A Matheuristic with Fixed-Sequence Reoptimization for a Real-Life Inventory Routing Problem
Yun He, Christian Artigues, Cyril Briand, Nicolas Jozefowiez, Sandra Ulrich Ngueveu

To cite this version:

HAL Id: hal-02944238
https://hal.science/hal-02944238
Submitted on 21 Sep 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Submitted to *Transportation Science*
manuscript TS-2017-0278.R2

Authors are encouraged to submit new papers to INFORMS journals by means of a style file template, which includes the journal title. However, use of a template does not certify that the paper has been accepted for publication in the named journal. INFORMS journal templates are for the exclusive purpose of submitting to an INFORMS journal and should not be used to distribute the papers in print or online or to submit the papers to another publication.

A Matheuristic with Fixed-sequence Re-optimization for a Real-life Inventory Routing Problem

Yun He
LAAS-CNRS, Université de Toulouse, CNRS, UPS, Toulouse, France
IMT-Atlantique, Laboratoire des Sciences de Numérique de Nantes (LS2N, UMR CNRS 6004), Nantes, France
School of Computer Science, University of Nottingham, Nottingham, United Kingdom, yun.he@nottingham.ac.uk

Christian Artigues, Cyril Briand
LAAS-CNRS, Université de Toulouse, CNRS, UPS, Toulouse, France, artigues@laas.fr, briand@laas.fr

Nicolas Jozefowiez
LCOMS EA 7306, Université de Lorraine, Metz 57000, France, nicolas.jozefowiez@univ-lorraine.fr

Sandra Ulrich Ngueveu
LAAS-CNRS, Université de Toulouse, CNRS, INP, Toulouse, France, ngueveu@laas.fr

This paper proposes a matheuristic for solving a real-life Inventory Routing Problem introduced in the ROADEF/EURO Challenge 2016. The method integrates a fixed-sequence mathematical program, two randomized greedy algorithms, and a column-generation based heuristic. In particular, the paper discusses the performance of the fixed-sequence mathematical program, which considers a fixed sequence of customer visits and aims at (re)optimizing partial solutions by modifying arrival times and delivered or loaded quantities. Experiments show that the proposed algorithm for the fixed-sequence sub-problem is efficient as a post-optimization process and is even able to improve the best results obtained during the Challenge.

**Key words**: Inventory routing problem; Matheuristic; Mixed Integer Linear Fractional Programming

1. Introduction

The aim of this paper is to propose a matheuristic for solving the Inventory Routing Problem introduced in the ROADEF/EURO Challenge 2016 (simply referred to as “the Challenge” in the sequel). In particular, the so-called Fixed-Sequence Continuous Inventory Routing Problem (FSCIRP) is identified. FSCIRP works on a fixed sequence of visits to customers and aims at re-optimizing the solution by modifying arrival times and delivered quantities. Experiments show that the proposed algorithm for the solution of FSCIRP is efficient as a post-optimization process. It is able to improve the best results obtained during the Challenge.
The Inventory Routing Problem (IRP) is an integration of inventory management and vehicle routing under the Vendor Managed Inventory (VMI) business model. Mainly found in road-based and maritime supply chains, especially in the distribution of industrial gases or petrol oil using trucks or ships, the problem considers a set of customers whose inventory is monitored by a vendor. The vendor has to decide when and how much to deliver to each customer and how to route vehicles so that the customers never run out of stock while total costs, induced by both inventory management and vehicle routing operations, are minimized.

Both complex in nature and important in industrial applications, the IRP has attracted the attention of practitioners and academic researchers since the first study by Bell et al. (1983). Our paper does not intend to provide a complete literature review of IRPs. For a general introduction of the problem, interested readers can refer to the tutorials of Bertazzi and Speranza (2012, 2013). Industrial aspects of IRPs in maritime and road-based transportation are described in the review of Andersson et al. (2010). A more recent comprehensive literature review is provided by Coelho et al. (2013), with a detailed classification and a focus on existing solution methods.

Most studies, such as those by Archetti et al. (2007, 2012), consider the problem with a time horizon divided into several periods (while the duration of a period is not specified). In each period, a subset of customers is chosen to be visited and a set of vehicles is routed to make deliveries among these customers. During each period, the inventory level of each customer is assumed constant. Each period is also assumed long enough for each vehicle to complete one tour. Each tour starts and ends at the depot and vehicles are automatically refilled once returned to the depot. The most common objective is the minimization of the combined costs of inventory holding and transportation. A set of benchmarks was provided by Archetti et al. (2007). They also proposed the first exact algorithm to solve the problem. Larger instances were created by Coelho et al. (2012) and compatibility issues among customers, drivers and vehicles were discussed. However, in comparison with real-life instances used in this paper, the benchmarks proposed by Archetti et al. (2007), Coelho et al. (2012) remain relatively small and simplified.

