CIRRELT

Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

Robustness of Inventory Replenishment and Customer Selection Policies for the Dynamic and Stochastic Inventory-Routing Problem

Rául F. Roldán
Rosa Basagoiti
Leandro C. Coelho

April 2016

CIRRELT-2016-20

Document de travail également publié par la Faculté des sciences de l'administration de l'Université Laval, sous le numéro FSA-2016-005.

Robustness of Inventory Replenishment and Customer Selection Policies for the Dynamic and Stochastic Inventory-Routing Problem

Rául F. Roldán[1,*], Rosa Basagoiti[1], Leandro C. Coelho[2]

[1] Electronics and Computing Departament, Mondragon University, Loramendi, 4. Apartado 23, 20500 Arrasate – Mondragon, Spain and Compensar Unipanamericana Fundación Universitaria Avenida (Calle) 32 No. 17 - 62 Bogotá D.C., Colombia

[2] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Operations and Decision Systems, 2325 de la Terrasse, Université Laval, Québec, Canada G1V 0A6

Abstract. When inventory management, distribution and routing decisions are determined simultaneously, implementing a vendor-managed inventory strategy, a difficult combinatorial optimization problem must be solved to determine which customers to visit, how much to replenish, and how to route the vehicles around them. This is known as the inventory-routing problem. We analyze a distribution system with one depot, one vehicle and many customers under the most commonly used inventory policy, namely the *s,S*, for different values of *s*. In this paper we propose three different customer selection methods: big orders first, lowest storage first and equal quantity discount. Each of these policies will select a different subset of customers to be replenished in each period. The selected customers must then be visited by a vehicle in order to deliver a commodity to satisfy the customers' demands. The system was analyzed using public benchmark instances of different sizes regarding the number of customers involved. We compare the quality and the robustness of our algorithms and detailed computational experiments show that our methods can significantly improve upon existing solutions from the literature.

Keywords. Inventory-routing problem, inventory policies, customer selection, dynamic and stochastic IRP, demand management.

Acknowledgements. This project was partly funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant 2014-05764. This support is greatly acknowledged.

---

* Corresponding author: raul.roldan@alumni.mondragon.edu

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2016

1 Introduction

Supply chain performance, coordination and integration are some key success factors in obtaining competitive advantages [17]. Inventory and distribution management are two main activities towards that integration, and are said to account for more than 60% of the total logistics costs [12]. The integration of inventory and distribution decisions gives rise to the inventory-routing problem (IRP), which has been studied for the past few decades and has received much attention lately [9]. However, most of these studies focus on optimizing a problem for which all information is known a priori, which is often not the case in practice.

The demand information in an IRP can be static when customers demand are known before the planning, or dynamic, which means it is gradually revealed over time [5, 10]. The dynamic and stochastic IRP (DSIRP) aims not at providing a static output, but rather a solution strategy that uses the revealed information, specifying which actions must be taken as time goes by [4]. Recently, Bertazzi et al. [5], Solyalıet al. [23] and Coelho et al. [10] have solved DSIRPs with the goal of minimizing the total inventory, distribution and shortage cost. They considered at least one of the classical inventory policies, i.e., maximum level or order-up-to-level (OU). They tested their algorithms on instances containing several customers and periods.

In what follows, we review some relevant papers that solve problems similar to the one addressed in this paper. Moin et al. [18] consider a deterministic scenario and develop a hybrid genetic algorithm for the multi-product multi-period IRP. The algorithm is designed to work in two steps: the first to pickup commodities from a supplier, and the second to plan visits and deliveries to the customers. Salhi et al. [21] solved a multi-depot IRP using a giant tour for each depot which was later improved by local search. This algorithm was compared to a truncated execution of CPLEX which provided bounds for the solution. Bertazzi et al. [5] have formulated the SIRP as a dynamic program and have solved it by means of a heuristic rollout algorithm, sampling unknown demands and solving a series of deterministic instances Finally, Coelho et al. [10] proposed a rolling horizon algorithm to solve the DSIRP. Demand was forecasted based on historical data, an adaptive large neighborhood search algorithm selected customers for replenishments, and a network flow model was solved to determine optimal delivery quantities. These authors allowed the use of transshipments and direct deliveries to avoid stockouts after the realization of the demand. The setting of our problem description follows that of this paper.

The choice of which inventory policy to apply largely influences the cost of the optimized function. Typically, an inventory policy uses three key parameters: when replenish, how much to replenish, and how often the inventory level is reviewed. For the periodic review inventory system, Wensing [24] describes three policies. One is the OU which refers to a (t, S) system. Here, in each period t, the quantity delivered is that to fill the inventory capacity up to S. Other policies are the (t, s, S) and the (t, s, q). In the former, the customer is served if the inventory level is less than s. In the latter, the replenishment level q is flexible but bounded by the storage capacity. The policies should be articulated with strategies for clients selection, because sometimes it is not possible to serve all clients due to vehicle capacities, and in such cases, it is necessary to prioritize some of them.

