An exact algorithm for the inventory routing problem with logistic ratio

Claudia Archetti, Leandro C. Coelho, M. Grazia Speranza

An exact algorithm for the inventory routing problem with logistic ratio

Claudia Archetti[a,*], Leandro C. Coelho[b], M. Grazia Speranza[c]

[a] *Department of Information Systems, Decision Sciences and Statistics, ESSEC Business School, Paris, France*
[b] *Canada Research Chair in Integrated Logistics, CIRRELT, Université Laval, Canada*
[c] *Department of Economics and Management, University of Brescia, Brescia, Italy*

Abstract

The Inventory Routing Problem with Logistic Ratio (IRPLR) is a variant of the classical IRP where, instead of the total distribution cost, the ratio between the total distribution cost and the total delivered quantity is minimized, giving rise to a fractional objective function. An exact algorithm is known, solving instances with up to 15 customers. We propose an iterative exact algorithm where, at each iteration, an IRP with a linear objective function is solved. Experiments show that the proposed algorithm is faster when the number of vehicles is small, solving instances with more customers and a longer planning horizon.

Keywords: Inventory routing problem, logistic ratio, exact method.

1. Introduction

In this paper the Inventory Routing Problem with Logistic Ratio (IRP-LR) is studied. The IRP is the problem of determining the minimum cost distribution plan of a commodity from a supplier to a set of geographically dispersed customers over a finite planning horizon. The planning horizon is discretized in periods and, for the ease of explanation and without loss of generalization, we consider periods as days.

*Corresponding author
Email addresses: archetti@essec.edu (Claudia Archetti), leandro.coelho@cirrelt.ca (Leandro C. Coelho), grazia.speranza@unibs.it (M. Grazia Speranza)

Customers have a daily demand and the distribution is performed through a fleet of capacitated vehicles that start and end their route at the supplier location. No stockout is allowed, and the objective is to design a distribution plan that minimizes the total distribution cost, which is usually the sum of the routing and inventory cost. For an exhaustive literature review on the IRP, we refer the reader to the recent tutorials and surveys [9, 10, 11, 22], and to [3] for an overview of applications of the IRP. The interest in studying the IRP is motivated by the large benefits that can be achieved by integrating distribution and inventory management, as demonstrated in [4]. Different exact solution approaches have been proposed for the optimal solution of the IRP. Most of them are based on branch-and-cut algorithms (see Coelho et al. [19, 20], Coelho and Laporte [17], Adulyasak et al. [1], Archetti et al. [6], Coelho and Laporte [18]). To the best of our knowledge, only one exact solution approach is based on branch-and-price, namely the one proposed in [24]. Surprisingly, the literature on heuristic approaches is scarcer than the one on exact algorithms. Recent contributions and variants of the problem include the works of [7] on a new heuristic, [16] for an algorithm for a problem involving also assembly and production, [27] for an application dealing with liquefied natural gas, and [29, 13] for a multi-depot contributions. Finally, the field of dynamic IRPs consider that some information arrives (or is realized) over time. Most algorithms rely on heuristics such as those of [21, 12, 30, 14, 35, 33]. For more detailed information we refer to the surveys of Ritzinger et al. [32], Roldán et al. [34].

The classical IRP focuses on minimizing the overall cost, composed of routing and inventory holding. Recently, other objectives have been proposed with the aim of improving the performance of the whole chain, or to shape solutions towards a specific goal. Escalona et al. [26] used a different objective function to drive the solution towards different replenishment policies. Rau et al. [31] uses a more comprehensive objective function encompassing inventory holding and handling, besides fixed and variable vehicle costs. This allows for measuring not only inventory management contributions to the total cost, but also to obtain a proxy for gas emissions. Also with a focus on emissions, Cheng et al. [15] extends it to consider the effect of a heterogeneous fleet into the cost and emissions of an IRP. The logistic ratio, as considered in this paper, is another type of objective function which aims to deliver more than the minimum necessary by minimizing the cost per unit delivered.

The IRP-LR, recently studied in Archetti et al. [8], is a variant of the IRP in which, instead of the total distribution cost, the *logistic ratio*, i.e., the ratio between the total distribution cost and the total delivered quantity, is minimized. The logistic ratio can be interpreted as the average cost of delivering one unit of commodity to the customers. As explained in Archetti et al. [8], the motivation for using the logistic ratio as objective function instead of the classical total distribution cost is twofold. On one side, the logistic ratio tends to smooth the 'end-of-horizon' effect that typically affects the solutions obtained through the minimization of the total distribution cost. The end-of-horizon effect appears when solving the problem over a finite planning horizon and is a tendency to let the inventory levels to decrease significantly towards the end of the horizon. This has the disadvantage that the next planning cycle will start with all customers having very low initial inventory levels. In addition, the logistic ratio is popular in companies as a performance measure. This is witnessed by the real application which inspired the ROADEF/EURO Challenge 2016 (ROADEF [2016]) proposed by Air Liquide. Although recently introduced in the literature, the logistic ratio has been used as an objective function in Giroudeau et al. [28] for optimizing bulk distribution of gas, and as an evaluation criterion in Alvarez et al. [2], Darvish et al. [23]. As shown in Archetti et al. [8], the use of the logistic ratio as objective function instead of the classical sum of inventory and distribution costs helps in mitigating the end-of-horizon effect. This motivates our study whose aim is enriching the literature on the IRP-LR. We tackle the problem as defined in Archetti et al. [8]. The problem is extremely difficult to solve and research is needed to acquire knowledge useful for the solution of the problem itself and of related (and more complex) variants.

The IRP-LR is the optimization problem arising in the applications mentioned above (ROADEF [2016], Giroudeau et al. [28]), where simplifying assumptions are made in order to obtain a 'tractable' problem for which solutions can be obtained in a reasonable computing time. In particular, the problem formulation relaxes all side constraints apart inventory and vehicle capacity, and it is assumed to have deterministic information over all values of data. Finally, the objective function is clear and well-defined, i.e., minimizing the unit distribution cost. In the IRP-LR three main decisions have to be taken, namely, the schedule of visits to customers, the quantities delivered at each visit and the routes that the vehicles have to travel when performing the deliveries, with the goal of minimizing the logistic ratio of the distribution plan.

In this paper we focus on the IRP-LR studied in Archetti et al. [8]. Differently from most IRPs studied in the literature, the total cost considered in Archetti et al. [8] consists of the routing cost per unit only, as no inventory cost is accounted for. While generally the objective of the IRPs is to minimize the total cost consisting of the sum of distribution and inventory holding (at the supplier and at the customers), the only costs of interest in this variant are the delivery costs. This is consistent with the case where the unitary inventory cost is equal at all customers and at the supplier. In fact, in this case, the distribution plan does not affect the total inventory cost which is a constant. In Archetti et al. [8], a path-flow formulation is proposed for the IRP-LR and the problem is solved through the Dinkelbach's algorithm (see [25]). This algorithm iteratively solves an IRP where the objective function is the difference between the total routing cost and the quantity delivered multiplied by the logistic ratio found at the previous iteration. Each IRP is solved through the branch-price-and-cut algorithm proposed in Desaulniers et al. [24]. This work differs from what is done in Archetti et al. [8] in the solution approach proposed for the solution of the IRP-LR. In fact, instead of using the Dinkelbach's algorithm as done in Archetti et al. [8], we exploit the structure of the problem to derive an ad hoc exact algorithm that works based on iteratively solving a linear maximization problem and a linear minimization problem. The convergence of the algorithm is guaranteed. It finishes in a finite number of steps with an optimal solution. In particular, the solution approach we propose consists in solving a sequence of IRPs with a linear objective function, where either the routing cost is minimized or the total quantity delivered is maximized. The algorithm stops when an optimality condition is satisfied. Acceleration techniques to speed-up the exact algorithm are also proposed. Computational tests are performed on the instances introduced in Archetti et al. [8]. The results show that the proposed algorithm outperforms the one by Archetti et al. [8] when the number of vehicles is small. In addition, it is able to solve instances with a larger number of customers and a longer planning horizon.

