j_2, \lambda_3\}$. Let M be the total number of vehicles available per day and let m_1 and m_2 be the number of remaining available vehicles for day 1 and 2, respectively. Redefine k to be the number of customers from set $\mathcal{A}$ that are served by a vehicle on day 1 and define ℓ to be the number of customers from set $\mathcal{C}$ that are served by a vehicle on day 2. Note that if $n = j_2$, then depending on the day j_2 is served, either customers from $\mathcal{A}$ or customers from $\mathcal{C}$ can be served by the same vehicle that serves j_2. Define the value 0 for the indices i_1, j_2 and λ_3 for the case that no customer exists in sets $\mathcal{A}$, $\mathcal{B}$ or $\mathcal{C}$, respectively, equivalent to the definition in Section 4.1.

Denote the current state of the DP by $<i_1, j_2, \lambda_3, m_1, m_2>$ and let $f(i_1, j_2, \lambda_3, m_1, m_2)$ be the minimal cost (time) of serving all customers in this state. The total minimal costs of an instance is given by $f(I_1, J_2, \Lambda_3, M, M)$. Three recursion formulas define the DP: the first one for $\max\{i_1, j_2, \lambda_3\} = i_1$, that is, a customer in $\mathcal{A}$ is the farthest unassigned customer:

$$f(i_1, j_2, \lambda_3, m_1, m_2) = \min_{k=1,\dots,\min\{C(i_1),i\}} \{2d_{i_1} + f((i-k)_1, ((j - C(i_1) + k)^+)_2, \lambda_3, m_1 - 1, m_2)\}$$

the second one for $\max\{i_1, j_2, \lambda_3\} = j_2$:

$$f(i_1, j_2, \lambda_3, m_1, m_2) = \min \left\{ \begin{array}{l} \min\limits_{k=0,\dots,\min\{C(j_2),i\}} \{2d_{j_2} + f((i-k)_1, ((j - C(j_2) + k)^+)_2, \lambda_3, m_1 - 1, m_2)\}, \\ \min\limits_{\ell=0,\dots,\min\{C(j_2),\lambda\}} \{2d_{j_2} + f(i_1, ((j - C(j_2) + \ell)^+)_2, (\lambda - \ell)_3, m_1, m_2 - 1)\} \end{array} \right\}$$

and the third one for $\max\{i_1, j_2, \lambda_3\} = \lambda_3$:

$$f(i_1, j_2, \lambda_3, m_1, m_2) = \min_{\ell=1,\dots,\min\{C(\lambda_3),\lambda\}} \{2d_{\lambda_3} + f(i_1, ((j - C(\lambda_3) + \ell)^+)_2, (\lambda - \ell)_3, m_1, m_2 - 1)\}.$$

The following starting conditions hold:

$$\begin{array}{ll} f(0,0,0,m_1,m_2) = 0 & \forall m_1, m_2 \geq 0 \\ f(i_1, j_2, \lambda_3, x, m_2) = \infty & \forall i_1 > 0, m_2 \geq 0, x \leq 0 \\ f(i_1, j_2, \lambda_3, m_1, x) = \infty & \forall \lambda_3 > 0, m_1 \geq 0, x \leq 0 \\ f(i_1, j_2, \lambda_3, x, y) = \infty & \forall \max\{i_1, j_2, \lambda_3\} > 0, x, y \leq 0. \end{array}$$

4.3 | Running time and generalizations

The running time of the DP is polynomial. To see this, note that the number of states to be considered is $\mathcal{O}(N^3M^2)$ and each computation of the recursion takes $\mathcal{O}(N)$ time. Moreover, we may assume without loss of generality that $M \leq N$. Hence, our DP runs in $\mathcal{O}(N^6)$ time.

Theorem 1. *Problem $M \mid half\text{-}line \mid s \mid 2$ can be solved in $\mathcal{O}(N^6)$ time.*

A first possible generalization of problem $M \mid half\text{-}line \mid s \mid 2$ is extending the planning horizon to more than 2 days, for example 3 days. Then, immediately an essentially different ingredient is added to the problem: serving a D2-customer with period 2 on day 1 leads to the obligation of serving it again latest on day 3. This periodicity, asking for repetitive service, is the main issue in problems related to the PSP. As mentioned in Section 2.2, this is a badly-understood problem.