Several other exact and metaheuristic methods have been used to find feasible solutions for this problem and its variants. Simic and Simic [22] argue that for complex optimization problems such as the IRP, hybrid methods with techniques such as artificial neural networks, genetic algorithms, tabu search, simulated annealing and evolutionary algorithms can be successfully applied. Some of the techniques used to solve IRPs are summarized next. Genetic algorithms have been employed by Christiansen et al. [6] and Liu and Lee [15], who clustered customers in geographical areas to serve them together. Local search operators were explored by Javid and Azad [13] and Qin et al. [19], who changed the delivery schedule for customers and adjusted the quantities delivered accordingly. Li et al. [14], Liu and Lin [16] and Sajjadi and Cheraghi [20] used simulated annealing to integrate location decisions into the IRP. Adaptive large neighborhood search [8] and a hybrid of mathematical programming and local search [3] have also been used.. Finally, exact methods relying on branch-and-cut [2, 7] and branch-cut-and-price [11] have also been developed.

In this paper we study a DSIRP in which decisions must be made without future information about the demand, which is gradually revealed over time. In this situation, we developed adaptive policies in order to select customers to be replenished in each period. We propose a new three-step solution algorithm, which is flexible enough to consider several different inventory replenishment policies. We are then able to evaluate and compare the performance of our policies on demand satisfaction, average inventory kept at the customers' site, transportation cost, and total cost. Moreover, we show the effect of integrating tactical and operational decisions into the same solution algorithm. We compare the performance of our algorithm on benchmark instances available in the literature, and our results show that the right combination of inventory replenishment policies and customer selection can yield significant savings over the best-known solutions from a competing algorithm.

The remainder of this paper is organized as follows. In Section 2 we formally describe the problem. In Section 3 we present our solution procedure which includes customer selection, quantities determination, and vehicle routing. In Section 4, we present the results of extensive computational experiments and we analyze the trade-off between inventory and transportation costs. We describe how we can identify dominated solutions under a multi-objective optimization approach, and we compare our solutions against the ones from the literature. In Section 5 we present our findings and conclusions.

2 Problem description

The IRP under study consists of one supplier and several retailers as depicted in Figure 1. We assume that the supplier has enough inventory to satisfy the demand of its customers. Customers demand are gradually revealed over time, thus it is said to be dynamic and unknown to the decision maker at the time all decisions are made. The problem is defined over several periods, typically days, and without loss of generality we assume the demand becomes known at the end of the period. This demand can encompass a set of products organized in a pallet, and we will then treat a single commodity as it is done in other

IRPs. The supplier has a single capacitated vehicle to distribute the products and to satisfy the final demand of the customers.

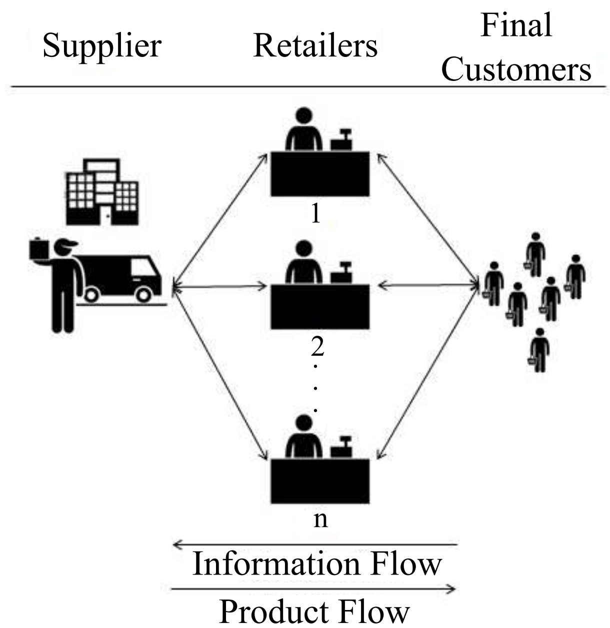


Figure 1: A typical IRP instance with one supplier, n retailers, and a set of final customers representing the demand of the retailers

The IRP is defined on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V} = \{0, \ldots, n\}$ is the vertex set and $\mathcal{A} = \{(i, j) : i, j \in \mathcal{V}, i \neq j\}$ is the arc set. Vertex 0 represents the supplier and the remainder vertices of $\mathcal{V}'$ represent n retailers. The problem is defined over a finite time horizon $\mathcal{H} = \{1, \ldots, p\}$.

The costs incurred is the total of inventory and transportation costs. Inventory costs include the inventory holding and shortage penalties. A transportation cost is paid for each arc traversed by the vehicle. The transportation cost is based on a symmetric distance matrix.

