INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

IFORS

WILEY

INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

Intl. Trans. in Op. Res. 27 (2020) 314–335
DOI: 10.1111/itor.12540

A variable neighborhood search simheuristic for the multiperiod inventory routing problem with stochastic demands

Aljoscha Gruler[a], Javier Panadero[a], Jesica de Armas[b], José A. Moreno Pérez[c] and Angel A. Juan[a]

[a] *IN3—Computer Science Department, Open University of Catalonia, Castelldefels, Spain*
[b] *Department of Economics and Business, Universitat Pompeu Fabra, Barcelona, Spain*
[c] *Instituto Universitario de Desarrollo Regional, Universidad de La Laguna, Tenerife, Spain*
E-mail: agruler@uoc.edu [Gruler]; jpanaderom@uoc.edu [Panadero]; jesica.dearmas@upf.edu [de Armas]; jamoreno@ull.es [Pérez]; ajuanp@uoc.edu [Juan]

Received 5 February 2017; received in revised form 2 February 2018; accepted 8 March 2018

Abstract

The inventory routing problem (IRP) combines inventory management and delivery route-planning decisions. This work presents a simheuristic approach that integrates Monte Carlo simulation within a variable neighborhood search (VNS) framework to solve the multiperiod IRP with stochastic customer demands. In this realistic variant of the problem, our goal is to establish the optimal refill policies for each customer–period combination, that is, those individual refill policies that minimize the total expected cost over the periods. This cost is the aggregation of both expected inventory and routing costs. Our simheuristic algorithm allows to consider the inventory changes between periods generated by the realization of the random demands in each period, which have an impact on the quantities to be delivered in the next period and, therefore, on the associated routing plans. A range of computational experiments are carried out in order to illustrate the potential of our simulation–optimization approach.

Keywords: multiperiod inventory routing problem; stochastic demands; variable neighborhood search; metaheuristics; simheuristics

1. Introduction

Freight logistics and road transportation is of micro- and macroscopic importance. On the one hand, the costs associated with moving and storing goods constitute a major economic factor for organizations of different sectors (European Commission, 2016). On the other hand, the usage of freight delivery vehicles leads to negative externalities such as air pollution, excessive noise, and traffic congestion (United Nations, 2011; European Union, 2012; United States Environmental Protection Agency, 2013). While practical and theoretical discussions related to freight logistics

Published by John Wiley & Sons Ltd, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main St, Malden, MA02148, USA.

and transportation (L&T) have long focused on the optimization of single company operations, collaborative business strategies are gaining increased attention (Ming et al., 2014; Ramanathan and Gunasekaran, 2014). By sharing customer orders, information, and resources, companies operating in the same supply chain hope to decrease overall value chain costs through centralized L&T planning.

In this context, the concept of vendor-managed inventories (VMIs) is based on the transfer of inventory management decisions to the central supplier. In a VMI, a total of n retail centers (RCs) is stocked from a single supplier located at a central warehouse. Inventory levels at each RC and necessary replenishment quantities delivered by the supplier are defined through final customer demands. In noncollaborative supply chains, each RC defines its own replenishment levels based on its own inventory management decisions, which directly affects the delivery route planning process of the supplier. On the contrary, the implementation of VMI centralizes inventory and routing decisions at the supplier, allowing the optimization of both decisions from an overall supply chain perspective (Andersson et al., 2010; Coelho et al., 2013).

From an optimization point of view, this supply chain strategy is represented by the inventory routing problem (IRP). This combinatorial optimization problem (COP) can be seen as an extension to the well-known vehicle routing problem (VRP), making the problem setting NP-hard (Lenstra and Kan, 1981; Caceres-Cruz et al., 2014). Due to the inherent complexity of the IRP, especially metaheuristic solving methodologies are used to create inventory and routing plans for large-sized IRP instances. As is the case of other COPs however, most solving frameworks are only capable of providing oversimplified problem solutions, in which the natural uncertainty of most real-life systems is left unaccounted for. This paper overcomes this drawback by presenting a simheuristic extension of the popular and efficient variable neighborhood search (VNS) metaheuristic (Mladenović and Hansen, 1997; Hansen et al., 2010) for solving the multiperiod IRP with stochastic customer demands at each RC (Fig. 1). Simheuristic algorithms integrate a metaheuristic framework with simulation to address optimization problems under uncertainty conditions (Juan et al., 2015; Grasas et al., 2016).

Through the integration of Monte Carlo Simulation (MCS) inside the VNS procedure, our simheuristic algorithm is able to consider random realizations of customer demands and, accordingly, changes in the inventory levels, delivery orders, and routing plans over a multiperiod planning horizon. In this realistic variant of the problem, our goal is to establish the optimal refill policies for each customer–period combination, that is, those individual refill policies that minimize the total expected cost over the periods. This total expected cost is computed as the aggregation of both expected inventory and routing costs.

This paper extends the work of Juan et al. (2014), who combined a simple multistart heuristic with simulation to solve the single-period IRP under demand uncertainty. The main contributions of this work can be summarized as follows: (a) a new simheuristic algorithm for the multiperiod IRP with stochastic demands is presented—the algorithm integrates simulation into a VNS metaheuristic; and (b) the algorithm is tested in a series of computational experiments that illustrate its potential. The remainder of this paper is structured as follows: relevant literature on the IRP is reviewed in Section 2; a mathematical formulation of the multiperiod IRP with stochastic demands is provided in Section 3; Section 4 outlines the proposed algorithm; a range of computational experiments are described and analyzed in Section 5; finally, Section 6 concludes this work and highlights future research lines.

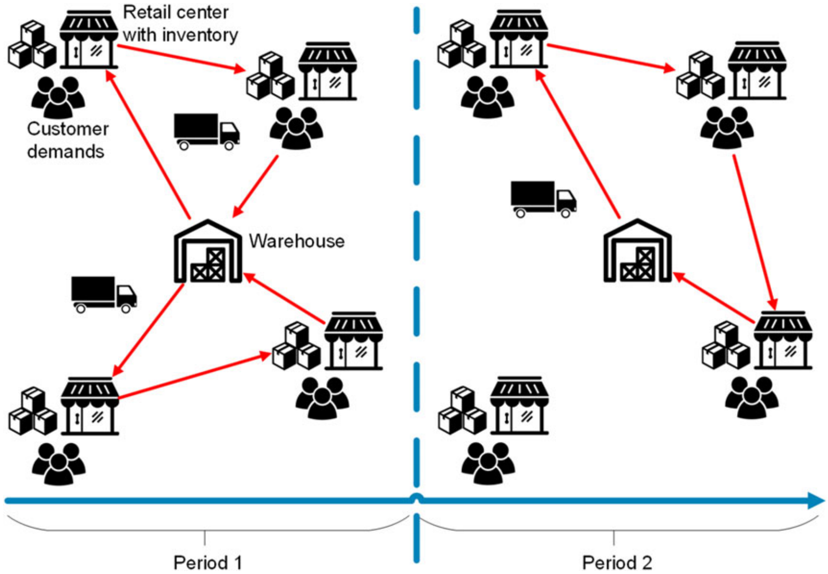

Fig. 1. A simple two-period inventory routing problem.

Table 1
Structural IRP variants and relationship of this paper to closely related works

Dimension	Structural variants			
Time horizon	Single-period[a]	Multi-period[b,c,d]	Cyclic	–
Supply chain structure	One-to-one	One-to-many[a,b,c,d]	Many-to-many	Many-to-one
Inventory policy	Maximum-level[a,d]	Order-up-to level[b,c]	–	–
Inventory decisions	Stockouts[a,d]	Backlogging[b,c]	Nonnegative	–
Input variables	Deterministic	Stochastic[a,b,c,d]	–	–

[a]Juan et al. (2014).
[b]Bertazzi et al. (2013).
[c]Solyalı et al. (2012).
[d]This paper.

2. Literature review

Two extensive literature reviews on different IRP settings and solving techniques are presented by Andersson et al. (2010) and Coelho et al. (2013). Apart from other structural problem variants related to vehicle fleet compositions and further routing-related aspects, the most important IRP problem alternatives are highlighted in Table 1. For a clearer distinction, the characterization of this work and the most closely related publications are depicted. In this context, the problem setting addressed in this paper establishes inventory and routing plans over multiple time periods in a one-to-many supply chain setting. In contrast to an order-up-to-level inventory strategy that defines a global refill stock level for all RCs, this work discusses the maximum-level strategy, which defines an individual inventory plan at each client for every time period. Furthermore, the possibility of product stockouts at the end of each period is considered. Final customer demands are considered to be of stochastic nature. Some metaheuristic solving methodologies for deterministic and stochastic IRP variants are discussed in more detail in the following.

2.1. Metaheuristic solving methodologies for the deterministic IRP

Liu and Lee (2011) propose a variable neighborhood tabu search (VNTS) for the IRP with time windows. Based on an initial solution, different neighborhood structures in reference to vehicle routing and inventory control strategies are investigated within the TS framework. The authors consider up to 100 customers over 100 planning periods. Different neighborhood structures are also used in the adaptive large neighborhood search (ALNS) metaheuristic developed by Aksen et al. (2014). The authors study a selective and periodic IRP in the collection of vegetable oil from different origin nodes. Problem settings with up to 100 source nodes and a 7-day planning horizon are addressed. Popović et al. (2012) developed a VNS algorithm for a multiproduct, multiperiod IRP in fuel delivery with homogeneous multicompartment vehicles. A two-phased VNS for the multiproduct IRP is discussed by Mjirda et al. (2014). During the first solution stage, the associated VRP is solved, which is then iteratively improved considering both transportation and inventory costs. Unlike previously cited authors, a many-to-one supply chain network in which different suppliers serve a single assembly line is discussed.