A second generalization concerns the underlying metric spaces. The DP can be adapted to yield a polynomial time algorithm for problem $M|\cdot|s|2$ on a line or a cycle. Furthermore, the problem on a tree is NP-hard, even if the planning horizon is a single day and the tree is a star with the depot at the center. This follows again through equivalence to BPP, because the travel time to each customer can be different. Section 5 considers the IRP as defined in Section 2 in the Euclidean plane.

4.4 | Vehicle capacity

Another aspect that can be incorporated in the studied IRP, is vehicle capacity in terms of load, that is, the maximum number of units demand that can be delivered by a vehicle in 1 day. To facilitate the exposition of the impact of vehicle capacity constraints, consider the case which discards service times.

In case the demand of each customer can be different, similar argumentation as in Section 3.2.1 for problems with arbitrary service times can be used to establish NP-hardness for all problems with more than one vehicle or more than 1 day. In case all demands are identical, the same argumentation can be followed as for equal service times ($s_i = s\,\forall\, i$) and tour duration limit L in all problem variants. Also the DP still holds for the equivalent problem by only adjusting the definition of the function $C(\ell)$ that defines the number of customers that can be served by a vehicle given that customer ℓ is served. Note that if a vehicle only has a capacity constraint instead of a tour duration constraint, the maximal number of customers that can be served $C(\ell)$ is the same for any customer ℓ. By discarding service times and replacing them with uniform demands for the customers and a capacity constraint on the vehicle, solving the problem on a star is no longer hard since BPP in terms of service time is no longer an underlying hard problem.

5 | INVENTORY ROUTING IN THE EUCLIDEAN PLANE

Consider the IRP as defined in Section 2 in the Euclidean plane with a single vehicle, uniform service times at the customers ($s_i = s\,\forall\, i$) and a time horizon of Z days, denoted by $1|plane|s|Z$. Again, the travel time is equal to the total distance traveled. To avoid immediate NP-hardness from routing, we approximate the route length which provides an easy route length computation, instead of computing the exact optimal route length which requires solving TSPs [28]. In spite of trivializing the routing cost computation, we show that the resulting problem is NP-hard.

This variant of the IRP is interesting to investigate theoretically, given the discussion on the PSP in Section 2.2, but also has a practical application [8]. For the tour length approximation, we use a result of Beardwood et al. [9] who show that the tour length is asymptotically equal to $\phi\sqrt{A \cdot N}$ for large N, where ϕ is a constant and A is the surface of the area in which the N points can be placed uniformly at random. Chien [14] considers approximation functions with a similar functional form, but considers several areas for A which take the actual depot and customer locations into account instead of the area in which the customers can be located as in Beardwood et al. [9]. As an approximation to the route length we use the same functional form as Beardwood et al. [9] and Chien [14], and compute the area of the customers as the convex hull of the locations of the customers. The objective is to find a feasible solution, obeying customer periods and the tour duration constraint, that minimizes the total approximated route length. This section shows strong NP-hardness for the studied IRP with this tour length approximation as route length function.

Note that this IRP with the route length approximation is a generalization of the PSP in which the tasks are executed at different locations and more than one task can be scheduled per day. Hence, this section shows that we can prove NP-hardness of a generalization of the PSP in which the tasks are executed at different locations without using the hardness of TSP. Besides that, this IRP has features of the joint replenishment problem (JRP), which is an NP-hard problem [6]. The JRP is a multiperiod replenishment problem in which a fixed fee is incurred per customer replenishment and per period in which at least one customer is replenished. Since in the JRP a fixed fee is paid per served customer, the JRP is not a special case of the IRP with approximated route length because of the different cost structure.