Let n represent the number of customers, each with an initial inventory I_i^0, and let the demand of customer i in period t be d_i^t. Each customer has a maximum inventory capacity C_i, and incurs a unit holding cost h_i per period. Shortages are penalized with z per unit per period.

A single vehicle with capacity Q is available at the depot. The depot has an initial inventory I_0^0, and inventories incur a unit holding cost h_0. A symmetric transportation cost c_{ij} is known. We denote by I_0^t the inventory level at the depot in period t, I_i^t the inventory level at customer i at the end of period t, and l_i^t its lost demand. Let q_i^t be the quantity of products delivered to customer i in period t. At the end of each period t, the inventory level I_i^t for each customer i is updated based on its demand d_i^t, its lost sales l_i^t, the inventory level at previous period I_i^{t-1}, and the quantity q_i^t delivered to it.

A solution to the problem determines the periods in which each customer must be visited, how much to deliver to each of them, and how to create vehicle routes that start at the supplier visit all customers selected to receive a delivery in the period, and return to the depot.

3 Solution algorithm

Our algorithm works by decomposing the problem into smaller parts and by solving them using specialized algorithms. The first part of our solution methodology determines which customers to be visited in each period. This can be done in different ways depending on which inventory replenishment strategy is used. We describe the details of this algorithm in Section 3.1. The second part of the solution algorithm determines how much to deliver to each customer in each period. At this phase, the selection of customers is already done, and one must then respect the capacity of the vehicle. The details on how we determine delivery quantities are described in Section 3.2. The third and last part of our algorithm is to create vehicle routes. This problem can be solved by different algorithms. Here, we use a specialized exact algorithm. It is briefly described in Section 3.3. A flowchart of our solution method to the problem is depicted in Figure 2.

3.1 Selecting customers to replenish

The selection of customers to replenish on a given period depends on the inventory policy used. In what follows we enumerate several different policies organized in four groups in

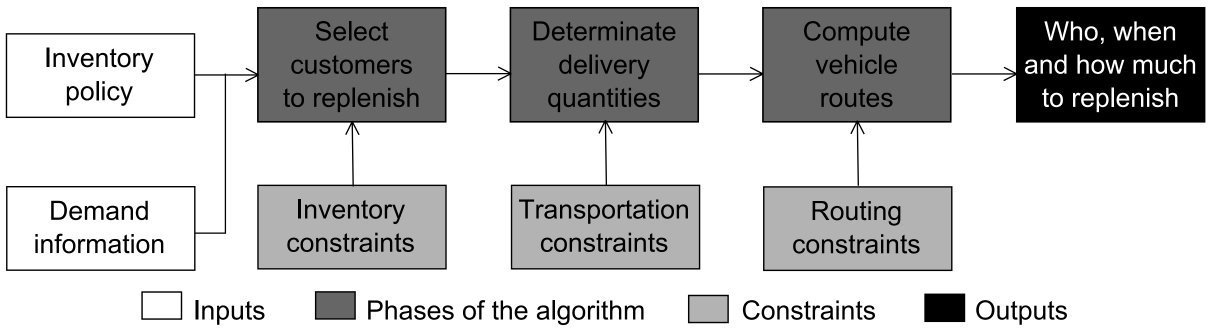


Figure 2: Overview of the main parts of our solution algorithm

Table 1. They are described next.

1. **Fixed quantity policy**: in this policy, the customer always receives a fixed quantity. The fixed quantity for each customer is defined as a fraction θ of its maximum inventory level, i.e., their inventory capacity. In our experiments, we have set $\theta = \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. We note that for $\theta = 0.0$, nothing is shipped and in case of $\theta = 1.0$ this policy coincides with the order-up-to (OU) one. Anything in between yields a maximum level (ML) policy.

2. **OU policy**: the decision maker enforces an OU policy, meaning that whenever a customer is visited, the quantity delivered is that to fill its inventory capacity.

3. **Look ahead**: the decision maker knows a one-step ahead demand. In this case, the delivery quantity is equal to the forthcoming demand.

4. **(s, S)**: the decision maker implements an (s, S) inventory policy. The value of S is set as the inventory capacity, and the parameter s is used to determine when to replenish. This (s, S) policy consists in ordering a variable quantity equal to the difference between S and the current inventory position I_i^t as soon as the inventory level is less than s. The parameter s can be set in different ways as follows:

   (a) The parameter s is determined for each retailer as one fraction α of the inventory capacity, where $\alpha = \{0.25, 0.50, 0.75\}$.

   (b) The values of the parameter s are computed for each retailer using the mean over its historical data.

(c) The value of the parameter s is calculated for each retailer using the mean plus a safety stock, computed as $s = \mu_H + z_\beta \sigma_H$, where β is the probability of a stock-out and z_β is the order quantile of the demand distribution. Here, $1 - \beta$ usually refers to the service level.

(d) The value of the parameter s is equal to the one-step ahead demand.