Abdelmaguid et al. (2009) put forward a heuristic approach for the finite multiperiod IRP with backlogging. After applying a constructive heuristic to estimate the transportation costs for each customer in each considered period, an improvement heuristic is applied. Then, customer delivery amounts are exchanged between periods to improve the initial solution. They report solutions for problem settings with up to 15 customers and a 7-day planning horizon. The multiperiod IRP is also addressed by Archetti et al. (2012). However, stockout situations are not considered in their work. The authors develop a hybrid heuristic based on a tabu search scheme combined with ad hoc designed mixed-integer programming models (Juan et al., 2015; Grasas et al., 2016). A three-phase heuristic for the multiproduct, multiperiod IRP is discussed by Cordeau et al. (2015), whereby each stage represents a decision process. First, replenishment plans are constructed using a Langragian-based method, specifying which customer to serve and how much to deliver during each period. Second, the vehicle routes are constructed. Finally, planning and routing decisions are combined into a mixed-integer linear programming model. They solve problems with up to 50 customers and five different products.

The same case of inbound logistics with up to 98 suppliers is also studied by Moin et al. (2011) to test their hybrid genetic algorithm (GA) for the multiproduct, multiperiod IRP. The algorithm follows the allocation-first, route-second strategy. A GA for the multiperiod IRP in a one-to-many supply chain network and the consideration of lost sales was recently presented by Park et al. (2016). Their experiments are performed using problem sets with up to 12 customers and 12 time periods. The special case of inventory routing in the petroleum and petrochemical industry is addressed by Li et al. (2014), who propose a TS metaheuristic. With the objective of minimizing route travel times, particularities of this problem setting include the high importance of avoiding stockouts and other operational constraints. A population-based simulated annealing algorithm for the multiproduct, multiretailer IRP with perishable goods is presented by Shaabani and Kamalabadi (2016).

Nambirajan et al. (2016) also extend the classic IRP formulation. A closer supply chain collaboration is considered by including replenishment activities at a central depot and different warehouses in a three echelon supply chain. First, the replenishment policy of a set of manufacturers to a single depot is defined. Then, the routing of the central depot to multiple warehouses is planned by using a three-stage heuristic based on clustering, allocation, and routing. An iterated local search algorithm for the cyclic IRP over an infinite planning horizon is discussed by Vansteenwegen and Mateo (2014). Other heuristic and metaheuristic techniques for the cyclic IRP have also been presented by Chitsaz et al. (2016), Raa and Dullaert (2017), and Zachariadis et al. (2009).

2.2. Metaheuristic solving methodologies for the stochastic IRP

Unlike for the deterministic case, literature on the IRP under uncertainty is more scarce. Random demands for inventory routing over an infinite horizon is addressed in Jaillet et al. (2002), who present incremental cost approximations in a rolling horizon framework. The stochastic IRP is formulated as Markov decision process by Adelman (2004). The author applies cost approximations by using dual prices of a linear program. Further, approximation methods for the IRP with demand uncertainty modeled as a Markov decision problem are discussed by Kleywegt et al. (2004). Another approach in which the stochastic IRP is modeled as Markov decision process is presented in Hvattum and Løkketangen (2008) and Hvattum et al. (2009). The authors model random demand through general discrete distributions, while their solution framework is based on the use of scenario trees. Solutions to the scenario tree problem are obtained by using a standard MIP solver, a greedy randomized adaptive search procedure (GRASP), and a progressive hedging algorithm (PHA).

More recent work on stochastic inventory routing include the one of Bertazzi et al. (2013), who consider an IRP with stockouts and a finite horizon, solved with a dynamic programming model and a hybrid roll-out algorithm. Bertazzi et al. (2015) address the IRP with stochastic demand and transportation procurement by developing a rollout-based matheuristic algorithm. A similar problem is addressed in a robust optimization approach through MILP formulations by Solyalı et al. (2012). Especially the works of Bertazzi et al. (2013) and Solyalı et al. (2012) discuss other variants of the problem addressed in this paper. Thus, while the former considers an order-up-to-level inventory strategy—in which each RC is filled to its maximum capacity each time it is visited—the main goal in our version is precisely to find out the optimal values for these refill strategies. Also, our approach allows to model customers' demands by using different probability distributions, while Solyalı et al. (2012) is a robust-based approach that assumes a uniform random behavior for these demands.

Huang and Lin (2010) develop a modified ant colony optimization metaheuristic for the multi-product IRP with demand uncertainty. A robust inventory routing policy, considering stochastic customer demands and replenishment lead times, is discussed by Li et al. (2016). The robustness of inventory replenishment and customer selection policies for the dynamic and stochastic IRP is addressed in Roldán et al. (2016). Yu et al. (2012) solve the stochastic IRP with split deliveries and service-level constraints. Soysal et al. (2015) include environmental concerns in their solution for the IRP with demand uncertainty by estimating CO_2 emissions in the route planning process. Greenhouse gas emissions are also minimized in the work of Niakan and Rahimi (2015), who propose a fuzzy possibilistic approach to a multiobjective IRP for medical drug distribution. Rahim et al. (2014) solve the multiperiod IRP with stochastic stationary demand through a deterministic

Table 2
Symbols

Symbol	Indices	Type	Meaning
P	$p = 1, ..., \|P\|$	Data	Set of periods included in the planning horizon
V	$i, j = 0, 1, ..., n$	Data	Set of locations, including the depot 0 and n RCs
V^*	$i, j = 1, ..., n$	Data	Set of RCs ($V^* = V \setminus \{0\}$)
c_{ij}	$i, j \in V$	Data	Cost of traveling from $i \in V$ to $j \in V$
$h > 0$		Data	Maximum vehicle capacity
$l_i^+ > 0$	$i \in V$	Data	Maximum storage capacity of RC i
L_{ip}^0	$i \in V$	Var	Initial stock available at RC i during period p
D_{ip}	$i \in V^*, \quad p \in P$	Var	Customers' aggregated demand at RC i during a period p
$q_{ip} \geq 0$	$i \in V^*, \quad p \in P$	Var	Quantity served at RC i at the beginning of period p
$y_{ip} \geq 0$	$i \in V^*, \quad p \in P$	Var	Binary variable that defines if RC i is visited at period p
x_{ij}^{pk}	$i, \quad j \in V, \quad k \in K$	Var	Binary variable that defines if vehicle k goes from i to j at period p

equivalent approximation model. Also addressing the IRP with stochastic demands, Chen and Lin (2009) consider a real life multiperiod, multiproduct case. In addition, the authors incorporate risk aversion into their model. Finally, Juan et al. (2014) propose a hybrid simulation–optimization approach, combining a multistart metaheuristic with MCS, to address the single-period IRP with stochastic demands.

3. Formal problem description

This section formally models the periodic IRP with stochastic customers' demands at each RC. Even though a mathematical formulation is not essentially necessary for the proposed metaheuristic-based solution technique, it allows for a better understanding of the complex decision-making problem. The used symbols for data and variables are listed in Table 2.

Let $V = \{0, 1, \ldots, n\}$ denote a finite set of locations consisting of the depot (node 0) and n RCs. The set of RCs will be denoted by $V^* = V \setminus \{0\}$. With the goal of minimizing total expected cost, the stochastic and periodic IRP combines inventory and routing decisions over a finite planning horizon P with $|P| \geq 1$ periods. The customers' aggregated demand at each RC $i \in V^*$ during a period $p \in P$ is a random variable, D_{ip}, which follows a known probability distribution. In this work, it is assumed that these random demands are independent across RCs and throughout periods—although they do not need to be identically distributed. Likewise, it will be assumed that the customers' aggregated demand at each RC and period will always be satisfied. Thus, should a stockout occur during a period p at RC i, an additional shipment from the depot to i will be placed by the end of period p to cover the nonsatisfied demand—the cost of this extra shipment will be accounted as stockout cost.

Regarding inventory management, the decision variables refer to the quantity of product, $q_{ip} \geq 0$, that must be served at RC i at the beginning of period p ($\forall i \in V^*, \forall p \in P$). If $l_i^+ > 0$ denotes the maximum storage capacity of RC i, and L_{ip}^0 denotes the initial stock available at RC i during period p ($0 \leq L_{ip}^0 \leq l_i^+$), then $q_{ip} \leq l_i^+ - L_{ip}^0$. The initial stock level for the first period ($p = 1$) is given as an input, that is, $L_{i1}^0 = l_{i1}^0$, $\forall i \in V^*$. By the end of each period, once the customers' aggregated demands are known, the initial stock level for the next period can be computed as $L^0_{i(p+1)} = max\{L^0_{ip} + q_{ip} - D_{ip}, 0\}$. Likewise, at this point the holding- or stockout inventory cost at RC, i, and period, p, can be obtained by using Equation (1), where λ represents the unitary cost of holding surplus inventory by the end of a period, and c_{0i} represents the cost of a direct shipment from the depot to RC i (this value is multiplied by 2 in order to account for the return trip to the depot):

$$f(q_{ip}, D_{ip}) = \begin{cases} \lambda\left(L^0_{ip} + q_{ip} - D_{ip}\right) & \text{if surplus} \quad L^0_{ip} + q_{ip} \geq D_{ip} \\ 2 \cdot c_{0i} & \text{if stockout} \quad L^0_{ip} + q_{ip} < D_{ip}. \end{cases} \tag{1}$$

Accordingly, the total inventory cost can be expressed as shown in Equation (2):

$$I(q_{ip}, D_{ip} \mid i \in V, p \in P) = \sum_{p \in P} \sum_{i \in V^*} f(q_{ip}, D_{ip}). \tag{2}$$