A reduction from 3-partition [21] shows strong NP-hardness for the studied IRP in the plane with approximated route length. 3-Partition is defined as follows: given $3m$ integers $a_1, \ldots, a_{3m}$ such that $\sum_{i=1}^{3m} a_i = mB$, the question is whether there exists a partition in sets $S_1, \ldots, S_m$ such that $|S_j| = 3$ and $\sum_{i \in S_j} a_i = B$ for $j = 1, \ldots, m$. The problem is hard even if $B/4 < a_i < B/2$ for $i = 1, \ldots, 3m$.

Theorem 2. *IRP in the plane with approximated route length is strongly NP-hard.*

Proof. Given an instance of 3-partition, create the following instance of IRP in the plane. Create customers with two different periods, period 1 and period m. Take $(c_1, \ldots, c_{3m})$ as the $3m$ extreme points of a regular polygon with area P. One of these points is chosen as the depot. Let all other points contain a customer with period 1.

Second, choose another $3m$ locations outside the polygon, each making a triangle of area Q with two neighboring extreme points of the regular polygon, as depicted in Figure 3. Let each of these locations have a set of customers with period m. At the first such location a number of a_1 customers is located, corresponding to the integer value a_1 from the 3-partition instance, at the second location a_2 customers are located, and so on. Moreover, P and Q are chosen such that an angle θ in Figure 3 is at most 180°. There is a single vehicle with tour duration limit $\phi\sqrt{(P + 3Q) \cdot (B + 3m)}$.

Thus, there are $\sum_{i=1}^{3m} a_i = mB$ Pm-customers spread over $3m$ locations. We will show that there is a 3-partition if and only if there is a feasible schedule in which no customer is out of stock and every day the approximated length of the tour is at most $\phi\sqrt{(P + 3Q) \cdot (B + 3m)}$.

If there is a 3-partition, serve all customers at the location corresponding to integer a_i for each a_i in set S_j on day j. Moreover, all $3m$ P1-customers are served every day. Clearly, this is a feasible solution for the IRP instance. It remains

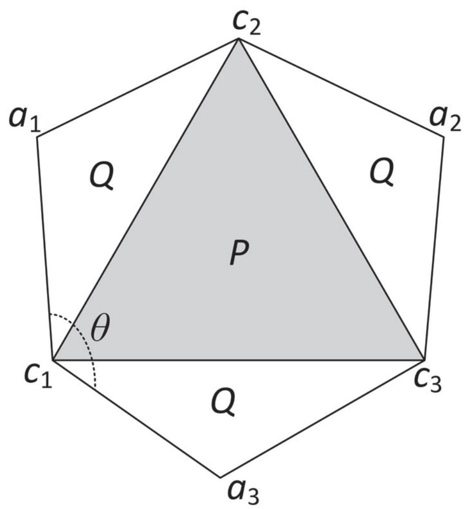

FIGURE 3 Positions customers for $m = 1$

to show that the bound on the route length holds. The total area per day to be covered in the route length function is P for the $3m$ customers with period 1 and Q per set a_i. Since there are exactly three such sets every day, the total area per day is $P + 3Q$. The number of customer services is $3m$ for the customers with period 1 plus $\sum_{k\in S_j} a_k$ for the selected sets on day j which is exactly equal to B by the 3-partition. Hence, each day the approximated length of a tour is $\phi\sqrt{(P+3Q)\cdot(B+3m)}$.

Reversely, if there is a feasible schedule for the planning horizon of m days for which the approximated route length on each day is at most $\phi\sqrt{(P+3Q)\cdot(B+3m)}$, a feasible 3-partition can be derived. First, since $\sum_i a_i = mB$, B Pm-customers and $3m$ P1-customers should be served on average per day. Since $a_i < B/2$, serving more than $B + 3m$ customers can only be done by adding at least three times Q to the area of the convex hull. Hence, exactly $B + 3m$ customers will be served every day. This means that the set of Pm-customers can be partitioned into subsets with exactly B customers each. Then, because $B/4 < a_i < B/2$ for all i, on each day exactly three sets of customers a_i are served. Hence, the schedule corresponds to a feasible 3-partition. ■

6 | CONCLUSION

The main positive result in this article is a polynomial time dynamic programming algorithm for the borderline problem variant $M\,|\,half\text{-}line\,|\,s\,|\,2$. In this problem on the half-line the planning horizon is 2 days, there are $M > 1$ vehicles available to serve the customers with uniform service times s.