Table 1: Different possible inventory policies

Group	Variant	Policy	Decision
1	$q_{it} = \theta C_i$	ML and OU	ML, if $I_i^t + q_i < C_i^t$ OU, otherwise
2	$q_{it} = C_i^t - I_i^t$	OU	OU, if $C_i^t - I_i > 0$ 0, otherwise
3	$q_{it} = D_i^t$	ML and OU	ML, if $I_i^t + Q_i < C_i^t$ OU, otherwise
4	$s_i = \alpha S_i$	(s, S)	$S_i - I_i^t$, if $I_i^t < s_i$
	$s_i = \mu_{H_i}$	(s, S)	0, otherwise
	$s_i = \mu_{H_i} + \sigma_{H_i} z_\beta$	(s, S)	
	$s_i = d_i^t$	(s, S)	

Under policies 1–3 of Table 1, all retailers are set to be visited in every period, and under the policies of group 4 only those whose inventory level is below the reorder point s are selected. In our tests, we have chosen policies 1 and 4a as they are representative of all the possible combinations of parameters and policies.

3.2 Determining delivery quantities

It is possible that after having selected the customers and an inventory policy, the capacity of the vehicle is not sufficient to guarantee that the policy is fully respected. Different strategies can be applied in order to rectify the situation. In this work, three different strategies are studied. Note that in all cases, the only information required is the size of the potential order (determined previously) and eventually the information regarding the capacity of the customers.

1. Big Orders First (BOF): under this strategy we prioritize customers requiring more products.

2. Lowest Storage First (LSF): here, we prioritize customers with the low storage capacity.

3. Equal Quantity Discount (EQD): in this strategy, we subtract the same amount to all orders until all customers can be served.

For the first and second strategies, it is important to notice that the last customer selected will be replenished with the remaining capacity of the vehicle. Note also that the same policy should be applied throughout the whole algorithm, so that all selected customers are subject to the same policy.

3.3 Computing vehicle routes

The remaining step in our algorithm is to create vehicle routes of minimum distance, leaving the supplier, visiting all selected customers in each period, and returning to the supplier. This problem is an instance of the traveling salesman problem (TSP) [1], a classical combinatorial optimization problem. Solutions for the TSP can be obtained by a myriad of heuristic and exact algorithms. One of these, Concorde [1], is a publicly available algorithm for solving TSPs to optimality. We use this algorithm to obtain solutions for the TSPs arising in our solution method.

At this point, our overall algorithm determines the inventory level of each customer, all the incurred costs, and the procedure is repeated for the next period of the planning horizon.

4 Computational experiments

We have implemented our algorithm in Matlab 2009b running under Windows 8.1. All computations were performed on a personal computer equipped with an Intel Core i3-2370M running at 2.40GHz and with 8GB of RAM memory. In Section 4.1 we describe the instances used to evaluate our algorithm and the comparisons described later. In Section 4.2 we describe the results of the multi-objective optimization, followed in Section 4.3 by the computational results of the optimization based on total cost minimization.

4.1 Instances description

We have used the large dataset of instances from Coelho et al. [10]. A brief description of their generation follows, and for details the reader is referred to their paper. The dataset can be downloaded from http://www.leandro-coelho.com/instances/.

The instances follow some standards defined by the deterministic IRP instances of Archetti et al. [2, 3], namely the mean customer demand, initial inventories, vehicle capacity and geographical location of the vertices. Each instance contains 50 past periods of demand information before the future p periods such that it can be used as historical data. The demand follows a normal distribution whose mean is generated as an integer random number following a discrete uniform distribution in the interval [10, 100], and standard deviation as an integer random number following a discrete uniform distribution in the interval [2, 10]. Maximum inventory capacities are a multiple of the average demand, and initial inventories are equal to the maximum capacity minus the average demand. Holding costs are randomly generated from a continuous uniform distribution in the interval [0.02, 0.10], and the shortage penalty cost equals 200 times the holding cost.

For early tests we have used the large instances containing 20 periods, ranging from five to 200 customers, for a total of 10 instances. They are identified as IRP-n-p-i, indicating n customers, and p periods. Each instance was tested under the two proposed inventory policies (with 10 different values for the parameter θ and three values for the parameter α), and for each one of three customer selection strategies.

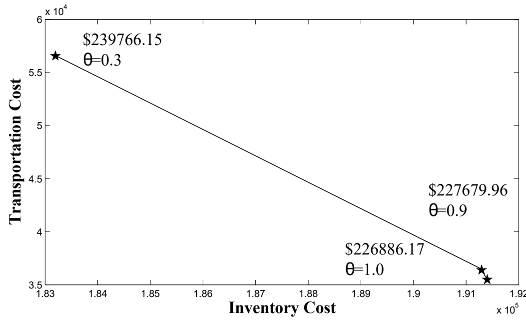
4.2 A multi-objective optimization analysis

The inventory-routing problem is intrinsically bi-objective because the solution is a trade-off between inventory and distribution costs. Pareto optimal solutions, i.e., the set of non-dominated solutions, are those where no improvement in one objective function is possible without deteriorating the other. This information can give insights to decision makers. The trade-off is implemented in the weights used to integrate the costs associated with inventory and transportation. Multi-objective optimization aims at finding the Pareto optimal set.