For each period $p \in P$, a VRP needs to be solved for those RCs i with $q_{ip} > 0$. As discussed in Toth and Vigo (2014), the VRP can be defined on a complete and undirected graph $G = (V, E)$, where V includes the depot from which n demand points (RCs) are served with a set K of homogeneous vehicles, and E is the set of edges connecting each pair of facilities in V. Each of the vehicles in the fleet has a maximum loading capacity given by $h > 0$. There is a traveling cost, $c_{ij} = c_{ji} > 0$ associated with moving from a facility i to a different facility j ($\forall i, j \in V, i \neq j$). The routing cost at period p depends on the binary decision variables x^{pk}_{ij}, which define whether or not the edge connecting facilities i and j is traversed at period p by a vehicle $k \in K$. Note that this might depend upon the specific values of the customers' demands D_{ip}, that is, $x^{pk}_{ij} = x^{pk}_{ij}(D_{ip})$. Accordingly, the total routing cost across all periods can be expressed as shown in Equation (3):

$$R\left(x^{pk}_{ij}, D_{ip} \mid i, j \in V, p \in P, k \in K\right) = \sum_{p \in P} \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} c_{ij} x^{pk}_{ij}. \tag{3}$$

The objective of simultaneously setting the q_{ip} and x^{pk}_{ij} decision variables in order to minimize the expected overall inventory and routing cost is formulated in Equation (4):

$$E\left[I(q_{ip}, D_{ip}) + R\left(x^{pk}_{ij}, D_{ip}\right)\right] = \sum_{p \in P} \left(\sum_{i \in V^*} E[f(q_{ip}, D_{ip})] + \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} c_{ij} E\left[x^{pk}_{ij}\right] \right). \tag{4}$$

Constraints (5) define the range of each inventory-refill decision variable:

$$0 \leq q_{ip} \leq l^+_i - L^0_{ip} \quad \forall i \in V^*, \forall p \in P. \tag{5}$$

Equation (6) relates the aggregated demand at period p with the initial inventory levels at the next period:

$$L^0_{i(p+1)} = \max\left\{L^0_{ip} + q_{ip} - D_{ip}, 0\right\} \quad \forall i \in V^*, \forall p \in \{1, 2, \ldots, |P| - 1\}. \tag{6}$$

Equations (7) and (8) state that, for each RC, i, $y_{ip} = 1$ if $q_{ip} > 0$ and $y_{ip} = 0$ otherwise (where M is a very large number):

$$q_{ip} \leq l_i^+ y_{ip} \quad \forall i \in V^*, \forall p \in P, \tag{7}$$

$$y_{ip} \leq M q_{ip} \quad \forall i \in V^*, \forall p \in P. \tag{8}$$

Given the set of RCs that are served at period p ($q_{ip} > 0$), the established vehicle routes are stated by setting the variables x_{ij}^{pk}, $\forall i, j \in V, k \in K$. For all $i \in V^*$ and $p \in P$, let y_{ip} be a binary variable that takes the value 1 if RC i has to be serviced at the beginning of period p (i.e., if $q_{ip} > 0$), and takes the value 0 otherwise.

Constraints (9) imply that each vehicle k leaves from, and returns to, the depot exactly once each period:

$$\sum_{i \in V^*} x_{0i}^{pk} = \sum_{i \in V^*} x_{i0}^{pk} = 1 \quad \forall k \in K, \ \forall p \in P. \tag{9}$$

Constraints (10) guarantee that each visited RC $i \in V^*$ is left after the service operation of any time period:

$$\sum_{j \in V \setminus \{i\}} x_{ij}^{pk} + \sum_{j \in V \setminus \{i\}} x_{ji}^{pk} = 2 \cdot y_{ip} \quad \forall i \in V^*, \forall k \in K, \forall p \in P. \tag{10}$$

Equation (11) is subtour elimination constraints (the set of RC of a subtour would violate the corresponding constraint) for each period:

$$\sum_{i \in S} \sum_{j \notin S} x_{ij}^{pk} \leq 2 \sum_{i \in S} y_{ip} \quad \forall S \subset V^*, \forall k \in K, \forall p \in P. \tag{11}$$

Meanwhile, Equation (12) avoid RC supplies exceeding vehicle loading capacities:

$$\sum_{i \in V^*} \sum_{j \in V} x_{ij}^{pk} q_{ip} \leq h \quad \forall k \in K, \forall p \in P. \tag{12}$$

Finally, Constraints (13) and (14) enforce binary conditions on the auxiliary and routing decision variables:

$$y_{ip} \in \{0, 1\} \quad \forall i \in V^*, \forall p \in P. \tag{13}$$

$$x_{ij}^{pk} \in \{0, 1\} \quad \forall i, j \in V, \forall k \in K, \forall p \in P. \tag{14}$$

The multiperiod IRP with stochastic customer demands consists in minimizing the objective function 4 subject to the Constraints (5).

4. Our simulation–optimization approach

A possible solution to the problem described in the previous section will have the form of a matrix with $|V^*|$ rows and $|P|$ columns, where element (i, p) in this matrix will represent the refill policy associated with RC i at period p ($\forall i \in V^*, \forall p \in P$). The approach we propose to solve the multiperiod IRP consists of three different stages. First, a constructive heuristic is employed to generate an initial solution. This initial solution will be a "homogeneous" matrix containing the same value in all its cells, that is: it will propose a unique refill policy that will be systematically applied to all the RCs across the different periods. This strategy will generate a particular expected inventory cost (sum of all expected inventory costs for each RC–period combination) as well as a specific expected routing cost (sum of all expected routing costs for each period). Note that both the inventory cost associated with each RC at the end of period p as well as the routing cost at period $p + 1$ will depend on the precise values of the random demands of each RC at period p (since these values will determine the inventory levels associated with each RC at the end of period p). Second, the constructive heuristic is integrated inside the destruction–construction phase of a VNS framework and then combined with MCS in order to iteratively enhance the initial solution. This procedure is based on the construction of a number of different solution neighborhoods and the subsequent local search phase that explores a neighborhood of the current solution. MCS is employed here to generate realizations of the random demands and then obtain an estimate of both expected inventory and routing costs. Finally, a refinement stage using a higher number of simulation runs is applied to the most "promising" or "elite" solutions obtained in the previous stage in order to obtain a more accurate estimation of the expected cost and select the final solution matrix.

In solving the proposed problem, different metaheuristics could be used, for example, GRASP, Iterated Local Search, Tabu Search, Genetic Algorithms, Simulated Annealing, VNS, etc. However, as discussed in Hansen and Mladenović (2014), VNS approaches are relatively easy-to-implement, do not contain a large number of parameters requiring time-consuming setting processes, and offer an excellent trade-off between simplicity and performance (both in terms of solutions quality as well as in terms of computing times).

4.1. Construction of the initial solution

With the aim of generating an initial solution, a constructive heuristic has been developed. The idea behind this heuristic is to assign a common policy to each RC and period. The policy selected will be the one providing the lowest expected total cost, which will include both expected inventory and routing costs. The heuristic is split in two phases. During the first one, different refill policies are tested, and the associated quantities to serve are estimated together with the expected inventory costs. During the second phase, routing costs are computed for each of these refill policies. In this phase, the quantities to serve generated in the previous phase are used. Finally, the policy providing the lowest total expected cost is implemented at every RC and period.

Algorithm 1 depicts the constructive heuristic in more detail. The input parameters are as follows: the set of RCs, the set of time periods, the initial inventory levels, the maximum storage capacity of each RC, the random demand of each RC at each time period, the set of refill policies, and the

Algorithm 1 Generate Initial Solution

Inputs:
$V = \{0, 1, \ldots, |V|\}$: Set of depot (0) and RCs (V^*)
$P = \{1, 2, \ldots, |P|\}$: Set of time periods
L^0_{i1}: Initial inventory level of RC i at period 1
l^+_i: Maximum storage capacity of RC i
D_{ip}: Customers' aggregated demand at RC i during period p
T: Refill policies as a % of l^+_i (e.g., 0%, 25%, 50%, 75%, 100%)
$maxRuns$: Maximum number of runs in the Monte Carlo simulation
% Phase 1: Compute avg. multi-period inventory costs for each center-policy combination
foreach *refill policy* $t \in T$ **do**
    $expInvCost[t] \leftarrow 0$ % expected inventory cost associated with policy t
end
foreach *RC* $i \in V^*$ **do**
    $accumInvCost \leftarrow 0$
    $iter \leftarrow 0$
    while $iter < maxRuns$ **do**
        foreach *period* $p \in P$ **do**
            $q_{ip}[t][iter] \leftarrow \max\{t \cdot l^+_i - L^0_{ip}, 0\}$
            $d_{ip} \leftarrow$ generate random observation of D_{ip} % Monte Carlo simulation
            $L^0_{i(p+1)} \leftarrow \max\{L^0_{ip} + q_{ip}[t][iter] - d_{ip}, 0\}$
            $invCost \leftarrow computeInventoryCost(t, L^0_{i(p+1)})$
            $accumInvCost \leftarrow accumInvCost + invCost$
            $iter \leftarrow iter + 1$
        end
        $avgInvCostRC \leftarrow accumInvCost/maxRuns$ % avg. inventory cost of RC i under policy t
        $expInvCost[t] \leftarrow expInvCost[t] + avgInvCostRC$
    end
end
% Phase 2: Compute avg. multi-period routing cost and total cost for each policy
$initSol \leftarrow emptySol$
$cost(initSol) \leftarrow \infty$
foreach *refill policy t in T* **do**
    $accumRoutingCost \leftarrow 0$
    $iter \leftarrow 0$
    while $iter < maxRuns$ **do**
        foreach *period* $p \in P$ **do**
            $routingCost \leftarrow estimateRoutingCost(q_{1p}[t][iter], \ldots, q_{|V|p}[t][iter])$ % use savings heuristic
            $accumRoutingCost \leftarrow accumRoutingCost + routingCost$
        end
        $iter \leftarrow iter + 1$
    end
    $expRoutingCost \leftarrow accumRoutingCost/maxRuns$
    $totalCost \leftarrow expInvCost[t] + expRoutingCost$
    if $totalCost < cost(initSol)$ **then**
        $initSol \leftarrow setAllRefillDecisionsToValue(t)$
        $cost(initSol) \leftarrow totalCost$
    end
end
return $initSol$

---

maximum number of simulation runs that must be executed. In our case, the possible refill policies considered are:

- No stock refill, that is, the RC can only count on its current stock level to satisfy the demand of its customers during the next period.
- Refill up to one quarter of total inventory capacity (1/4-refill), that is, if necessary, additional product will be served from the depot to reach that level.
- Refill up to half of total inventory capacity (1/2-refill).
- Refill up to three quarters of total inventory capacity (3/4-refill).
- Refill up to full capacity (full-refill).