The paper is organized as follows. In Section 2 we provide the problem description and a mathematical formulation. Section 3 describes the exact algorithm and the acceleration techniques. Computational results are presented in Section 4 while conclusions are drawn in Section 5.

2. Problem description and formulation

Let $G = (N, E)$ be a complete undirected graph where $N = \{0\} \cup N'$ is the set of locations (nodes) and E is the set of edges between locations. Node 0 is the depot corresponding to the supplier location and N' is the set of customers. A cost c_{ij} is associated with each edge $< i, j >$. We assume that costs c_{ij} satisfy the triangle inequality. The planning horizon $T = \{1, \ldots, H\}$ is composed of H days. Each customer $i \in N'$ is associated with inventory capacity U_i and a daily demand r_{it},

$i \in N'$, $t \in T$. The quantity produced at the depot at period t is denoted as r_{0t} while I_{i0} is the initial inventory level at location $i \in N$. A fleet K of homogeneous vehicles of capacity Q is available to distribute the goods from the depot to the customers. The IRP-LR is the problem of determining the distribution plan that minimizes the logistic ratio while satisfying all customer demands and not violating any constraint. In particular, the following constraints are considered:

1. Vehicle capacity is not exceeded.
2. No stockout occurs at any node in each time period.
3. The maximum inventory level at customers is not exceeded.

We now propose an arc-flow formulation for the IRP-LR involving the following variables:

- I_{it}: Inventory level at location i at the end of period t;
- y_{ij}^{kt}: Number of times edge $< i, j >$ is traversed by vehicle k in period t;
- q_{it}^{k}: Quantity delivered to customer i by vehicle k in period t;
- z_{it}^{k}: Binary variable equal to 1 if location i is visited by vehicle k in period t.

For a subset of customers $S \subseteq N'$, let $E(S) = \{< i, j >\in E \mid i, j \in S\}$. Using this notation, the IRP-LR is formulated as follows:

$$\min \quad \sum_{k\in K} \sum_{<i,j>\in E} \sum_{t\in T} c_{ij} y_{ij}^{kt} \Big/ \sum_{k\in K} \sum_{i\in N'} \sum_{t\in T} q_{it}^{k} \tag{1a}$$

$$\text{s.t.} \quad I_{0t} = I_{0,t-1} + r_{0t} - \sum_{k\in K} \sum_{i\in N'} q_{it}^{k} \qquad t \in T \tag{1b}$$

$$I_{it} = I_{i,t-1} - r_{it} + \sum_{k\in K} q_{it}^{k} \qquad i \in N',\ t \in T \tag{1c}$$

$$I_{it} \geq 0 \qquad i \in N,\ t \in T \tag{1d}$$

$$\sum_{k\in K} q_{it}^k \leq U_i - I_{it-1} \qquad i \in N',\ t \in T \tag{1e}$$

$$q_{it}^k \leq U_i z_{it}^k \qquad i \in N',\ k \in K,\ t \in T \tag{1f}$$

$$\sum_{i\in N'} q_{it}^k \leq Q z_{0t}^k \qquad k \in K,\ t \in T \tag{1g}$$

$$\sum_{k\in K} z_{it}^k \leq 1 \qquad i \in N',\ t \in T \tag{1h}$$

$$\sum_{j:<i,j>\in E} y_{ij}^{kt} = 2z_{it}^k \qquad i \in N,\ k \in K,\ t \in T \tag{1i}$$

$$\sum_{<i,j>\in E(S)} y_{ij}^{kt} \leq \sum_{i\in S} z_{it}^k - z_{st}^k \quad S \subseteq N',\ s \in S,\ k \in K,\ t \in T \tag{1j}$$

$$z_{it}^k \in \{0,1\} \qquad i \in N,\ k \in K,\ t \in T \tag{1k}$$

$$q_{it}^k \geq 0 \qquad i \in N',\ k \in K,\ t \in T \tag{1l}$$

$$y_{ij}^{kt} \in \{0,1\} \qquad < i,j >\in E,\ k \in K,\ t \in T \tag{1m}$$

$$y_{0j}^{kt} \in \{0,1,2\} \qquad j \in N',\ k \in K,\ t \in T. \tag{1n}$$

The objective function (1a) is the logistic ratio, i.e., the ratio between the total routing cost and the total quantity delivered. Constraints (1b)–(1c) are inventory balance constraints for the supplier and the customers, respectively. In particular, the inventory level at time t at the supplier is equal to the inventory level at time $t-1$, plus the quantity produced at time t, minus the quantity distributed to all customers at time t. Similarly, the inventory level at each customer at time t is given by the inventory level at time $t-1$, minus the demand at time t, plus the quantity received at time t. Constraints (1d) establish that stockout is not permitted at the supplier and at the customers. (1e) are the maximum inventory capacity constraints at customers, i.e., the quantity delivered at time t should be not greater than the difference between the inventory capacity and the inventory level at time $t-1$. (1f) link q and z variables while (1g) are vehicle capacity constraints. Constraints (1h) state that no split delivery is allowed, i.e., each customer can be visited by one vehicle at most in any time period. Constraints (1i)–(1j) are classical routing constraints.

In particular, (1i) establish that exactly two edges incident to each visited node must be traversed in any time period. Subtours are eliminated through (1j) which work as follows: for any subset S of customers, any time period and any vehicle, the number of edges traversed by the vehicle linking two customers belonging to S have to be not greater than the number of customers in S minus one. As the number of subsets S is exponential, subtour elimination constraints (1j) are exponentially many and, consequently, have to be separated dynamically when violated. Finally, (1k)–(1n) define the domain of the variables.

3. Exact algorithm

The main idea of the exact algorithm is to sequentially solve different IRPs with a linear objective function which alternates between the routing cost, to be minimized, and the quantity delivered, to be maximized, until an optimality condition is met. As will be seen, convergence is guaranteed and an optimality condition exists, after which the algorithm stops.

Let us introduce the following definitions and notations.

D_{max}**:** The maximum quantity that can be delivered:

$$D_{max} = \min \left\{ HQ|K|, \sum_{i \in N'} \max \left\{ 0, \sum_{t \in T} r_{it} - I_{i0} \right\} + \sum_{i \in N'} \left(U_i - r_{iH} - \max\{0, I_{i0} - \sum_{t \in T} r_{it}\} \right) \right\}. \tag{2}$$

D_{max} corresponds to the maximum quantity that can be delivered by all vehicles over the entire planning horizon. It is the minimum of two terms. The first corresponds to the total vehicle capacity ($HQ|K|$). The second term is the sum, over all customers, of the maximum quantity that can be delivered to each customer on the basis of the demands and inventory capacity. This quantity is given by the sum of two terms:

the difference between the total demand over the planning horizon and the initial inventory level ($\max\{0, \sum_{t \in T} r_{it} - I_{i0}\}$) and the maximum quantity that can be delivered in the last day, i.e., day H ($U_i - r_{iH} - \max\{0, I_{i0} - \sum_{t \in T} r_{it}\}$).