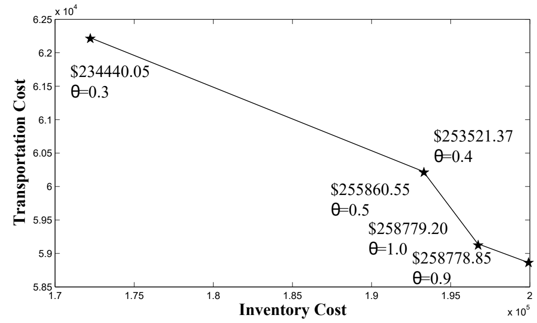
We have solved the instances of the problem using the different methods proposed in this paper. Non-dominated solutions obtained by the procedure were drawn as points in a plane, with the Y axis representing the transportation cost and the X axis representing the inventory costs. In what follows, each figure depicts the Pareto frontier points with annotations for the total average cost, delivery quantity strategy and inventory policy.

In Figure 3 we show the dominant solutions for the *fixed quantity policies*. We observe that we have reduced the vehicle capacity by half, since the original one did not yield any alternative solutions. In Figure 3 we show the dominant solutions under the BOF and LSF delivery strategies. For BOF, three possibilities of replenishment to customers are obtained. The one with $q = 0.3C$ provides a lower inventory cost than those with $q = 0.9C$ and $q = 1.0C$, although the latter yields a lower transportation cost. For the LSF strategy we see five distinct solutions. The EQD policy did not yield different solutions.

In Figure 4 we show the dominant solutions for the *reorder point policies* under the BOF, LSF and EQD delivery strategies. For each delivery policy, three distinct and non-dominated solutions were obtained. The lowest transportation cost was always achieved with $\alpha = 0.25$ at the expense of a very high inventory cost. Alternatively, $\alpha = 0.75$ provided the lowest inventory costs, but very high transportation costs. Note that the average solution provided by Coelho et al. [10] is 69349.97, and hence all three new policies were able to outperform it.

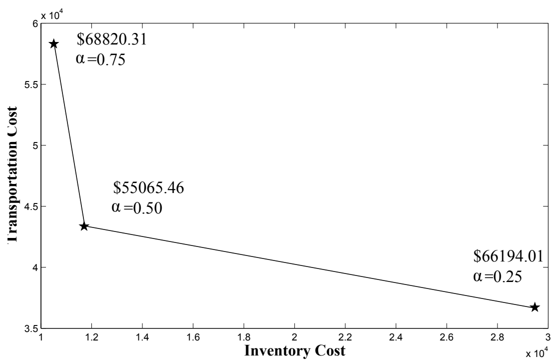


(a) BOF policy

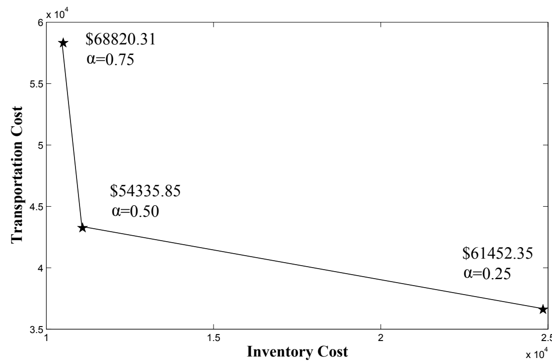


(b) LSF policy

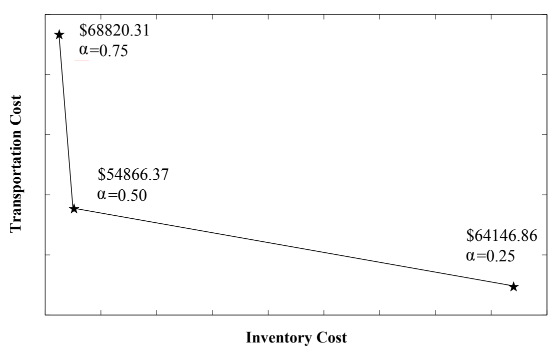
Figure 3: Pareto frontier for the fixed quantity policy with two different customer selection strategies. The EQD strategy did not yield different solutions. Vehicle capacity halved.



(a) BOF Policy



(b) LSF Policy



(c) EQD Policy

Figure 4: Pareto frontier for the reorder point policy with three different customer selection strategies.

4.3 Single objective: total cost minimization

In order to minimize the total cost of inventory and distribution, we have tested the same policies and compared our solutions against those from the literature.