Thus, for each policy and RC, a short number of simulation runs is executed (e.g., 30–100 runs) to obtain initial estimates. During each of these runs, the quantity to be served is obtained for each RC–period combination (line 8). This quantity is used in the second phase of the heuristic, and it is computed considering the maximum storage capacity of the RC and its initial inventory level. For each RC and period, the specific value of the random aggregated demand is generated using random sampling (line 9). Hence, it is possible to compute the inventory level at the end of the current period (line 10), which will be the initial inventory level for the next period. This is computed as the sum of the initial inventory level and the quantity served minus the aggregated customer demand. In the case a stockout occurs, then a penalty cost is applied and the final inventory level is set to 0 (it can never be negative at the beginning of a new period). Note that the system evolves considering the dependencies between the realization of the demands at one period and the inventory levels at the beginning of the next one. Finally, the inventory cost is computed (line 11) for each RC and policy. As pointed out already, if a stockout occurs, the cost of a round trip to the depot is charged as part of the inventory cost. Otherwise, the inventory cost is obtained as the number of units in stock multiplied by a λ parameter. In our computational experiments, this parameter has been set to 0.25, as suggested by some authors (Juan et al., 2014). This process (lines 6–13) is repeated until the total number of simulation runs has been reached. Inventory costs are accumulated in each run (line 12), and then average inventory costs are computed for each RC (line 14). The resulting value is added to the total expected inventory cost associated with the current policy (line 15).

At the end of this first phase, the expected inventory costs for each considered replenishment policy is obtained. Also, the computed quantities to serve, for each RC and period, are stored for each simulation run. These quantities are used in the second phase to estimate the expected routing costs associated with each policy. Thus, for each series of delivery quantities the savings heuristic (Clarke and Wright, 1964) is employed to estimate the associated routing cost (line 23). Finally, the expected routing cost is computed (line 26). At the end of this phase, the policy involving the lowest expected total cost (inventory plus routing) is chosen.

4.2. SimVNS stage

During the second stage of our methodology, the classical descendant VNS algorithm is applied (Hansen et al., 2010). Algorithm 2 describes the process. In the first phase, the VNS keep a given number of elite solutions consisting of the best solutions found in trajectory. It starts by assigning the initial solution generated by Algorithm 1 to the current base solution *baseSol* and adding it into a an

Algorithm 2 SimVNS for the periodic IRP

```
   Inputs:
   V = {0, 1, ..., |V|}: Set of depot (0) and RCs (V*)
   P = {1, 2, ..., |P|}: Set of time periods
   maxRuns: Maximum number of simulation runs in the VNS phase
   maxLongRuns: Maximum number of runs in the refinement phase
   k_max: Maximum percentage of policies to reset (defines the number of neighborhoods)
   eliteSetSize: Number of best solutions for the second phase initSol: initial solution
   % Phase 1: VNS with Monte Carlo simulation
1  baseSol ← generate initial solution using Algorithm 1
2  eliteSols ← {baseSol}
3  while stopping criteria not met do
4      k ← 1
5      repeat
6          newSol ← shaking(k, maxRuns, baseSol)
7          newSol ← localSearch(newSol)
8          if number of sols in eliteSols < eliteSetSize then
9              eliteSol ← add(eliteSol, newSol)
           end
10         else
11             if cost(newSol) < cost(worstSol(eliteSols)) then
12                 eliteSols ← update(eliteSols, newSol)
               end
           end
13         if cost(newSol) < cost(baseSol) then
14             baseSol ← newSol
15             k ← 1
           end
16         else
17             k ← k + 1
           end
       until k > k_max
   end
   % Phase 2: Refinement of best solutions
18 bestSol ← baseSol
19 for each sol ∈ eliteSols do
20     sol ← longSimulation(sol, maxLongRuns)
21     if cost(sol) < cost(bestSol) then
22         bestSol ← sol
       end
   end
23 return bestSol
```

empty pool of elite solutions $eliteSols$. Then, it continues with the main VNS loop, which is applied until a predefined stopping criteria is reached. Inside this loop, the VNS modulates the exploitation and exploration by conducting the search to larger and larger neighborhoods when needed, and improving the local optima found by iteratively applying local searches. The movement through these k neighborhoods is carried out using the shaking operator (line 6). Our shaking operator works as follows. The number of policies to be modified from the original policy matrix is given by k, which moves from smallest to largest values. These policies are then reset (destruction process) and redefined (reconstruction process) by using the constructive heuristic designed to obtain the initial solution (the only difference is that now it only affects to a subset of elements in the policy matrix). By increasing the value of k, the changes introduced are larger and larger, allowing the movement through the neighborhoods. After this shaking procedure, the algorithm performs (line

7) the subsequent local search to get a local minimum within the current solution neighborhood. This local search consists in selecting one policy matrix element at random and changing the associated policy for the one providing the best result. This process is iteratively repeated while new improvements in total expected costs are achieved. The problem-specific operators used for the exploitation and exploration of the solution search space are summarized in Table 3. Each time a local minimum is generated, it is included into the set of elite solutions $eliteSol$ if the number of solutions has not reached its size (lines 8 and 9) or it is better than the worst elite solution (lines 11 and 12). As in classical VNS, if the local minimum is better than the base solution, it is the new base solution and the value of k is set again to 1 (lines 13–15). Otherwise, this value is increased by 1 (line 17) until the maximum number k_{max}.

4.3. *Refinement stage*

The aim of this stage is to refine the elite solutions achieved by the VNS algorithm in form of a matrix including the replenishment policy for each RC–period combination.

5. Computational experiments and analysis of results

A set of computational experiments has been carried out to illustrate the utility of our approach for solving the multiperiod IRP with stochastic demands. The set of 27 VRP instances proposed by Augerat et al. (1995) and adapted for the IRP by Juan et al. (2014) are used as a testbed. These instances contain between 27 and 80 RC nodes, a single central depot, and a fleet of 5–10 homogeneous vehicles. The algorithm is implemented as a Java application and executed with the following parameter specifications:

- Inventory holding cost: $\lambda = 0.25$.
- Algorithm stopping criteria: 100 seconds $\times$ number of considered periods.
- No. of simulation runs in Phase 1: 30.
- No. of simulation runs in the refinement phase: 1000.
- Maximum value for the shaking operator k: 40%.

The aggregated customer demands are assumed to follow a log-normal probability distribution with the same average values as the ones proposed in the original instances. Moreover, following Juan et al. (2014) three different variance levels are considered: low (factor = 0.25), medium (factor = 0.50), and large (factor = 0.75). As the main novelty with respect to previous works, three different planning horizons are analyzed: 3, 5, and 7 time periods. The obtained results are reported in Tables 4–6. The number of RCs n and vehicles k in each problem setting are reflected in the instance

Table 3
Shaking and local search operators used in the VNS metaheuristic

Operator	Type	Description
Random policy change	Exploitation	Destroy and repair solution by randomly changing k policy decisions
Guided policy change	Exploration	Destroy and repair solution by changing single policy decision to the best one found

Table 4
Comparison our best solution to initial solution (*Variance* = *0.25*)