FR_{IRP}**:** Feasible region of the IRP defined by inequalities (1b)-(1n).

$z(R)$**:** Total routing cost:

$$\sum_{k \in K} \sum_{<i,j> \in E} \sum_{t \in T} c_{ij} y_{ij}^{kt}. \tag{3}$$

$z(D)$**:** Total quantity delivered:

$$\sum_{k \in K} \sum_{i \in N'} \sum_{t \in T} q_{it}^{k}. \tag{4}$$

$z^*(R)$**:** Minimum total routing cost. It corresponds to the optimal solution over FR_{IRP} when the objective function corresponds to (3) (as detailed in phase 1 below).

$z^*(D)$**:** Maximum total quantity delivered. It corresponds to the optimal solution over FR_{IRP} when the objective function corresponds to (3) (as detailed in phase 3 below).

LB_n**:** Lower bound on the value of the logistic ratio at iteration n.

D_n^{min}**:** Minimum quantity that has to be delivered at iteration n.

r_{best}**:** Current best value of the logistic ratio.

Each iteration n of the algorithm is associated with a minimum quantity that has to be delivered D_n^{min}. This quantity is initially set to 0 and is incremented at every iteration. Then, each iteration consists in first determining the solution with minimum routing cost $z^*(R)$ delivering a quantity which is at least D_n^{min}, and then the solution with routing cost $z^*(R)$ delivering the maximum possible quantity. More in details, the main phases of each iteration of the algorithm are the following:

1. Determine the solution with minimum total routing cost, i.e., minimize $z(R)$, subject to FP_{IRP} and $z(D) \geq D_n^{min}$. If the problem is infeasible the algorithm stops and the best logistic ratio found is the optimum. The infeasibility is caused by the introduction of constraint $z(D) \geq D_n^{min}$, i.e., there exists no solution in FP_{IRP} satisfying this constraint. Otherwise, let $z^*(R)$ be the value of an optimal solution.
2. Update the lower bound LB_n. At each iteration n, $z^*(R)$ is a lower bound on the routing cost of solutions obtained from iteration n on. Thus, a valid lower bound on the logistic ratio for solutions found from iteration n on is $z^*(R)/D_{max}$.
3. As there may be different solutions in FR_{IRP} with a value of $z(R)$ equal to $z^*(R)$, the next step is to search for the solution in FR_{IRP} with a routing cost equal to $z^*(R)$ for which $z(D)$ is maximized. Thus, the problem of maximizing $z(D)$ subject to FR_{IRP} and $z(R) \leq z^*(R)$ is solved. The corresponding logistic ratio is calculated and the best current value of the logistic ratio is possibly updated. In case the solution is lower than or equal to the lower bound, then the algorithm stops: the optimum has been found. Otherwise, D_n^{min} is set to $z^*(D) + 1$ and a new iteration starts. In fact, once $z^*(D)$ is calculated, then the algorithm has identified the best logistic ratio for solutions delivering a quantity lower than or equal to $z^*(D)$. Thus, the search continues in the next iterations on larger quantities. As quantities are integer, the minimum quantity greater than $z^*(D)$ is $z^*(D) + 1$.

Note that the value of $z(R)$ increases at each iteration as the value of D_n^{min} increases. This implies that the value of LB_n increases as well and thus the algorithm converges to optimality in a finite number of steps.

A pseudo-code of the exact algorithm is outlined in Algorithm 1.

The distribution plan with the minimum routing cost delivering at least a quan-

Algorithm 1 An exact algorithm for the IRP-LR

1: $n := 0$

2: $r_{best} := +\infty$

3: $D_0^{min} = 0$

4: **repeat**

5: Solve the problem of minimizing $z(R)$ subject to FR_{IRP} and $z(D) \geq D_n^{min}$

6: **if** the problem is infeasible **then**

7: Stop and return r_{best}

8: **end if**

9: Let $z^*(R)$ be the optimum

10: Solve the problem of maximizing $z(D)$ subject to FR_{IRP} and $z(R) \leq z^*(R)$

11: Let $z^*(D)$ be the optimum

12: **if** $z^*(R)/z^*(D) < r_{best}$ **then**

13: $r_{best} := z^*(R)/z^*(D)$

14: **end if**

15: $n := n + 1$

16: $D_n^{min} = z^*(D) + 1$

17: $LB_n := z^*(R)/D_{max}$

18: **until** $r_{best} \leq LB_n$

19: Return r_{best}

tity equal to D_n^{min} is determined at line 5. Note that, at the first iteration ($n = 0$), this corresponds to determining the solution in FR_{IRP} with minimum routing cost. If the problem is infeasible the algorithm stops (line 7) and r_{best} is the optimum. In fact, r_{best} is the best logistic ratio for a delivered quantity lower than D_n^{min} and no solution exists for a delivered quantity larger than or equal to D_n^{min}. Otherwise, at line 10, the maximum quantity that can be delivered when the routing cost is lower than or equal to $z^*(R)$ is obtained and the value of r_{best} is updated (line 13). If $r_{best} \leq LB_n$, the algorithm stops and r_{best} is the optimum as LB_n is a lower bound on the logistic ratio for solutions delivering a quantity which is at least equal to D_n^{min}.

3.1. Acceleration techniques

Two acceleration techniques to speed-up Algorithm 1 have been implemented. The first is related to the introduction of a new condition on the minimum quantity that has to be delivered. The second technique concerns the introduction of a new stopping condition. We now describe them in detail.

The first acceleration technique works as follows. Once the value of $z^*(R)$ is calculated at line 5, then it is possible to determine the minimum quantity that needs to be delivered in order to improve the value of the logistic ratio r_{best}. This quantity is:

$$\bar{D}_n^{min} = \lceil z^*(R)/r_{best} \rceil + 1. \tag{5}$$

Quantity $\bar{D}_n^{min}$ is determined on the basis of the following observation. Given the minimum routing cost $z^*(R)$, in order to obtain a value of the logistic ratio improving upon the best solution found so far, i.e., r_{best}, a quantity equal to at least $\lceil z^*(R)/r_{best} \rceil + 1$ must be delivered, otherwise no improvement is achieved.

Algorithm 1 is modified as follows. Once the value of $z^*(R)$ is determined at line 5, the algorithm solves again the problem of minimizing $z(R)$ subject to FR_{IRP} and $z(D) \geq \bar{D}_n^{min}$. Then, three situations may occur:

1. The problem is infeasible. In this case, r_{best} is the optimum value of the logistic ratio and the algorithm stops.
2. The routing cost has not increased. In this case, the algorithm goes to line 10 and determines the maximum quantity $z^*(D)$ with a routing cost not greater than $z^*(R)$.
3. The routing cost has increased. In this case, the value of $\bar{D}_n^{min}$ is updated and the algorithm solves again the problem of minimizing $z(R)$ with $z(D) \geq \bar{D}_n^{min}$.

The second acceleration technique consists in introducing a new stopping criterion which is based on determining an upper bound on the routing cost. The stopping criterion is based on the observation that, as the routing cost is increasing with n, as soon as $z^*(R) \geq r_{best} D_{max}$, the algorithm can be interrupted as no logistic ratio better than r_{best} exists.