Since this problem allows stockouts, a quick way to find feasible solutions and a benchmark value other than solutions listed in the literature is the case in which the supplier chooses not to replenish, and thus pay the stockout costs. This strategy, also called "wait and see", coincides with policy number 1 with θ=0.0. We show its cost in Table 2. The total cost of the system was separated in its inventory, transportation, and stockout components. Obviously, this policy does not perform well and its costs are significantly higher than those of Coelho et al. [10].

Table 2: Detailed costs for the first policy θ=0.0 compared with those of Coelho et al. [10]

Instance	Inventory holding	Vehicle routing	Stockout	Total cost	Coelho et al. [10]
IRP-5-20	298.37	0.00	45588.00	45886.37	17188.00
IRP-10-20	487.07	0.00	91020.00	91507.07	20182.80
IRP-15-20	842.72	0.00	153808.00	154650.72	33848.20
IRP-25-20	1233.55	0.00	275068.00	276301.55	36455.10
IRP-50-20	2913.80	0.00	506978.00	509891.80	58807.70
IRP-75-20	4025.12	0.00	822502.00	826527.12	77171.90
IRP-100-20	5067.09	0.00	1169380.00	1174447.09	90398.00
IRP-125-20	6870.42	0.00	1349788.00	1356658.42	106242.00
IRP-150-20	7313.46	0.00	1608706.00	1616019.46	114352.00
IRP-200-20	9642.02	0.00	2151808.00	2161450.02	138854.00
Average	3869.36	0.00	817464.60	821333.96	69349.97

The first of our proposed policies rely on the supplier replenishing each retailer with a predetermined quantity, as computed from policy 1 from Table 1. Observe that we have evaluated ten different values for the parameter θ. Under these fixed quantity policies, we note that all strategies of delivery quantities presented in Section 3.2 (BOF, LSF, and EQD) yielded the same transportation costs due to the vehicle capacity never being exceeded. For this reason, the transportation cost is stable throughout the ten values of θ, whereas stockouts costs are drastically reduced, at the expense of a slight increase on inventory holding costs. The reduction of the average total cost from the $0.5C_i$ to $1.0C_i$ policies are very similar and yield the best comparison against the results of Coelho et al. [10]. The difference in these values arises in the average stockout: while in Coelho et al. [10] there is no stockout, in our policies low values are obtained. Coelho et al. [10] used transshipments and direct deliveries to mitigate stockouts after deliveries and demand reealization (similar to a recourse function). Overall, the fixed quantity policy does not outperform the solutions obtained by Coelho et al. [10].

Table 3: Detailed costs for the fixed quantity policy compared those of Coelho et al. [10]. All customer selection strategies yielded the same solution.

Policy	Inventory holding	Vehicle routing	Stockout	Total cost	Coelho et al. [10]
$q = 0.1C_i$	3689.84	63028.93	545634.20	612352.97	69349.97
$q = 0.2C_i$	4616.62	63028.93	275232.80	342878.35	69349.97
$q = 0.3C_i$	8268.17	63028.93	115060.60	186357.69	69349.97
$q = 0.4C_i$	9576.26	63028.93	47539.00	120144.19	69349.97
$q = 0.5C_i$	10712.07	63028.93	1047.80	74788.80	69349.97
$q = 0.6C_i$	10925.41	63028.93	337.20	74291.54	69349.97
$q = 0.7C_i$	10935.15	63028.93	182.80	74146.88	69349.97
$q = 0.8C_i$	10937.22	63028.93	123.00	74089.15	69349.97
$q = 0.9C_i$	10938.00	63028.93	111.60	74078.53	69349.97
$q = 1.0C_i$	10938.34	63028.93	109.20	74076.47	69349.97

The second of our proposed policies is based on replenishments triggered by a reorder point as proposed by item $4a$ of Table 1. The results obtained are presented in Table 4 for the BOF policy, in Table 5 for the LSF policy, and in Table 6 for the EQD policy. Here, we have tested three different values for the parameter α, and the results show that $\alpha = 0.50S$ yields the best solution cost across all three policies. Moreover, all three policies have outperformed the solutions of Coelho et al. [10], with an average total cost reduced by about 20%.

Table 4: Detailed costs for the reorder point policy under the BOF customer selection strategy, compared to those of Coelho et al. [10]

Policy	Inventory holding	Vehicle routing	Stockout	Total cost	Coelho et al. [10]
$s = 0.25S$	6844.48	36718.53	22631.00	66194.01	69349.97
$s = 0.50S$	8391.94	43360.32	3313.20	55065.46	69349.97
$s = 0.75S$	10371.04	58312.87	136.40	68820.31	69349.97

Table 5: Detailed costs for the reorder point policy under the LSF customer selection strategy, compared to those of Coelho et al. [10]

Policy	Inventory holding	Vehicle routing	Stockout	Total cost	Coelho et al. [10]
$s = 0.25S$	6827.28	36601.67	18023.40	61452.35	69349.97
$s = 0.50S$	8388.97	43249.28	2697.60	54335.85	69349.97
$s = 0.75S$	10371.04	58312.87	136.40	68820.31	69349.97

Having identified that the reorder point policies are the best ones proposed in this paper, we have then applied all its variants, comprising three values of the parameter α and three customer selection strategies, to all instances of the dataset of Coelho et al. [10]. Like those authors, we also report our findings by grouping instances into small, medium and large. These are reported in Table 7 and show that our algorithms can always find better solutions than those of Coelho et al. [10]. It also shows that, as previously expected, the policy with $\alpha = 0.50S$ yields the best results. All customer selection methods performed well, but the LSF outperformed the other two by a small margin.