Instance	Periods: 3					Periods: 5					Periods: 7				
	Single period (1)	Multiperiod InitSol (2)	Multiperiod OBS (3)	Percentage gap (1)-(3)	Percentage gap (2)-(3)	Single period (4)	Multiperiod InitSol (5)	Multiperiod OBS (6)	Percentage gap (4)-(6)	Percentage gap (5)-(6)	Single period (7)	Multiperiod InitSol (8)	Multiperiod OBS (9)	Percentage gap (7)-(9)	Percentage gap (8)-(9)
A-n32-k5	3968,9	3715,6	3615,1	−8,9	−2,7	6563,1	5872,4	5807,2	−11,5	−1,1	9523,5	8452,6	8069,6	−15,3	−4,5
A-n33-k5	3308,5	3213,5	2966,6	−10,3	−7,7	5573,8	5143,5	4922,7	−11,7	−4,3	8111,7	7217,8	6901,5	−14,9	−4,4
A-n33-k6	3576,5	3349,8	3184,5	−11,0	−4,9	5665,0	5493,7	5207,6	−8,1	−5,2	8111,7	7593,9	7174,4	−11,6	−5,5
A-n37-k5	3059,4	3045,1	2951,2	−3,5	−3,1	5163,6	5010,3	4929,7	−4,5	−1,6	7354,5	7156,7	6867,0	−6,6	−4,0
A-n38-k5	3724,5	3548,0	3399,3	−8,7	−4,2	6287,3	5814,4	5659,7	−10,0	−2,7	8943,4	8216,1	7946,9	−11,1	−3,3
A-n39-k6	4124,9	3852,0	3627,9	−12,0	−5,8	6701,6	6262,3	6004,1	−10,4	−4,1	9421,2	8620,6	8358,1	−11,3	−3,0
A-n45-k6	4694,5	4206,8	4114,3	−12,4	−2,2	7691,9	6935,3	6729,9	−12,5	−3,0	10637,6	9408,4	9329,8	−12,3	−0,8
A-n45-k7	5876,1	5452,9	5251,2	−10,6	−3,7	9665,6	8719,5	8531,7	−11,7	−2,2	13435,4	11968,5	11713,7	−12,8	−2,1
A-n55-k9	5730,4	5244,8	4970,5	−13,3	−5,2	9632,4	8383,7	8194,6	−14,9	−2,3	13134,4	11756,9	11278,7	−14,1	−4,1
A-n60-k9	7102,5	6325,8	6154,8	−13,3	−2,7	11864,4	10262,7	10174,6	−14,2	−0,9	16237,1	14300,9	13855,0	−14,7	−3,1
A-n61-k9	5245,1	4878,1	4621,9	−11,9	−5,3	8776,0	7954,4	7705,5	−12,2	−3,1	12302,1	11026,5	10701,3	−13,0	−2,9
A-n63-k9	8850,0	7913,2	7668,7	−13,3	−3,1	14579,3	12810,8	12479,3	−14,4	−2,6	19657,6	17629,7	17008,7	−13,5	−3,5
A-n65-k9	6388,9	5658,7	5549,5	−13,1	−1,9	10453,7	9304,1	9050,5	−13,4	−2,7	14757,2	12895,0	12581,8	−14,7	−2,4
A-n80-k10	9673,5	8716,3	8322,2	−14,0	−4,5	15563,4	13717,0	13463,4	−13,5	−1,8	19862,5	19268,0	18505,7	−6,8	−4,0
B-n31-k5	3447,5	3344,8	3108,5	−9,8	−7,1	5826,8	5442,8	5098,0	−12,5	−6,3	8164,6	7608,2	7121,1	−12,8	−6,4
B-n35-k5	4766,8	4554,3	4298,3	−9,8	−5,6	7890,5	7361,9	6987,0	−11,5	−5,1	11080,3	9927,0	9801,1	−11,5	−1,3
B-n39-k5	3082,7	2981,9	2910,9	−5,6	−2,4	5131,2	4920,7	4756,0	−7,3	−3,3	7223,2	6903,5	6637,6	−8,1	−3,9
B-n41-k6	4374,7	3942,5	3823,2	−12,6	−3,0	6970,2	6392,6	6208,5	−10,9	−2,9	9863,3	9096,5	8680,4	−12,0	−4,6
B-n45-k5	3882,0	3548,6	3411,1	−12,1	−3,9	6213,1	5755,0	5529,1	−11,0	−3,9	8661,0	7883,8	7761,7	−10,4	−1,5
B-n50-k7	4036,6	3655,6	3538,9	−12,3	−3,2	6860,1	6059,7	5856,7	−14,6	−3,3	9494,1	8354,4	8112,4	−14,6	−2,9
B-n52-k7	3894,4	3825,3	3652,4	−6,2	−4,5	6466,5	6276,2	6001,8	−7,2	−4,4	9064,2	8368,6	8257,8	−8,9	−1,3
B-n56-k7	3464,7	3380,2	3286,7	−5,1	−2,8	5848,2	5606,6	5516,4	−5,7	−1,6	8129,3	8113,3	7580,8	−6,7	−6,6
B-n57-k9	8205,8	7454,9	7174,5	−12,6	−3,8	13288,3	11987,2	11544,4	−13,1	−3,7	16863,3	16566,8	15867,6	−5,9	−4,2
B-n64-k9	4617,3	4339,3	4219,3	−8,6	−2,8	7580,0	7066,6	6889,1	−9,1	−2,5	10893,5	9872,0	9600,7	−11,9	−2,7
B-n67-k10	5793,1	5315,5	5125,7	−11,5	−3,6	9448,3	8581,6	8369,3	−11,4	−2,5	13218,5	12064,7	11535,8	−12,7	−4,4
B-n68-k9	6485,2	6107,6	5824,0	−10,2	−4,6	10720,7	9943,7	9414,4	−12,2	−5,3	15266,7	13490,3	13196,8	−13,6	−2,2
B-n78-k10	6498,5	5881,6	5653,7	−13,0	−3,9	10631,8	9664,1	9207,1	−13,4	−4,7	14778,4	13399,1	12756,9	−13,7	−4,8
Average	*5106,4*	*4720,5*	*4534,2*	*−10,6*	*−4,0*	*8409,5*	*7657,1*	*7416,2*	*−11,2*	*−3,2*	*11636,7*	*10635,5*	*10266,8*	*−11,7*	*−3,5*

Table 5
Comparison our best solution to initial solution (variance = 0.5)

	Periods: 3					Periods: 5					Periods: 7				
Instance	Single period (1)	Multiperiod InitSol (2)	Multiperiod OBS (3)	Percentage gap (1)-(3)	Percentage gap (2)-(3)	Single period (4)	Multiperiod InitSol (5)	Multiperiod OBS (6)	Percentage gap (4)-(6)	Percentage gap (5)-(6)	Single period (7)	Multiperiod InitSol (8)	Multiperiod OBS (9)	Percentage gap (7)-(9)	Percentage gap (8)-(9)
A-n32-k5	3770,6	3715,6	3625,1	−3,9	−2,4	6382,3	6287,8	6001,0	−6,0	−4,6	8924,2	8574,7	8375,4	−6,1	−2,3
A-n33-k5	3323,5	3215,6	3094,7	−6,9	−3,8	5604,6	5236,7	5014,6	−10,5	−4,2	7987,3	7327,7	7132,3	−10,7	−2,7
A-n33-k6	3310,1	3354,9	3222,8	−2,6	−3,9	5696,6	5587,1	5294,9	−7,1	−5,2	8342,3	7697,5	7538,5	−9,6	−2,1
A-n37-k5	3132,8	3115,8	3054,1	−2,5	−2,0	5254,6	5190,7	5043,2	−4,0	−2,8	7734,9	7314,1	7095,4	−8,3	−3,0
A-n38-k5	4035,2	3619,0	3546,9	−12,1	−2,0	6530,2	5967,5	5713,6	−12,5	−4,3	9197,1	8372,6	7947,8	−13,6	−5,1
A-n39-k6	3851,4	3883,7	3721,6	−3,4	−4,2	6413,6	6373,5	6171,8	−3,8	−3,2	9043,4	8859,2	8658,5	−4,3	−2,3
A-n45-k6	4697,0	4365,0	4278,0	−8,9	−2,0	7736,1	7044,2	6887,0	−11,0	−2,2	10867,8	9661,7	9603,9	−11,6	−0,6
A-n45-k7	5986,2	5576,7	5395,4	−9,9	−3,3	9592,5	8972,4	8634,6	−10,0	−3,8	13347,4	12557,8	11959,9	−10,4	−4,8
A-n55-k9	5781,2	5332,5	5172,4	−10,5	−3,0	9247,8	8670,3	8242,8	−10,9	−4,9	12897,0	11844,4	11487,6	−10,9	−3,0
A-n60-k9	7378,0	6550,7	6480,2	−12,2	−1,1	11647,1	10601,7	10276,3	−11,8	−3,1	16347,4	14956,4	14282,6	−12,6	−4,5
A-n61-k9	5262,3	4887,3	4824,9	−8,3	−1,3	8629,0	8126,6	7827,5	−9,3	−3,7	12234,6	11327,5	11010,2	−10,0	−2,8
A-n63-k9	9036,9	8165,7	8008,7	−11,4	−1,9	14446,4	13224,4	12733,0	−11,9	−3,7	20479,5	18163,6	17927,9	−12,5	−1,3
A-n65-k9	6220,4	5793,7	5663,9	−8,9	−2,2	10234,2	9495,2	9199,1	−10,1	−3,1	14493,6	13162,9	12930,1	−10,8	−1,8
A-n80-k10	9747,7	8784,2	8690,6	−10,8	−1,1	15732,4	14362,0	13951,5	−11,3	−2,9	21991,7	19902,1	19343,5	−12,0	−2,8
B-n31-k5	3657,3	3401,7	3265,3	−10,7	−4,0	5766,8	5513,7	5184,1	−10,1	−6,0	8159,0	7507,3	7203,2	−11,7	−4,1
B-n35-k5	4702,7	4683,0	4512,5	−4,0	−3,6	7635,5	7726,2	7274,8	−4,7	−5,8	10686,6	10516,3	10126,0	−5,2	−3,7
B-n39-k5	3435,4	3117,6	3025,5	−11,9	−3,0	5670,9	5189,6	4953,1	−12,7	−4,6	8005,6	7196,9	6882,3	−14,0	−4,4
B-n41-k6	4306,7	4134,7	3910,9	−9,2	−5,4	7090,2	6690,1	6370,8	−10,1	−4,8	9903,1	9375,6	8867,5	−10,5	−5,4
B-n45-k5	3807,5	3566,0	3540,1	−7,0	−0,7	6230,1	5953,3	5746,1	−7,8	−3,5	8818,3	8130,7	8020,3	−9,0	−1,4
B-n50-k7	3972,6	3862,1	3702,8	−6,8	−4,1	6533,1	6188,0	5995,3	−8,2	−3,1	9216,5	8563,6	8425,4	−8,6	−1,6
B-n52-k7	4263,6	3870,0	3795,7	−11,0	−1,9	6987,5	6447,7	6153,0	−11,9	−4,6	9718,9	8851,4	8510,5	−12,4	−3,9
B-n56-k7	3567,5	3526,3	3450,3	−3,3	−2,2	5913,3	6008,6	5651,3	−4,4	−5,9	8271,4	8336,9	7850,8	−5,1	−5,8
B-n57-k9	8299,4	7658,7	7418,5	−10,6	−3,1	13253,9	12408,3	11833,8	−10,7	−4,6	18283,5	17006,8	16444,4	−10,1	−3,3
B-n64-k9	4786,7	4473,3	4384,5	−8,4	−2,0	7582,0	7335,2	7079,8	−6,6	−3,5	11243,3	10213,0	9875,2	−12,2	−3,3
B-n67-k10	5853,4	5424,4	5201,8	−11,1	−4,1	9525,4	8772,9	8431,7	−11,5	−3,9	13397,7	12284,9	11837,1	−11,6	−3,6
B-n68-k9	7063,9	6312,6	6202,8	−12,2	−1,7	11084,4	10400,1	9749,5	−12,0	−6,3	15566,9	13968,2	13664,0	−12,2	−2,2
B-n78-k10	6591,1	6032,1	5930,3	−10,0	−1,7	10613,9	9807,6	9515,7	−10,3	−3,0	14829,2	13728,2	13219,3	−10,9	−3,7
Average	*5179,3*	*4830,5*	*4708,2*	*−8,5*	*−2,7*	*8448,7*	*7910,4*	*7590,0*	*−9,3*	*−4,1*	*11851,4*	*10940,8*	*10600,7*	*−10,3*	*−3,2*