If we extend the planning horizon from 2 days to any number of days, the problems, even on a point, are at least as hard as the PSP, for which the complexity has not been determined. This is due to the fixed periods of the customers, which leads to a compact input description. The complexity of the considered IRP is open if the number of days is fixed but greater than 2, for example, if $M\,|\,half\text{-}line\,|\,s\,|\,3$. These problem variants may very well be polynomially solvable, independent of the complexity of the PSP.

Less surprising is that allowing customers to have arbitrary service times introduces bin packing aspects into the problem, making the resulting IRP NP-hard, even when defined on a point. Our results show that not only the presence of a routing problem contributes to the complexity of the IRP, but also the service times and the periodicity of replenishments of the customers.

ACKNOWLEDGMENTS

A.C.B. and M.H. were funded by the Netherlands Organisation for Scientific Research (NWO), project number 407-13-050. M.v.E. was funded by NWO, project number 612.001.215. M.v.E. was employed by Vrije Universiteit Amsterdam when working on this article. Research by L.S. was partially supported by the NWO through the Gravitation Programme Networks (024.002.003). The authors want to thank the anonymous reviewers and the editors for their valuable comments.

ORCID

Annelieke C. Baller https://orcid.org/0000-0002-3826-0057

REFERENCES

[1] A. Alvarez, P. Munari, and R. Morabito, *Iterated local search and simulated annealing algorithms for the inventory routing problem*, Int. Trans. Oper. Res. **25** (2018), 1785–1809.

[2] H. Andersson, A. Hoff, M. Christiansen, G. Hasle, and A. Løkketangen, *Industrial aspects and literature survey: Combined inventory management and routing*, Comput. Oper. Res. **37** (2010), 1515–1536.

[3] C. Archetti, N. Bianchessi, S. Irnich, and M.G. Speranza, *Formulations for an inventory routing problem*, Int. Trans. Oper Res. **21** (2014), 353–374.

[4] C. Archetti, N. Boland, and M. Grazia Speranza, *A matheuristic for the multivehicle inventory routing problem*, INFORMS J. Comput. **29** (2017), 377–387.

[5] C. Archetti, D. Feillet, M. Gendreau, and M.G. Speranza, *Complexity of the VRP and SDVRP*, Transp. Res. C: Emerg. Technol. **19** (2011), 741–750.

[6] E. Arkin, D. Joneja, and R. Roundy, *Computational complexity of uncapacitated multi-echelon production planning problems*, Oper. Res. Lett. **8** (1989), 61–66.

[7] F. Baita, W. Ukovich, R. Pesenti, and D. Favaretto, *Dynamic routing-and-inventory problems: A review*, Transp. Res. A: Policy Pract. **32** (1998), 585–598.

[8] A.C. Baller, S. Dabia, W.E.H. Dullaert, and D. Vigo, *The dynamic-demand joint replenishment problem with approximated transportation costs*, European J. Oper. Res. **276** (2019), 1013–1033.

[9] J. Beardwood, J.H. Halton, and J.M. Hammersley, *The shortest path through many points*, Math. Proc. Cambridge Philos. Soc. **55** (1959), 299–327.

[10] L. Bertazzi, M. Savelsbergh, and M.G. Speranza, "*Inventory routing*," The Vehicle Routing Problem: Latest Advances and New Challenges **43**, B.L. Golden, S. Raghavan, E.A. Wasil (Eds.), Springer, Boston MA, 2008, pp. 49–72.

[11] L. Bertazzi and M.G. Speranza, *Inventory routing problems: An introduction*, EURO J. Transp. Logist. **1** (2012), 307–326.

[12] L. Bertazzi and M.G. Speranza, *Inventory routing problems with multiple customers*, EURO J. Transp. Logist. **2** (2013), 255–275.