By incorporating these acceleration techniques in Algorithm 1, we obtain the accelerated exact algorithm described in Algorithm 2.

4. Computational tests

Algorithm 2 has been tested on benchmark instances for the IRP. The branch-and-cut algorithm presented in Coelho and Laporte [17] is used for solving the IRP at lines 5, 14 and 24. Due to the exponential number of subtour elimination constraints (1j), the algorithm relaxes these constraints and introduces them dynamically when violated. Note that any solution algorithm for the IRP can be embedded in our solution approach for the IRP-LR. As a consequence, computational results may vary according to the IRP solution approach used, as commented in Section 4.1. The instances were proposed in Archetti et al. [5] for the single-vehicle case and were adapted to the multiple-vehicle case in Archetti et al. [6], Coelho and Laporte [17] and Desaulniers et al. [24]. The instances have a value of H equal to 3 and 6 and the number of customers is $|N'| = 5\ell$ with $\ell = 1, ..., 10$ when $H = 3$ and $\ell = 1, ..., 6$ when $H = 3$. The number of vehicles $|K|$ varies from 1 to 5. For each instance characteristic (number of customers, number of vehicles), there exist 5 random instances with different locations of customers and supplier. Thus, we have 250 instances with $H = 3$ and 150 instances with $H = 6$. Note that in Archetti et al. [5], Archetti et al. [6], Coelho and Laporte [17] and Desaulniers et al. [24], two classes of instances were generated, with high and low inventory cost. This distinction is not considered in the present work as no inventory cost is accounted for in the IRP-LR.

In Archetti et al. [8], among the benchmark, only the instances with $H = 3$ and with a number of customers up to 15 were considered. In addition, new instances with 5 and 10 customers were tested where all characteristics remained identical with the exception of the value of the horizon which was set to $H = 4$ and $H = 5$.

Algorithm 2 An accelerated exact algorithm for the IRP-LR

1: $n := 0$
2: $r_{best} := +\infty$
3: $D_0^{min} = 0$
4: **repeat**
5: Solve the problem of minimizing $z(R)$ subject to FR_{IRP} and $z(D) \geq D_n^{min}$
6: **if** the problem is infeasible **then**
7: Stop and return r_{best}
8: **end if**
9: Let $z^*(R)$ be the optimum
10: **if** $z^*(R) \geq r_{best} D_{max}$ **then**
11: Stop and return r_{best}
12: **end if**
13: $\bar{D}_n^{min} = \lceil z^*(R)/r_{best} \rceil + 1$
14: Solve the problem of minimizing $z(R)$ subject to FR_{IRP} and $z(D) \geq \bar{D}_n^{min}$.
15: **if** the problem is infeasible **then**
16: Stop and return r_{best}
17: **end if**
18: **if** the optimum of the problem is not greater than $z^*(R)$ **then**
19: Go to line 24
20: **else**
21: Let $z^*(R)$ be the optimum
22: Go to line 13
23: **end if**
24: Solve the problem of maximizing $z(D)$ subject to FR_{IRP} and $z(R) \leq z^*(R)$
25: Let $z^*(D)$ be the optimum
26: **if** $z^*(R)/z^*(D) < r_{best}$ **then**
27: $r_{best} := z^*(R)/z^*(D)$
28: **end if**
29: $n := n + 1$
30: $LB_n := z^*(R)/D_{max}$
31: **until** $r_{best} \geq LB_n$
32: Return r_{best}.

For these latter instances, no result is available for a number of customer greater than 10. In addition, no instances are tested with an horizon larger than 5.

We have tested Algorithm 2 on all instances tested in Archetti et al. [8]. In addition, we have tested instances with a larger number of customers (up to 25 customers) and with a longer horizon (6 days). More exactly, we have tested instances with up to 25 customers when $H = 3$ and 15 customers when $H = 4, 5, 6$.

4.1. Computational results

In the following we compare the results obtained by Algorithm 2, called `ACS` from now on, and the algorithm presented in Archetti et al. [8], called `ADS`. Tests have been run on machines equipped with Intel Xeon X5650 processors running at 2.67GHz and up to 48GB of RAM. The algorithm was coded in C++ and used CPLEX 12.8 as mathematical programming solver. A time limit of 1 hour was imposed on each run. Tests in ADS [8] were conducted on a Linux computer equipped with an Intel Core i7-4770 processor clocked at 3.4 GHz (a single core was used). CPLEX 12.4.0.0 was used as MILP solver. In Archetti et al. [8] the maximum computing time was set to three hours. All tested instances were solved to optimality except one instance with 10 customers, 3 vehicles and 4 periods.

Results are presented in Tables 1–4 for the different values of H. The first two columns report the values of $|K|$ and $|N'|$, respectively. The following columns report average values over the five instances with the corresponding values of K and N'. In particular, for `ACS` we report: the average value of the logistic ratio, the average CPU time (in seconds), the average number of iterations, the number of instances for which a feasible solution has been found, the number of instances solved to optimality and the average percentage optimality gap at termination. Defining as UB and LB the values of the upper and lower bounds, respectively, found by `ACS` at termination, then the percentage optimality gap is calculated as:

$$100\frac{UB - LB}{LB}.$$

Then, apart for the case with $H = 6$ that was not tested in Archetti et al. [8], the last two columns report the average CPU time (in seconds) and the average number of iterations for `ADS`, respectively.

Focusing on the case with $H = 3$ (Table 1), we see that `ACS` is able to solve all instances with up to 25 customers when $|K| = 1$. When increasing the number of vehicles, the number of instances solved to optimality decreases. In particular, all instances with up to 15 customers are solved to optimality when $|K| = 2, 3$, when $|K| = 4$ the maximum size of instances solved to optimality decreases to 10 customers while when $|K| = 5$, 2 instances with 10 customers are not solved. When comparing with the results obtained by `ADS`, we notice that `ACS` performs better than `ADS` for small values of $|K|$, while the opposite is true when the value of $|K|$ increases. In our opinion, this is primarily due to the solution algorithm used to solve each iteration in each algorithm: while `ACS` is a branch-and-cut, `ADS` is a branch-and-price. We note that the benchmark IRP instances that we used in this study are such that, the higher is the number of vehicles, the shorter are the vehicle routes, in terms of number of customers visited, as the smaller is the vehicle capacity with respect to customer demands. As shown in Desaulniers et al. [24] where different exact solution approaches for the IRP are proposed, branch-and-price performs better than branch-and-cut when routes are shorter.