Table 6
Comparison our best solution to initial solution (Variance = 0.75)

	Periods: 3					Periods: 5					Periods: 7				
Instance	Single period (1)	Multiperiod InitSol (2)	Multiperiod OBS (3)	Percentage gap (1)-(3)	Percentage gap (2)-(3)	Single period (4)	Multiperiod InitSol (5)	Multiperiod OBS (6)	Percentage gap (4)-(6)	Percentage gap (5)-(6)	Single period (7)	Multiperiod InitSol (8)	Multiperiod OBS (9)	Percentage gap (7)-(9)	Percentage gap (8)-(9)
A-n32-k5	4049,3	3907,1	3824,3	−5,6	−2,1	6648,4	6352,0	6174,1	−7,1	−2,8	9421,6	8740,5	8602,7	−8,7	−1,6
A-n33-k5	3478,8	3278,6	3096,6	−11,0	−5,6	5811,4	5320,7	5149,9	−11,4	−3,2	8229,1	7427,6	7179,0	−12,8	−3,3
A-n33-k6	3385,6	3416,9	3231,4	−4,6	−5,4	5773,0	5537,4	5360,0	−7,2	−3,2	8331,3	7920,9	7657,8	−8,1	−3,3
A-n37-k5	3607,2	3174,0	3146,4	−12,8	−0,9	5933,0	5284,2	5184,4	−12,6	−1,9	8461,4	7388,1	7357,0	−13,1	−0,4
A-n38-k5	3994,4	3658,0	3580,7	−10,4	−2,1	6538,9	6140,9	5859,5	−10,4	−4,6	9321,5	8444,9	8296,6	−11,0	−1,8
A-n39-k6	4234,1	3905,3	3804,1	−10,2	−2,6	6978,9	6478,2	6243,1	−10,5	−3,6	9978,5	8963,8	8758,4	−12,2	−2,3
A-n45-k6	4823,4	4356,9	4308,5	−10,7	−1,1	8129,2	7214,7	7086,3	−12,8	−1,8	11294,6	10014,3	9884,7	−12,5	−1,3
A-n45-k7	5868,9	5556,5	5408,9	−7,8	−2,7	9653,2	9191,9	8831,0	−8,5	−3,9	13654,2	12721,7	12228,9	−10,4	−3,9
A-n55-k9	5842,7	5402,6	5281,0	−9,6	−2,3	9331,5	8758,3	8328,9	−10,7	−4,9	13321,2	12021,3	11812,5	−11,3	−1,7
A-n60-k9	7348,7	6612,8	6570,7	−10,6	−0,6	12043,3	10783,1	10579,3	−12,2	−1,9	16642,7	15060,4	14662,7	−11,9	−2,6
A-n61-k9	5621,3	4994,6	4921,6	−12,4	−1,5	9174,1	8228,6	7981,5	−13,0	−3,0	12921,1	11458,4	11147,4	−13,7	−2,7
A-n63-k9	8891,2	8277,6	8209,0	−7,7	−0,8	14632,1	13613,7	13342,9	−8,8	−2,0	20627,7	18948,1	18495,0	−10,3	−2,4
A-n65-k9	6445,6	5859,0	5845,6	−9,3	−0,2	10642,5	9735,8	9519,1	−10,6	−2,2	14931,6	13617,0	13294,8	−11,0	−2,4
A-n80-k10	10262,9	9052,6	8982,3	−12,5	−0,8	16682,3	14728,2	14498,9	−13,1	−1,6	23342,1	20895,4	19987,5	−14,4	−4,3
B-n31-k5	3776,9	3481,0	3345,0	−11,4	−3,9	6149,5	5689,5	5355,8	−12,9	−5,9	8822,3	7845,1	7622,9	−13,6	−2,8
B-n35-k5	5000,2	4679,3	4590,3	−8,2	−1,9	8239,4	7782,1	7497,6	−9,0	−3,7	11736,4	11018,1	10400,6	−11,4	−5,6
B-n39-k5	3558,3	3281,0	3127,0	−12,1	−4,7	5881,3	5386,1	5159,1	−12,3	−4,2	8233,0	7537,1	7143,6	−13,2	−5,2
B-n41-k6	4270,6	4174,6	4017,3	−5,9	−3,8	7067,4	6826,6	6516,8	−7,8	−4,5	9825,2	9493,7	9012,1	−8,3	−5,1
B-n45-k5	4170,6	3777,3	3614,0	−13,3	−4,3	6961,4	6024,9	5956,4	−14,4	−1,1	9792,8	8541,8	8364,6	−14,6	−2,1
B-n50-k7	4038,0	3845,5	3752,4	−7,1	−2,4	6627,0	6283,6	6058,3	−8,6	−3,6	9497,8	8722,0	8497,8	−10,5	−2,6
B-n52-k7	4345,3	3962,5	3910,4	−10,0	−1,3	7133,1	6537,4	6334,4	−11,2	−3,1	9854,8	9045,1	8804,2	−10,7	−2,7
B-n56-k7	3756,0	3593,0	3521,9	−6,2	−2,0	6237,1	5951,1	5826,6	−6,6	−2,1	8876,6	8413,0	8086,8	−8,9	−3,9
B-n57-k9	8721,5	7799,2	7646,2	−12,3	−2,0	14217,5	12782,3	12337,0	−13,2	−3,5	19837,2	17874,9	16957,6	−14,5	−5,1
B-n64-k9	5089,9	4546,4	4486,5	−11,9	−1,3	8431,3	7317,5	7291,3	−13,5	−0,4	11735,9	10326,3	10165,3	−13,4	−1,6
B-n67-k10	5924,4	5524,7	5426,9	−8,4	−1,8	9678,7	9013,6	8688,8	−10,2	−3,6	13516,2	12700,3	12084,7	−10,6	−4,8
B-n68-k9	6813,9	6560,4	6286,8	−7,7	−4,2	11481,3	10484,1	10175,9	−11,4	−2,9	15912,5	14891,3	14012,5	−11,9	−5,9
B-n78-k10	6905,5	6226,7	6163,4	−10,7	−1,0	11326,2	10141,9	9890,9	−12,7	−2,5	15725,3	14229,5	13605,3	−13,5	−4,4
Average	*5341,7*	*4922,4*	*4818,5*	*−9,6*	*−2,3*	*8792,7*	*8058,8*	*7823,2*	*−10,8*	*−3,0*	*12364,6*	*11268,9*	*10893,4*	*−11,7*	*−3,2*

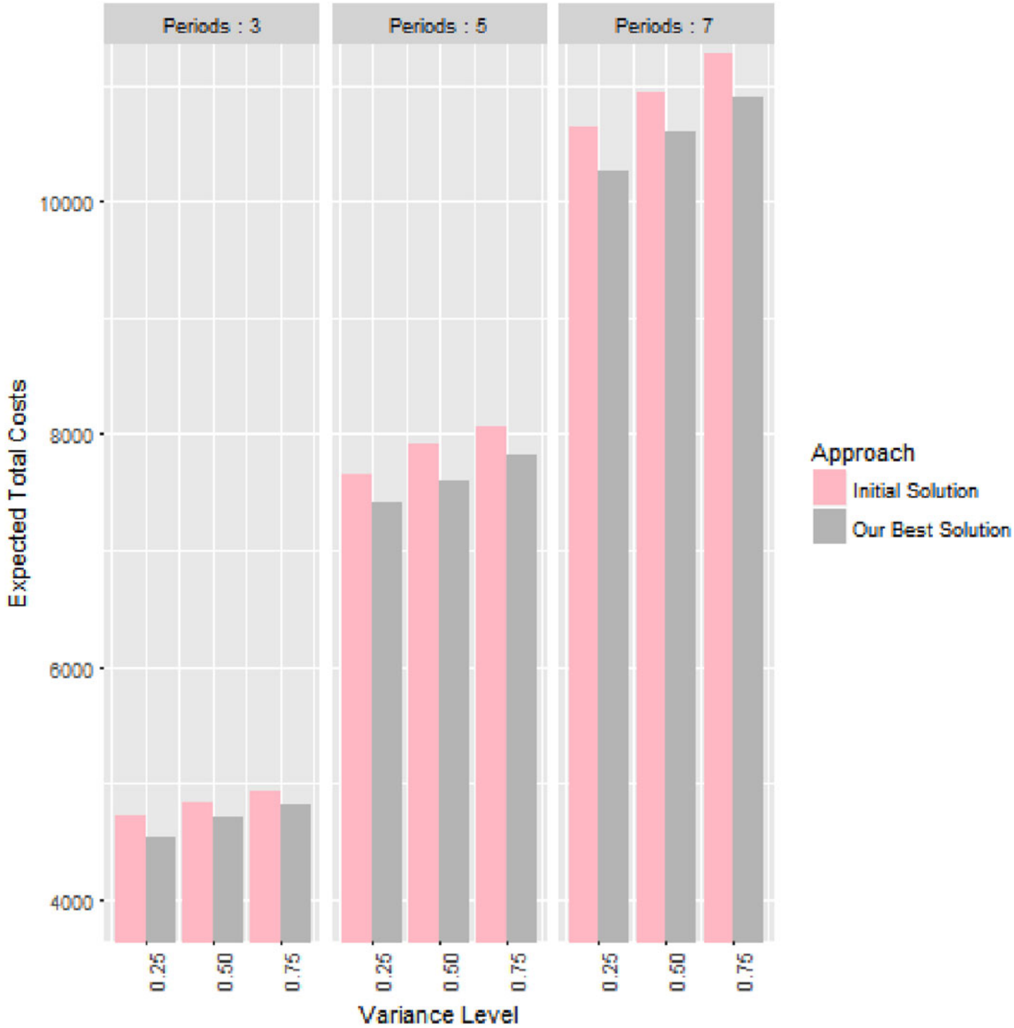

Fig. 2. Expected total costs over all instances for different variance levels and planning horizons.

name. For each considered planning horizon and demand variance level, the results for a single period planning approach (i.e., planning one period at a time) and the multiperiod framework discussed in this paper are outlined. In contrast to our holistic multiperiod framework, the single period approach does not consider future periods by simply minimizing total costs for a single period before updating the initial stock levels for subsequent periods. While this typically leads to low stock levels at the end of each single period, it neglects the increasing transportation costs in subsequent planning periods, leading to significantly higher total costs in comparison to a multiperiod planning approach.