[13] T. Bosman, M. van Ee, Y. Jiao, A. Marchetti-Spaccamela, R. Ravi, and L. Stougie, "*Approximation algorithms for replenishment problems with fixed turnover times*," *LATIN 2018: Theoretical Informatics. LATIN 2018. Lecture Notes in Computer Science*, vol. **10807**, M. Bender, M. Farach-Colton, and M. Mosteiro (eds), Springer, Cham, 2018, pp. 217–230.

[14] T.W. Chien, *Operational estimators for the length of a traveling salesman tour*, Comput. Oper. Res. **19** (1992), 469–478.

[15] L.C. Coelho, J.F. Cordeau, and G. Laporte, *Thirty years of inventory routing*, Transp. Sci. **48** (2014), 1–19.

[16] J.F. Cordeau, G. Laporte, M.W. Savelsbergh, and D. Vigo, "*Vehicle routing*," *Handbooks in Operations Research and Management Science*, vol. **14**, Elsevier, North Holland, 2007, pp. 367–428.

[17] A. Das, C. Mathieu, and S. Mozes, "*The train delivery problem—Vehicle routing meets bin packing*," *Approximation and Online Algorithms. WAOA 2010. Lecture Notes in Computer Science*, vol. **6534**, K. Jansen and R. Solis-Oba (eds), Springer, Berlin, Heidelberg, 2011, pp. 94–105.

[18] W.E. de Paepe, J.K. Lenstra, J. Sgall, R.A. Sitters, and L. Stougie, *Computer-aided complexity classification of dial-a-ride problems*, INFORMS J. Comput. **16** (2004), 120–132.

[19] G. Desaulniers, J.G. Rakke, and L.C. Coelho, *A branch-price-and-cut algorithm for the inventory-routing problem*, Transp. Sci. **50** (2016), 1060–1076.

[20] A. Federgruen and D. Simchi-Levi, *Analysis of vehicle routing and inventory-routing problems*, Handb. Oper. Res. Manag. Sci. **8** (1995), 297–373.

[21] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to NP-Completeness*, WH Freeman, New York, 1979.

[22] R.L. Graham, E.L. Lawler, J.K. Lenstra, and A.H.G. Rinnooy Kan, *Optimization and approximation in deterministic sequencing and scheduling: A survey*, Ann. Discrete Math. **5** (1979), 287–326.

[23] V. Hemmelmayr, K.F. Doerner, R.F. Hartl, and M.W.P. Savelsbergh, *Delivery strategies for blood products supplies*, OR Spectrum **31** (2009), 707–725.

[24] R. Holte, A. Mok, L. Rosier, I. Tulchinsky, and D. Varvel, "*The pinwheel: A real-time scheduling problem*," *Proceedings of the 22nd Annual Hawaii International Conference on System Sciences. Volume II: Software Track*, IEEE, vol. **2**, 1989, pp. 693–702.

[25] T. Jacobs and S. Longo, *A New Perspective on the Windows Scheduling Problem*, arXiv:1410.7237 [cs.CC], 2014.

[26] R.M. Karp, "*Reducibility among combinatorial problems*," *Complexity of Computer Computations*, Raymond E. Miller, James W. Thatcher and Jean D. Bohlinger (eds.), Springer, Boston, MA. 1972, pp. 85–103.

[27] N.H. Moin and S. Salhi, *Inventory routing problems: A logistical overview*, J. Oper. Res. Soc. **58** (2007), 1185–1194.

[28] C.H. Papadimitriou, *The Euclidean traveling salesman problem is NP-complete*, Theoret. Comput. Sci. **4** (1977), 237–244.

[29] A.M. Sarmiento and R. Nagi, *A review of integrated analysis of production-distribution systems*, IIE Trans. **31** (1999), 1061–1074.

How to cite this article: Baller AC, van Ee M, Hoogeboom M, Stougie L. Complexity of inventory routing problems when routing is easy. *Networks.* 2019;1–11. https://doi.org/10.1002/net.21908