Similar considerations can be applied to the case $H = 4$ (Table 2) where `ACS` solves to optimality all instances with 15 customers and 3 vehicles (`ADS` has been tested on instances with 10 customers at most), all instances with 10 customers and 4 vehicles while 2 instances with 10 customers and 5 vehicles are not solved. For the case $H = 5$, all instances with 15 customers and 3 vehicles are solved, 2

Table 1: Results with $H = 3$

		ACS						ADS [8]	
$\|K\|$	$\|N'\|$	Log. ratio	CPU time	# iter.	# feas.	# opt.	Opt. gap	CPU time	# iter.
1	5	2.54	0.39	4.20	5	5	0.00	93.80	3.20
	10	1.42	5.68	7.20	5	5	0.00	93.80	3.20
	15	1.15	27.24	14.80	5	5	0.00	8022.80	3.40
	20	1.01	131.10	18.80	5	5	0.00		
	25	0.85	504.49	25.60	5	5	0.00		
Average		1.39	133.78	14.12	25	25	0.00	2736.80	3.27
2	5	3.18	1.66	4.80	5	5	0.00	0.20	3.20
	10	1.86	35.83	10.00	5	5	0.00	8.70	3.20
	15	1.41	188.34	14.60	5	5	0.00	422.80	3.20
	20	1.23	2535.02	17.60	5	2	11.43		
	25	1.06	2682.59	10.00	4	2	35.39		
Average		1.78	1088.69	11.40	24	19	9.36	143.90	3.20
3	5	4.19	2.63	6.40	5	5	0.00	0.20	3.20
	10	2.35	245.17	13.80	5	5	0.00	32.40	3.40
	15	1.68	1740.52	19.20	5	5	0.00	1320.60	3.00
	20	1.64	3600.00	7.40	4	0	47.24		
	25	1.36	3600.00	3.20	2	0	74.97		
Average		2.40	1837.66	10.00	21	15	24.44	451.07	3.20
4	5	5.05	3.18	6.20	5	5	0.00	0.12	3.00
	10	2.79	1322.34	14.20	5	5	0.00	10.80	3.60
	15	2.12	3305.81	8.60	5	1	27.13	231.60	3.60
	20	1.85	3600.00	1.40	2	0	79.52		
	25	1.36	3600.00	1.80	1	0	74.66		
Average		3.05	2366.27	6.44	18	11	36.26	80.84	3.40
5	5	5.99	3.15	6.20	5	5	0.00	0.20	3.40
	10	3.26	2168.22	14.40	5	3	8.23	8.32	3.20
	15	2.74	3600.03	2.80	5	0	42.52	1564.00	3.80
	20	2.04	3600.00	1.00	2	0	79.99		
	25	-	3600.00	1.20	0	0	87.98		
Average		3.77	2594.28	5.12	17	8	43.74	524.17	3.47
Global Average		2.35	1604.14	9.42	105	78	22.76	787.36	3.31

instances with 10 customers and 4 vehicles are not solved while only 1 instance with 10 customers and 5 vehicles is solved (Table 3). For the case with $H = 6$ (Table 4) there is no comparison with `ADS`. In this case, we see that the difficulty of the problem increases further. Again, all instances with 15 customers and 2 vehicles are solved, while already with 3 vehicles there are some instances with 10 customers which are not solved to optimality.

Table 2: Results with $H = 4$

		ACS						ADS [8]	
$\|K\|$	$\|N'\|$	Log. ratio	CPU time	# iter.	# feas.	# opt.	Opt. gap	CPU time	# iter.
1	5	2.66	1.20	7.40	5	5	0.00	0.40	3.00
	10	1.43	12.81	9.80	5	5	0.00	2577.60	3.20
	15	1.18	66.31	19.20	5	5	0.00		
Average		1.76	26.77	12.13	15	15	0.00	1289.00	3.10
2	5	3.28	3.37	6.80	5	5	0.00	0.40	3.20
	10	1.87	669.48	19.20	5	5	0.00	101.40	3.00
	15	1.44	2297.09	24.00	5	4	3.41		
Average		2.20	989.98	16.67	15	14	1.14	50.90	3.10
3	5	4.15	8.25	7.80	5	5	0.00	0.64	3.40
	10	2.38	2078.91	13.60	5	4	4.88	76.75	3.25
	15	1.86	3633.62	7.00	5	0	26.14		
Average		2.80	1906.93	9.47	15	9	10.34	34.47	3.33
4	5	5.33	13.33	9.00	5	5	0.00	1.30	3.00
	10	2.98	3210.41	7.00	5	1	22.23	1046.80	3.40
	15	2.28	3600.20	1.40	4	0	45.80		
Average		3.62	2274.65	5.80	14	6	22.68	524.05	3.20
5	5	6.35	11.97	6.80	5	5	0.00	1.96	3.00
	10	3.37	3600.34	4.20	3	0	56.72	630.40	3.80
	15	-	3600.26	1.00	0	0	100.00		
Average		5.23	2404.19	4.00	8	5	52.24	316.18	3.40
Global Average		2.89	1520.50	9.61	67	49	17.28	451.26	3.22

Finally, in Table 5 we provide the number of feasible and optimal solutions found by `ACS` and the number of optimal solutions found by `ADS` over all values of H. `ADS` was tested on 200 instances and it solved all of them to optimality apart one instance with 10 customers, 3 vehicles and $H = 4$. Results are grouped by value of H, value of $|K|$ and value of $|N'|$. When focusing on the second block, the one where values are averaged over $|K|$, we clearly see what was mentioned before, i.e., `ACS` performs better than `ADS` for small values of $|K|$, while the opposite is true. However, we have to notice that `ACS` was able to solve larger instances, in terms of number of

Table 3: Results with $H = 5$

$\lvert K\rvert$	$\lvert N'\rvert$	ACS Log. ratio	ACS CPU time	ACS # iter.	ACS # feas.	ACS # opt.	ACS Opt. gap	ADS [8] CPU time	ADS [8] # iter.
1	5	2.63	1.72	5.60	5	5	0.00	0.68	3.00
	10	1.42	16.65	8.20	5	5	0.00	240.00	3.00
	15	1.16	146.95	20.80	5	5	0.00		
Average		1.74	55.11	11.53	15	15	0.00	120.34	3.00
2	5	3.19	5.42	7.60	5	5	0.00	4.04	3.00
	10	1.86	985.68	14.80	5	4	2.65	791.80	3.40
	15	1.42	3367.27	15.80	5	1	9.73		
Average		2.16	1452.79	12.73	15	10	4.12	397.92	3.20
3	5	4.21	14.36	8.00	5	5	0.00	2.48	3.60
	10	2.37	3600.08	9.80	5	0	15.93	3644.20	3.20
	15	1.83	3600.13	2.20	5	0	26.18		
Average		2.80	2404.86	6.67	15	5	14.04	1823.34	3.40
4	5	5.30	38.01	9.00	5	5	0.00	7.46	3.00
	10	2.78	3600.12	2.80	3	0	53.49	2114.80	3.20
	15	2.12	3600.09	1.00	2	0	71.05		
Average		3.91	2412.74	4.27	10	5	41.51	1061.13	3.10
5	5	6.46	25.46	6.60	5	5	0.00	2.33	3.25
	10	3.84	3600.30	1.00	1	0	85.62	4485.60	3.40
	15	-	3600.63	1.00	0	0	100.00		
Average		6.02	2408.80	2.87	6	5	61.87	2493.03	3.33
Global Average		2.88	1746.86	7.61	61	40	24.31	1152.34	3.20

Table 4: Results with $H = 6$

$\lvert K\rvert$	$\lvert N'\rvert$	ACS Log. ratio	ACS CPU time	ACS # iter.	ACS # feas.	ACS # opt.	ACS Opt. gap
1	5	2.69	3.20	6.40	5	5	0.00
	10	1.46	47.75	12.40	5	5	0.00
	15	1.18	434.39	24.40	5	5	0.00
Average		1.78	161.78	14.40	15	15	0.00
2	5	3.36	18.91	10.00	5	5	0.00
	10	1.89	2890.84	23.20	5	4	2.72
	15	1.51	3600.19	4.80	5	0	19.15
Average		2.25	2169.98	12.67	15	9	7.29
3	5	4.29	82.46	8.40	5	5	0.00
	10	2.30	3600.13	7.60	3	0	51.21
	15	2.06	3600.19	1.20	3	0	56.07
Average		3.14	2427.59	5.73	11	5	35.76
4	5	5.39	387.29	9.20	5	5	0.00
	10	2.79	3600.36	2.40	3	0	53.94
	15	-	3600.55	1.00	0	0	100.00
Average		4.42	2529.40	4.20	8	5	51.31
5	5	6.15	155.68	8.75	4	4	0.00
	10	3.84	3600.55	1.00	1	0	85.62
	15	-	3601.18	1.00	0	0	100.00
Average		5.69	2616.53	3.21	5	4	66.29
Global Average		2.94	1972.47	8.11	54	38	31.67

customers, and instances with a longer planning horizon with respect to `ADS` which did not solve, in 3 hours of computing time, instances with more than 10 customers when $H = 4, 5$ and instances with H larger than 5.