The average total costs over all instances for each variance level and planning horizon of the holistic multiperiod planning framework can be seen in Fig. 2. Cumulated routing and inventory costs are depicted for the initial solution—in which the same replenishment policy is applied at all RCs in each individual period—and our best solution (OBS) found by the simheuristic for the

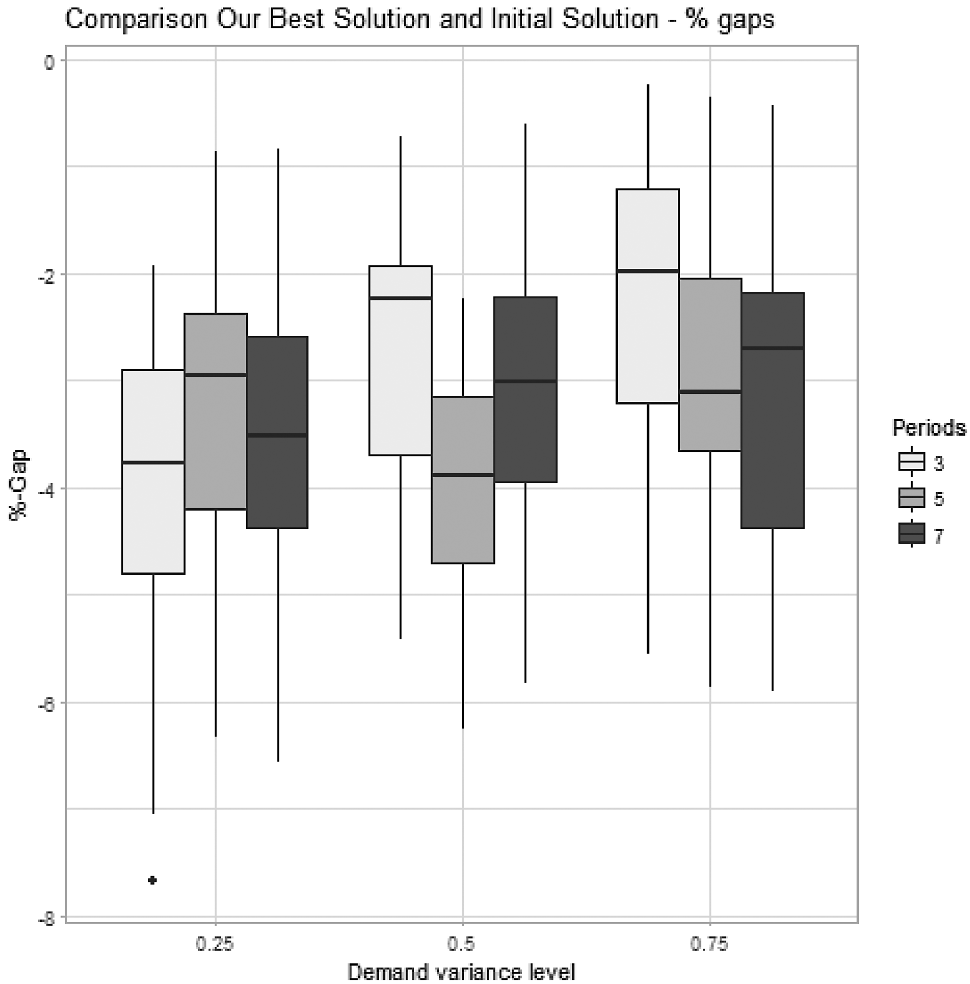

Fig. 3. Boxplot of percentage gaps between initial and best found solution for different planing horizons and demand variance levels.

multiperiod IRP. In both approaches, increasing costs can be observed with higher levels of demand uncertainty, which can be explained with higher inventory (holding or stockout) costs.

Furthermore, it can be seen that the improved inventory decision provided by our algorithm decreases the average total costs in all cases. The same holds for the distribution of the average percentage improvement of the initial solution with OBS for the tested benchmark set, as shown in Fig. 3. The output in form of an individual replenishment policy matrix across all time periods for each RC reaches an average improvement of over 3%. Finally, Fig. 4 shows the expected routing and inventory costs for different replenishment policies and variance levels. For each variance level, higher inventory replenishment levels lead to lower expected inventory (stockout and holding) costs. At the same time, higher replenishment levels lead to an increased proportion of routing costs in all cases.

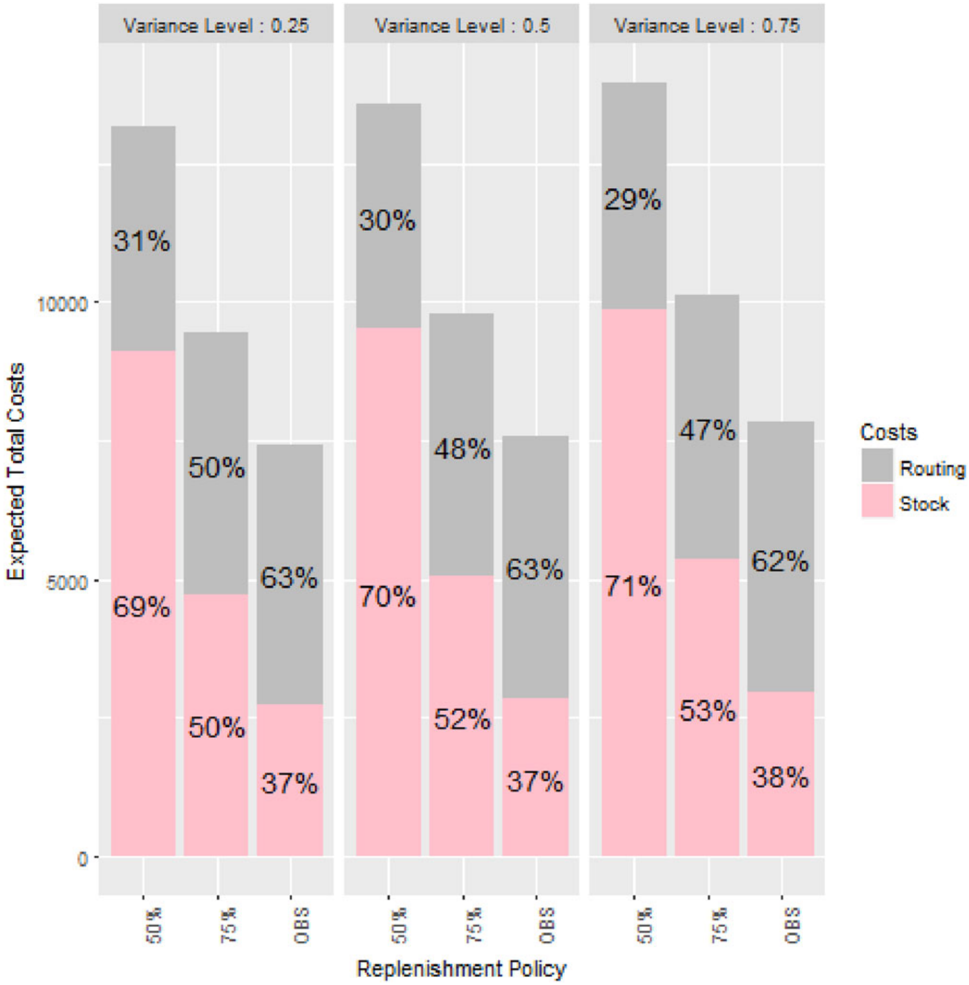

Fig. 4. Expected routing and inventory costs for different replenishment policies and variance levels (five period planning horizon).

6. Conclusions

This paper presents a simheuristic algorithm, based on VNS, to solve the multiperiod IRP with stochastic demands. The integration of MCS techniques into the metaheuristic solving framework allows the consideration of different probability distributions to model the random aggregated demands over various planning horizons. Our approach relies on the selection of the best possible refill policies for each RC and period configuration. Thus, a constructive procedure uses simulation to generate an initial solution, which is iteratively improved by the VNS framework. That way the complexity of the problem—including the interdependencies between consecutive periods due to the random demands—are effectively addressed by our relatively easy-to-implement approach.

Several research lines are possible. On the one hand, our algorithm could be adapted to solve other multiperiod routing problems. On the other hand, the IRP variant considered in this paper could be extended (e.g., by including a heterogeneous vehicle fleet, multiple depots, etc.) in order to address even more realistic IRP settings. Also, simulation techniques could be applied to consider further stochastic input variables, for example, travel times or even distribution costs. Additionally, other metaheuristic approaches for the problem could be implemented in order to develop a comparison with the VNS. Finally, a small-scale problem could be solved using exact methods, although additional assumptions on the probabilities that model customers' demands might be required.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness, FEDER (TRA2013-48180-C3-P, TRA2015-71883-REDT), and the Erasmus+ programme (2016-1-ES01-KA108-023465).