It is relevant to notice that the running times remain low even when the size of the instance increases, unlike the method of Coelho et al. [10]. The difference in our running times between small and large instances is less than one second, which represents approximately

Table 6: Detailed costs for the reorder point policy under the EQD customer selection strategy, compared to those of Coelho et al. [10]

Policy	Inventory holding	Vehicle routing	Stockout	Total cost	Coelho et al. [10]
$s = 0.25S$	6792.95	37341.31	20012.60	64146.85	69349.97
$s = 0.50S$	8406.25	43827.92	2632.20	54866.37	69349.97
$s = 0.75S$	10371.04	58312.87	136.40	68820.31	69349.97

doubling the running time, and never achieving two seconds for the large instances. Those of Coelho et al. [10] increase significantly faster, reaching more than 400 seconds. Finally, one can observe that our algorithm can better manage the trade-off between stockout costs and overall costs. With respect to the competition, our average lost demand is about four times as high, but the overall cost is significantly decreased.

We have performed sensitivity analyses to identify how the algorithms perform and how the solutions change when the distribution capacity is drastically reduced. This experiment is motivated by the fact that for the first policy, the vehicle capacity was not binding. Thus, we have reduced it by 50%. These results are no longer comparable to those of the literature, and a much higher level of lost demand is incurred. The results of these new tests indicate that under the fixed quantity policy, serving big orders first gives significantly better results than prioritizing customers based on their inventory capacities or on decreasing delivery quantities equally among all customers. Moreover, using the reorder point method does not yield better results than the fixed order, despite having some configurations with similar results.

5 Conclusions

In this paper we have solved the Dynamic and Stochastic Inventory-Routing Problem. This problem appears in the literature as that of managing inventory control and distribution simultaneously, minimizing the total inventory holding, transportation, and stock-

Table 7: Summary of computational results for the reorder point policies under the three customer selection strategies, compared to those of Coelho et al. [10] (150 instances)

Policy	Instance	$\alpha = 0.25$			$\alpha = 0.50$			$\alpha = 0.75$			Coelho et al. [10]		
	size	Cost	Avg lost	Time (s)	Cost	Avg lost	Time (s)	Cost	Avg lost	Time (s)	Cost	Avg lost	Time (s)
BOF	small	**14005.52**	19.70	0.62	15259.22	4.63	0.69	18662.84	0.15	0.70	14974.17	0.62	0.00
	medium	38963.21	15.87	0.81	**36681.73**	3.58	0.88	43137.95	0.06	1.16	39546.01	0.41	4.30
	large	64830.24	15.53	1.01	**56425.41**	3.63	1.22	64539.19	0.06	1.88	64854.75	0.46	408.50
Average		36740.24	17.30	0.79	**34035.83**	4.02	0.91	39768.28	0.10	1.19	39791.64	0.50	137.60
LSF	small	**14085.09**	13.81	0.60	14997.94	1.80	0.70	18796.17	0.07	0.65	14974.17	0.62	0.00
	medium	38819.17	13.11	0.81	**36338.41**	2.13	0.90	43169.84	0.06	1.15	39546.01	0.41	4.30
	large	63197.06	12.83	1.03	**55038.15**	2.15	1.23	64540.46	0.06	1.86	64854.75	0.46	408.50
Average		36238.90	13.30	0.79	**33412.14**	2.00	0.92	39831.55	0.07	1.16	39791.64	0.50	137.60
EQD	small	**15006.36**	15.05	0.61	15370.60	2.02	0.69	18819.59	0.10	0.63	14974.17	0.62	0.00
	medium	40513.46	14.75	0.81	**36573.57**	2.02	0.89	43170.08	0.06	1.14	39546.01	0.41	4.30
	large	66260.57	14.39	1.03	**55457.50**	2.13	1.27	64540.86	0.06	1.87	64854.75	0.46	408.50
Average		38034.75	14.76	0.79	**33757.56**	2.05	0.92	39841.12	0.08	1.16	39791.64	0.50	137.60

out costs. Customers demands are revealed dynamically over time, thus one must derive a policy to serve customers accordingly. We have proposed several policies and tested different configurations of the fixed quantity and of the reorder point policies. If the vehicle capacity is not sufficient, we have created three strategies to prioritize some customers. We have tested our policies on a large dataset containing up to 20 periods and 200 customers, and our results significantly improve upon those available in the literature.