We highlight that these results were obtained on slower machines using only one third of the maximum running time.

Table 5: Results summary

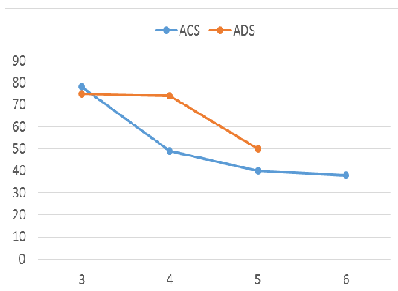
		ACS		ADS
		# feas.	# opt.	# opt.
H	3	105	78	75
	4	67	49	74
	5	61	40	50
	6	54	38	
$\|K\|$	1	70	70	40
	2	69	52	40
	3	62	34	40
	4	50	27	39
	5	36	22	40
$\|N'\|$	5	99	99	75
	10	84	56	74
	15	74	36	50
	20	18	7	
	25	12	7	
Total		287	205	199

For an easier comparison, in Figure 1 we report the number of optimal solutions found by `ACS` and `ADS` classified by the value of H, $|K|$ and $|N'|$. The figure confirms what underlined above: `ADS` works better when the number of vehicle increases, while `ACS` is able to solve instances with larger values of both values of H and $|N'|$.

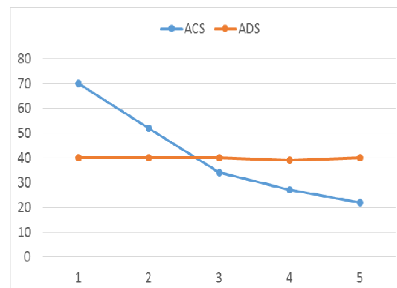
4.2. Sensitivity analysis

In order to obtain detailed sensitivity analysis we have run all instances with 3 and 4 periods, 2 and 3 vehicles, and 10 and 15 customers. For each of these instances, we made six tests: with half and double demand, half and double vehicle capacity, and half and double distances. The corresponding results are detailed in Table 6.

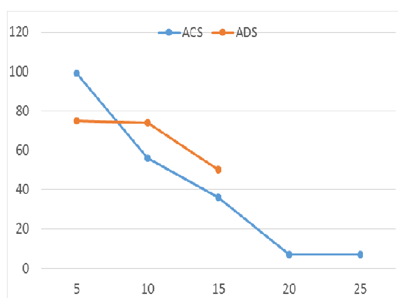
As can be seen from the table, doubling the demand makes all instances infeasible as there is not enough vehicle capacity and initial inventory to satisfy all demands.



a) Number of optimal solutions for different values of H



b) Number of optimal solutions for different values of $|K|$



c) Number of optimal solutions for different values of $|N'|$

Figure 1: Comparison of optimal solutions found by ACS and ADS.

Table 6: Senstivity analysis of several parameters

H	K	n	Instance	LR double demand	Time (s)	LR half demand	Time (s)	LR double veh capacity	Time (s)	LR half veh capacity	Time (s)	LR double distance	Time (s)	LR half distance	Time (s)
3	2	10	1	infeasible	infeasible	1.58	151.47	infeasible	infeasible	2.71	1801.96	3.31	442.30	0.82	410.24
			2	infeasible	infeasible	2.03	42.55	1.68	60.89	3.81	668.51	4.31	99.90	1.07	102.40
			3	infeasible	infeasible	1.86	58.03	1.66	34.36	3.30	118.29	4.23	107.80	1.05	103.62
			4	infeasible	infeasible	2.15	67.41	1.37	18.77	3.28	525.16	3.87	85.64	0.97	83.24
			5	infeasible	infeasible	1.49	51.31	1.19	57.98	2.20	298.23	2.87	83.03	0.71	78.68
		15	1	infeasible	infeasible	1.23	314.61	0.94	115.57	1.85	3595.43	2.27	391.48	0.57	386.79
			2	infeasible	infeasible	1.34	479.75	1.15	363.40	2.20	3600.38	2.71	1436.87	0.67	1449.93
			3	infeasible	infeasible	1.27	796.57	1.16	361.07	2.23	3600.06	2.96	3600.01	0.74	3541.84
			4	infeasible	infeasible	1.55	760.95	1.22	223.53	2.19	1582.71	2.96	3593.22	0.74	880.85
			5	infeasible	infeasible	1.66	1030.64	1.28	311.08	2.67	3600.03	3.19	576.00	0.79	573.32
	3	10	1	infeasible	infeasible	1.92	1196.76	1.39	201.01	3.40	3600.17	4.19	3591.55	1.04	3600.06
			2	infeasible	infeasible	2.57	225.23	1.96	107.71	5.26	3600.28	5.82	1556.81	1.45	1410.55
			3	infeasible	infeasible	2.36	135.11	1.83	63.44	4.38	3594.77	5.19	724.28	1.29	691.99
			4	infeasible	infeasible	2.63	464.80	1.67	52.65	infeasible	infeasible	4.97	1134.11	1.24	1182.41
			5	infeasible	infeasible	1.79	3591.43	1.26	62.54	3.11	3600.01	3.34	351.99	0.83	362.84
		15	1	infeasible	infeasible	1.44	3581.21	1.00	339.99	2.52	3600.55	2.70	3600.07	0.67	3600.09
			2	infeasible	infeasible	1.60	3599.96	1.25	3599.96	-	3600.43	3.21	3600.13	0.80	3600.13
			3	infeasible	infeasible	1.45	3477.15	1.30	823.64	-	3600.01	3.57	3599.51	0.89	3600.01
			4	infeasible	infeasible	1.79	2646.67	1.31	588.73	2.95	3600.03	3.48	3600.05	0.87	3598.85
			5	infeasible	infeasible	1.94	2015.13	1.44	707.06	-	3600.42	3.92	3600.02	0.98	3600.00
4	2	10	1	infeasible	infeasible	1.54	1525.29	1.13	74.99	2.95	3600.09	3.22	3011.34	0.80	2687.36
			2	infeasible	infeasible	1.93	145.43	1.60	45.13	3.79	3600.10	4.34	447.57	1.08	458.97
			3	infeasible	infeasible	1.67	111.27	1.66	110.63	3.32	865.99	4.07	359.28	1.01	350.24
			4	infeasible	infeasible	1.98	231.32	1.50	21.03	3.56	3600.04	4.17	654.85	1.04	665.68
			5	infeasible	infeasible	1.47	216.49	1.18	73.20	2.29	3600.21	2.86	369.13	0.71	369.82
		15	1	infeasible	infeasible	1.12	1088.98	1.04	582.54	2.01	3600.04	2.48	3600.05	0.62	3600.04
			2	infeasible	infeasible	1.30	1433.79	1.10	3600.00	-	3600.07	2.80	3600.03	0.67	3600.04
			3	infeasible	infeasible	1.20	3293.68	1.19	1112.53	2.46	3600.24	2.98	3600.05	0.74	3590.78
			4	infeasible	infeasible	1.51	2191.92	1.24	354.10	2.63	3600.20	3.01	3600.02	0.75	3115.50
			5	infeasible	infeasible	1.50	1920.71	1.24	510.90	-	3600.34	3.26	3600.10	0.81	3600.06
	3	10	1	infeasible	infeasible	1.88	3600.00	1.35	811.42	-	3600.74	4.10	3600.05	1.02	3600.01
			2	infeasible	infeasible	2.46	1250.75	1.89	244.70	-	3600.04	5.67	3600.02	1.40	3600.03
			3	infeasible	infeasible	2.34	640.28	1.86	237.84	4.81	3600.19	5.07	3599.95	1.26	3511.62
			4	infeasible	infeasible	2.47	2269.02	1.81	174.41	-	3600.18	5.55	3600.01	infeasible	infeasible
			5	infeasible	infeasible	1.73	681.01	1.26	137.27	-	3600.03	3.37	3595.89	0.84	3600.04
		15	1	infeasible	infeasible	1.33	3600.08	1.09	1875.49	-	3600.22	3.24	3600.01	0.81	3600.05
			2	infeasible	infeasible	1.56	3600.01	1.28	3600.57	-	3601.06	3.51	3600.02	0.88	3600.11
			3	infeasible	infeasible	1.38	3600.01	1.31	3600.06	-	3600.01	3.97	3600.14	0.98	3600.23
			4	infeasible	infeasible	1.75	3600.02	1.38	1918.20	-	3601.70	3.74	3600.02	0.93	3600.01
			5	infeasible	infeasible	1.93	3600.01	1.40	2886.90	-	3600.85	4.15	3600.05	1.04	3600.08
Average						1.74	1582.17	1.37	770.91	3.04	3104.10	3.72	2365.33	0.91	2236.12