References

Abdelmaguid, T.F., Dessouky, M.M., Ordóñez, F., 2009. Heuristic approaches for the inventory-routing problem with backlogging. *Computers & Industrial Engineering* 56, 4, 1519–1534.

Adelman, D., 2004. A price-directed approach to stochastic inventory/routing. *Operations Research* 52, 4, 499–514.

Aksen, D., Kaya, O., Salman, F.S., Tüncel, Ö., 2014. An adaptive large neighborhood search algorithm for a selective and periodic inventory routing problem. *European Journal of Operational Research* 239, 2, 413–426.

Andersson, H., Hoff, A., Christiansen, M., Hasle, G., Løkketangen, A., 2010. Industrial aspects and literature survey: combined inventory management and routing. *Computers & Operations Research* 37, 9, 1515–1536.

Archetti, C., Bertazzi, L., Hertz, A., Speranza, M.G., 2012. A hybrid heuristic for an inventory routing problem. *INFORMS Journal on Computing* 24, 1, 101–116.

Augerat, P., Belenguer, J.M., Benavent, E., Corberan, A., Naddef, D., Rinaldi, G., 1995. Computational results with a branch-and-cut code for the capacitated vehicle routing problem. Technical Report 949-M, Université Joseph Fourier, Grenoble, France.

Bertazzi, L., Bosco, A., Guerriero, F., Laganà, D., 2013. A stochastic inventory routing problem with stock-out. *Transportation Research Part C: Emerging Technologies* 27, 89–107.

Bertazzi, L., Bosco, A., Laganà, D., 2015. Managing stochastic demand in an inventory routing problem with transportation procurement. *Omega* 56, 112–121.

Caceres-Cruz, J., Arias, P., Guimarans, D., Riera, D., Juan, A.A., 2014. Rich vehicle routing problem: survey. *ACM Computing Surveys* 47, 2, 32:1–32:28.

Chen, Y.M., Lin, C.T., 2009. A coordinated approach to hedge the risks in stochastic inventory-routing problem. *Computers & Industrial Engineering* 56, 3, 1095–1112.

Chitsaz, M., Divsalar, A., Vansteenwegen, P., 2016. A two-phase algorithm for the cyclic inventory routing problem. *European Journal of Operational Research* 254, 2, 410–426.

Clarke, G., Wright, J., 1964. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research* 12, 568–581.

Coelho, L.C., Cordeau, J.F., Laporte, G., 2013. Thirty years of inventory routing. *Transportation Science* 48, 1, 1–19.

Cordeau, J.F., Laganà, D., Musmanno, R., Vocaturo, F., 2015. A decomposition-based heuristic for the multiple-product inventory-routing problem. *Computers & Operations Research* 55, 153–166.

European Commission, 2016. Transport sector economic analysis. Available at https://ec.europa.eu/jrc/en/research-topic/transport-sector-economic-analysis (accessed September 1, 2017).

European Union, 2012. EU transport GHG: routes to 2050 II. The role of GHG emissions from infrastructure construction, vehicle manufacturing, and ELVs in overall transport sector emissions. Available at http://www.eutransportghg2050.eu/cms/assets/Uploads/Reports/EU-Transport-GHG-2050-II-Task-2-FINAL-30Apr12.pdf (accessed January 1, 2017).

Grasas, A., Juan, A.A., Lourenço, H.R., 2016. Simils: a simulation-based extension of the iterated local search metaheuristic for stochastic combinatorial optimization. *Journal of Simulation* 10, 1, 69–77.

Hansen, P., Mladenović, N., 2014. *Variable Neighborhood Search*. Springer US, Boston, MA, pp. 313–337.

Hansen, P., Mladenović, N., Moreno-Pérez, J.A., 2010. Variable neighbourhood search: methods and applications. *Annals of Operations Research* 175, 367–407.

Huang, S.H., Lin, P.C., 2010. A modified ant colony optimization algorithm for multi-item inventory routing problems with demand uncertainty. *Transportation Research Part E: Logistics and Transportation Review* 46, 5, 598–611.

Hvattum, L.M., Løkketangen, A., 2008. Using scenario trees and progressive hedging for stochastic inventory routing problems. *Journal of Heuristics* 15, 6, 527–557.

Hvattum, L.M., Løkketangen, A., Laporte, G., 2009. Scenario tree-based heuristics for stochastic inventory-routing problems. *INFORMS Journal on Computing* 21, 2, 268–285.

Jaillet, P., Bard, J.F., Huang, L., Dror, M., 2002. Delivery cost approximations for inventory routing problems in a rolling horizon framework. *Transportation Science* 36, 3, 292–300.

Juan, A.A., Faulin, J., Grasman, S.E., Rabe, M., Figueira, G., 2015. A review of simheuristics: extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspectives* 2, 62–72.

Juan, A.A., Grasman, S.E., Caceres-Cruz, J., Bektaş, T., 2014. A simheuristic algorithm for the single-period stochastic inventory-routing problem with stock-outs. *Simulation Modelling Practice and Theory* 46, 40–52.

Kleywegt, A.J., Nori, V.S., Savelsbergh, M.W.P., 2004. Dynamic programming approximations for a stochastic inventory routing problem. *Transportation Science* 38, 1, 42–70.

Lenstra, J.K., Kan, A.H.G.R., 1981. Complexity of vehicle routing and scheduling problems. *Networks* 11, 2, 221–227.

Li, K., Chen, B., Sivakumar, A.I., Wu, Y., 2014. An inventory–routing problem with the objective of travel time minimization. *European Journal of Operational Research* 236, 3, 936–945.

Li, M., Wang, Z., Chan, F.T., 2016. A robust inventory routing policy under inventory inaccuracy and replenishment lead-time. *Transportation Research Part E: Logistics and Transportation Review* 91, 290–305.

Liu, S.C., Lee, W.T., 2011. A heuristic method for the inventory routing problem with time windows. *Expert Systems with Applications* 38, 10, 13223–13231.

Ming, Y., Grabot, B., Houé, R., 2014. A typology of the situations of cooperation in supply chains. *Computers & Industrial Engineering* 67, 56–71.

Mjirda, A., Jarboui, B., Macedo, R., Hanafi, S., Mladenović, N., 2014. A two phase variable neighborhood search for the multi-product inventory routing problem. *Computers & Operations Research* 52, 291–299.

Mladenović, N., Hansen, P., 1997. Variable neighborhood search. *Computers & Operations Research* 24, 11, 1097–1100.

Moin, N., Salhi, S., Aziz, N., 2011. An efficient hybrid genetic algorithm for the multi-product multi-period inventory routing problem. *International Journal of Production Economics* 133, 1, 334–343.

Nambirajan, R., Mendoza, A., Pazhani, S., Narendran, T., Ganesh, K., 2016. CARE: Heuristics for two-stage multi-product inventory routing problems with replenishments. *Computers & Industrial Engineering* 97, 41–57.

Niakan, F., Rahimi, M., 2015. A multi-objective healthcare inventory routing problem; a fuzzy possibilistic approach. *Transportation Research Part E: Logistics and Transportation Review* 80, 74–94.

Park, Y.B., Yoo, J.S., Park, H.S., 2016. A genetic algorithm for the vendor-managed inventory routing problem with lost sales. *Expert Systems with Applications* 53, 149–159.

Popović, D., Vidović, M., Radivojević, G., 2012. Variable neighborhood search heuristic for the inventory routing problem in fuel delivery. *Expert Systems with Applications* 39, 18, 13390–13398.

Raa, B., Dullaert, W., 2017. Route and fleet design for cyclic inventory routing. *European Journal of Operational Research* 256, 2, 404–411.

Rahim, M.K.I.A., Zhong, Y., Aghezzaf, E.H., Aouam, T., 2014. Modelling and solving the multiperiod inventory-routing problem with stochastic stationary demand rates. *International Journal of Production Research* 52, 14, 4351–4363.

Ramanathan, U., Gunasekaran, A., 2014. Supply chain collaboration: Impact of success in long-term partnerships. *International Journal of Production Economics* 147, 252–259.

Roldán, R.F., Basagoiti, R., Coelho, L.C., 2016. Robustness of inventory replenishment and customer selection policies for the dynamic and stochastic inventory-routing problem. *Computers & Operations Research* 74, 14–20.

Shaabani, H., Kamalabadi, I.N., 2016. An efficient population-based simulated annealing algorithm for the multi-product multi-retailer perishable inventory routing problem. *Computers & Industrial Engineering* 99, 189–201.

Solyalí, O., Cordeau, J.F., Laporte, G., 2012. Robust inventory routing under demand uncertainty. *Transportation Science* 46, 3, 327–340.

Soysal, M., Bloemhof-Ruwaard, J.M., Haijema, R., van der Vorst, J.G., 2015. Modeling an inventory routing problem for perishable products with environmental considerations and demand uncertainty. *International Journal of Production Economics* 164, 118–133.

Toth, P., Vigo, D., 2014. *Vehicle Routing —Problems, Methods and Applications* (2nd edn). SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, PA.

United Nations, 2011. Statistical yearbook for Asia and the Pacific 2011. Available at http://www.unescap.org/stat/data/syb2011/escap-syb2011.pdf (accessed January 15, 2017).

United States Environmental Protection Agency, 2013. Greenhouse gas emissions 1990–2013. Available at http://www3.epa.gov/otaq/climate/documents/420f15032.pdf (accessed January 15, 2017).

Vansteenwegen, P., Mateo, M., 2014. An iterated local search algorithm for the single-vehicle cyclic inventory routing problem. *European Journal of Operational Research* 237, 3, 802–813.

Yu, Y., Chu, C., Chen, H., Chu, F., 2012. Large scale stochastic inventory routing problems with split delivery and service level constraints. *Annals of Operations Research* 197, 1, 135–158.

Zachariadis, E.E., Tarantilis, C.D., Kiranoudis, C.T., 2009. An integrated local search method for inventory and routing decisions. *Expert Systems with Applications* 36, 7, 10239–10248.