Acknowledgments

This project was partly funded by the Canadian Natural Science and Engineering Council (NSERC) under grant 2014-05764. This suport is greatly acknowledged. We thank an associate editor and two anonymous referees for their comments on an earlier version of this paper.

References

[1] D. L. Applegate, R. E. Bixby, V. Chvátal, and W. J. Cook. *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, Princeton, NJ, 2011.

[2] C. Archetti, L. Bertazzi, G. Laporte, and M. G. Speranza. A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation Science*, 41 (3):382–391, 2007.

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

[4] G. Berbeglia, J.-F. Cordeau, and G. Laporte. Dynamic pickup and delivery problems. *European Journal of Operational Research*, 202(1):8–15, 2010.

[5] L. Bertazzi, A. Bosco, F. Guerriero, and D. Laganà. A stochastic inventory routing problem with stock-out. *Transportation Research Part C: Emerging Technologies*, 27 (1):89–107, 2013.

[6] M. Christiansen, K. Fagerholt, T. Flatberg, O. Haugen, O. Kloster, and E. H. Lund. Maritime inventory routing with multiple products: A case study from the cement industry. *European Journal of Operational Research*, 208(1):86–94, 2011.

[7] L. C. Coelho and G. Laporte. A branch-and-cut algorithm for the multi-product multi-vehicle inventory-routing problem. *International Journal of Production Research*, 51:23–24, 2013.

[8] L. C. Coelho, J.-F. Cordeau, and G. Laporte. The inventory-routing problem with transshipment. *Computers & Operations Research*, 39(11):2537–2548, 2012.

[9] L. C. Coelho, J.-F. Cordeau, and G. Laporte. Thirty years of inventory-routing. *Transportation Science*, 48(1):1–19, 2014.

[10] L. C. Coelho, J.-F. Cordeau, and G. Laporte. Heuristics for dynamic and stochastic inventory-routing. *Computers & Operations Research*, 52(Part A):55–67, 2014.

[11] G. Desaulniers, J. G. Rakke, and L. C. Coelho. A branch-price-and-cut algorithm for the inventory-routing problem. *Transportation Science*, forthcoming, 2015.

[12] J. L. Guasch. *Logistic Costs in Latin America and Caribbean.* World Bank, Washington, DC, 2008.

[13] A. A. Javid and N. Azad. Incorporating location, routing and inventory decisions in supply chain network design. *Transportation Research Part E: Logistics and Transportation Review*, 46(5):582–597, 2010.

[14] Y. Li, H. Guo, L. Wang, and J. Fu. A hybrid genetic-simulated annealing algorithm for the location-inventory-routing problem considering returns under e-supply chain environment. *The Scientific World Journal*, 2013:1–10, 2013.

[15] S. C. Liu and W.-T. Lee. A heuristic method for the inventory routing problem with time windows. *Expert Systems with Applications*, 38(10):13223–13231, 2011.

[16] S. C. Liu and C.nC. Lin. A heuristic method for the combined location routing and inventory problem. *International Journal of Advanced Manufacturing Technology*, 26 (4):372–381, 2005.

[17] N. H. Moin and S. Salhi. Inventory routing problems: A logistical overview. *Journal of the Operational Research Society*, 58(9):1185–1194, 2007.

[18] N. H. Moin, S. Salhi, and N. A. B. Aziz. An efficient hybrid genetic algorithm for the multi-product multi-period inventory routing problem. *International Journal of Production Economics*, 133(1):334–343, 2011.

[19] L. Qin, L. Miao, Q. Ruan, and Y. Zhang. A local search method for periodic inventory routing problem. *Expert Systems with Applications*, 41(2):765–778, 2014.

[20] S. R. Sajjadi and S. H. Cheraghi. Multi-products location-routing problem integrated with inventory under stochastic demand. *International Journal of Industrial and Systems Engineering*, 7(4):454–476, 2011.

[21] S. Salhi, A. Imran, and N. A. Wassan. The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation. *Computers & Operations Research*, 52:315–325, 2014.

[22] D. Simic and S. Simic. Evolutionary approach in inventory routing problem. In Ignacio Rojas, Gonzalo Joya, and Joan Cabestany, editors, *Advances in Computational Intelligence*, volume 7903 of *Lecture Notes in Computer Science*, pages 395–403. Springer Berlin Heidelberg, 2013.

[23] O. Solyalı, J.-F. Cordeau, and G. Laporte. Robust inventory routing under demand uncertainty. *Transportation Science*, 46(3):327–340, 2012.

[24] T. Wensing. *Periodic Review Inventory Systems*, volume 651 of *Lecture Notes in Economics and Mathematical Systems*. Springer, Ingolstadt, Germany, 2011.