Decreasing vehicle capacity (thus requiring more visits to customers) and doubling distances significantly increase the logistic ratio value. On the other hand, decreasing routing costs (distances) has the most positive impact on improving the logistic ratio. From a computational aspect, doubling the vehicle capacity yields much easier problems, as can be seen from the lowest average runtime, with only 770s on average (compared to 3104s for the most difficult one – when vehicle capacities are halved).

5. Conclusions

The IRP-LR has been recently introduced in the literature as a variant of the classical IRP where the logistic ratio, i.e., the ratio between the routing cost and the total quantity delivered, is minimized. It is motivated by the fact that this objective function mitigates one of the main drawbacks of the classical IRP, the so-called 'end-of-the-horizon' effect, which consists in obtaining low inventory levels at the end of the planning horizon. The new objective function makes the problem more difficult to solve to optimality than the classical IRP. In this paper, a new exact algorithm for the IRP-LR is proposed and compared against the only known exact approach. The results show that the new algorithm outperforms the previous approach when the number of vehicles is small. Moreover, the new algorithm is able to solve instances with a larger number of customers and a longer planning horizon.

Future research needs to be done on the IRP-LR. In particular, given the complexity of the problem, a promising direction would be the design of heuristic algorithms capable of solving instances of larger size with respect to the ones tested in this paper. In addition, it would be interesting to include additional side constraints that often appear in real applications, like, for example, time windows at the customers and/or minimum quantities to be delivered at each visit.

Acknowledgments

This project was partly funded by the Canadian Natural Sciences and Engineering Research Council (NSERC) under grant 2019-00094. We thank Calcul Québec for proving high performance parallel computing facilities. We thank the editor and five anonymous referees for their valuable suggestions on an earlier version of this paper.

References

[1] Y. Adulyasak, J.-F. Cordeau, and R. Jans. Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems. *INFORMS Journal on Computing*, 26:103–120, 2013.

[2] A. Alvarez, P. Munari, and R. Morabito. Iterated local search and simulated annealing algorithms for the inventory routing problem. *International Transactions in Operational Research*, 25(6):1785–1809, 2018.

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

[4] C. Archetti and M. G. Speranza. The inventory routing problem: The value of integration. *International Transactions in Operational Research*, 23:393–407, 2016.

[5] 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: 382–391, 2007.

[6] C. Archetti, N. Bianchessi, S. Irnich, and M. G. Speranza. Formulations for an inventory routing problem. *International Transactions in Operational Research*, 21: 353–374, 2014.

[7] C. Archetti, N. Boland, and M. G. Speranza. A matheuristic for the multivehicle inventory routing problem. *INFORMS Journal on Computing*, 29(3):377–387, 2017.

[8] C. Archetti, G. Desaulniers, and M.G. Speranza. Minimizing the logistic ratio in the inventory routing problem. *EURO Journal on Transportation and Logistics*, 6: 289–306, 2017.

[9] L. Bertazzi and M.G. Speranza. Inventory routing problems: an introduction. *EURO Journal on Transportation and Logistics*, 1:307–326, 2012.

[10] L. Bertazzi and M.G. Speranza. Inventory routing problems with multiple customers. *EURO Journal on Transportation and Logistics*, 2:255–275, 2013.

[11] L. Bertazzi, M.G. Speranza, and M.W.P. Savelsbergh. Inventory routing. In B. Golden, R. Raghavan, and E. Wasil, editors, *The Vehicle Routing Problem Latest Advances and New Challenges, Operations Research/Computer Science Interfaces Series 43*, pages 49–72. Springer-Verlag, 2008.

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

[13] L. Bertazzi, L. C. Coelho, A. De Maio, and D. Laganà. A matheuristic algorithm for the multi-depot inventory routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 122:524–544, 2019.

[14] J. Brinkmann, M. W. Ulmer, and D. C. Mattfeld. Dynamic lookahead policies for stochastic-dynamic inventory routing in bike sharing systems. *Computers & Operations Research*, 106:260–279, 2019.

[15] C. Cheng, P. Yang, M. Qi, and L.-M. Rousseau. Modeling a green inventory routing problem with a heterogeneous fleet. *Transportation Research Part E: Logistics and Transportation Review*, 97:97–112, 2017.

[16] M. Chitsaz, J.-F. Cordeau, and R. Jans. A unified decomposition matheuristic for assembly, production and inventory routing. *INFORMS Journal on Computing*, 31 (1):134–152, 2019.

[17] L. C. Coelho and G. Laporte. The exact solution of several classes of inventory-routing problems. *Computers and Operations Research*, 40:558–565, 2013.

[18] L. C. Coelho and G. Laporte. Improved solutions for inventory-routing problems through valid inequalities and input ordering. *International Journal of Production Economics*, 155:391–397, 2014.

[19] L. C. Coelho, J.-F. Cordeau, and G. Laporte. The inventory-routing problem with transhipment. *Computer and Operations Research*, 39:2537–2548, 2012.

[20] L. C. Coelho, J.-F. Cordeau, and G. Laporte. Consistency in multi-vehicle inventory-routing. *Transportation Research Part C*, 24:270–287, 2012.

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

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

[23] M. Darvish, C. Archetti, and L. C. Coelho. Trade-offs between environmental and economic performance in production and inventory-routing problems. *International Journal of Production Economics*, forthcoming, 2018.

[24] G. Desaulniers, J. Rakke, and L.C. Coelho. A branch-price-and-cut algorithm for the inventory-routing problem. *Transportation Science*, 50:1060–1076, 2016.

[25] W. Dinkelbach. On nonlinear fractional programming. *Management Science*, 13: 492–498, 1967.

[26] P. Escalona, V. Marianov, F. Ordóñez, and R. Stegmaier. On the effect of inventory policies on distribution network design with several demand classes. *Transportation Research Part E: Logistics and Transportation Review*, 111:229–240, 2018.

[27] Y. Ghiami, E. Demir, T. Van Woensel, M. Christiansen, and G. Laporte. A deteriorating inventory routing problem for an inland liquefied natural gas distribution network. *Transportation Research Part B: Methodological*, 126:45–67, 2019.

[28] R. Giroudeau, M. Vasquez, and T. Jovanovic. Lower bound on the logistic ratio objective function for bulk distribution inventory-routing problem. In *VeRoLog: Vehicle Routing and Logistics optimization*, Nantes, France, 2016. URL `https://hal-lirmm.ccsd.cnrs.fr/lirmm-01378572`.

[29] T. A. Guimarães, L. C. Coelho, C. M. Schenekemberg, and C. T. Scarpin. The two-echelon multi-depot inventory-routing problem. *Computers & Operations Research*, 101:220–233, 2019.

[30] M. Mes, M. Schutten, and A. P. Rivera. Inventory routing for dynamic waste collection. *Waste Management*, 34(9):1564–1576, 2014.

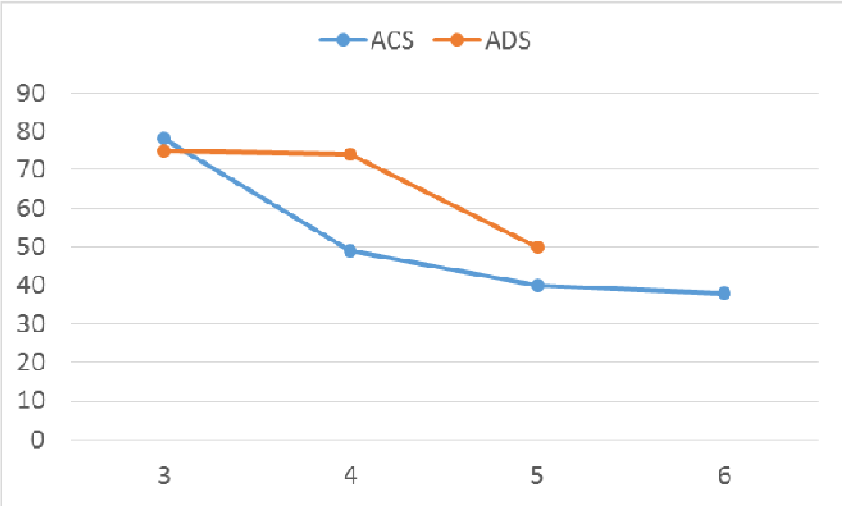
[31] H. Rau, S. D. Budiman, and G. A. Widyadana. Optimization of the multi-objective green cyclical inventory routing problem using discrete multi-swarm PSO method. *Transportation Research Part E: Logistics and Transportation Review*, 120:51–75, 2018.

[32] U. Ritzinger, J. Puchinger, and R. F. Hartl. A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research*, 54(1):215–231, 2016.

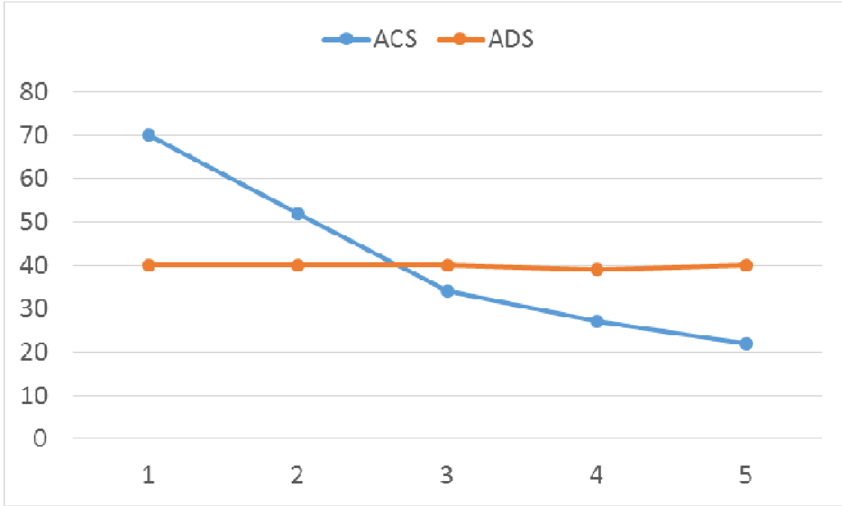
[33] R. F. Roldán, R. Basagoiti, and L. C. Coelho. Robustness of inventory replenishment and customer selection policies for the dynamic and stochastic inventory-routing problem. *Computers & Operations Research*, 74:14–20, 2016.

[34] R. F. Roldán, R. Basagoiti, and L. C. Coelho. A survey on the inventory-routing problem with stochastic lead times and demands. *Journal of Applied Logic*, 24:15–24, 2017.

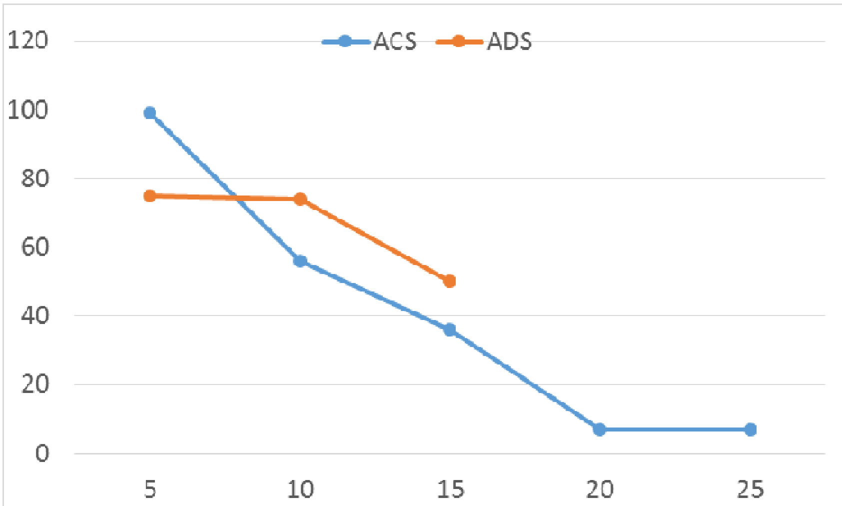
[35] H. R. Sayarshad and H. O. Gao. A non-myopic dynamic inventory routing and pricing problem. *Transportation Research Part E: Logistics and Transportation Review*, 109: 83–98, 2018.



a) Number of optimal solutions for different values of *H*



b) Number of optimal solutions for different values of $|K|$



c) Number of optimal solutions for different values of $|N'|$