In the literature, IRPs with real applications are usually decomposed into several subproblems which can be solved separately using hybrid methods relying on mathematical programming and heuristics, called “matheuristics”. A common decomposition is to divide the whole horizon into smaller periods. The long-term problem is reduced to several short-term sub-problems, each being solved as a classic routing problem with some additional features regarding customer inventory levels. Rolling-horizon approaches are often applied as in Campbell et al. (2001), Campbell and Savelsbergh (2004). The decomposition can also be done according to different decision processes. For example, Cordeau et al. (2015) developed a decomposition method that divides the problem into one subproblem of inventory replenishment planning and another of route construction. The
former subproblem is solved first using a Lagrangian-based heuristic and then vehicle routes are built using another heuristic. Eventually, the solutions of the two subproblems are given to a post-processing mathematical model for re-optimization. Similar ideas can be found in Grønhaug et al. (2010), Andersson et al. (2016), where “duties” consisting of a geographical route, a schedule and a vessel unloading plan are generated a priori. A mathematical model is then solved for selecting the “best” duties. In Hewitt et al. (2013), a Branch–and–Price (B&P) guided search was applied to a real-world maritime IRP where a series of small-size Mixed Integer Linear Programming (MILP) is solved to obtain heuristic solutions. Among the matheuristics with the best performance, Archetti et al. (2012) proposed a hybridization of a mathematical programming method with a Tabu search. Coelho et al. (2012) used a MILP component inside the scheme of the Adapted Large Neighbourhood Search (ALNS). Desaulniers et al. (2016) proposed a mathematical formulation for the IRP and a Branch-Cut-and-Price algorithm, where several valid inequalities are included.

In 2016, the Challenge focused on a large-scale real-life IRP met by the French gas company Air Liquide. The problem deals with two kinds of customers: *VMI customers*, having their inventory managed by the gas company, and *call-in customers*, who place refilling orders directly to the company when needed. The goal of the problem is to plan shifts for drivers using trailers to deliver products to the customers. The inventory levels of VMI customers must never fall below a given safety level and the orders of call-in customers must be fulfilled in time. In comparison to classic IRPs, one main difficulty is that the routing part of the problem has to be managed in nearly continuous time with a precision of one minute. A second complex feature is that the problem deals with two different time scales: minutes for routing and hours for inventory level monitoring. A third complicating factor is the need to assign each route to a pair of driver/trailer, so that compatibility constraints are satisfied, and limits on daily working duration of drivers are also respected. In addition, the trailers are not automatically refilled at the beginning of each route, so the quantity in each trailer has also to be managed. One last complicating feature, not often studied in classic IRP literature, is that the objective function aims to minimize a logistic ratio, i.e. the ratio between the total cost (induced by shifts, drivers, and trailers) and the total delivered quantity.

A problem similar to the Challenge problem was studied by Benoist et al. (2011), who proposed a local search method. Some of the complex features mentioned above have also been partly addressed in the literature. For example, combined vehicle routing and scheduling is a common topic on Vehicle Routing Problems (VRPs) as reviewed by Laporte (2016). Although it was considered in the initial paper of Bell et al. (1983), this combination is rare in classic IRPs. The management of vehicle quantity and compatibility issues have also been partly studied in Coelho et al. (2012). During and after the Challenge, variants of IRPs continue to emerge that tackle one or several
of these features. For the time discretization, Lagos et al. (2018) studied a simplified variant of IRP in continuous time with a homogeneous fleet of vehicles, with a constant demand rate per customer, aiming to minimize total travel cost. They revealed interesting properties of the problem and applied the Dynamic Discretization Discovery scheme for solving it. For optimizing the logistic ratio, Archetti et al. (2017) adapted the classic IRP model, provided theoretical insights on bounds of the logistic ratio and adapted Dinkelbach’s algorithm for the solution. Later, Archetti et al. (2018) proposed an exact method for solving the problem. To the best of the knowledge of the authors, the Challenge problem studied in this paper is original and there does not exist any mathematical model that includes all the practical aspects of the Challenge problem.

The main contribution of this work is an integrated method that solves real-life IRP problem instances using three components: a fixed-sequence mathematical program, two randomized greedy algorithms, and a column-generation based heuristic. More specifically, we highlight the interest of the fixed-sequence mathematical program, which can be successfully used as a re-optimization method to improve any solution (in particular those provided by the other two components). It consists in a model and an algorithm to re-optimize a solution by optimally recomputing the arrival time and quantity of each operation, provided that the sequence of operations and the assignment of drivers and trailers are kept unchanged. We refer to this re-optimization problem as the Fixed-Sequence Continuous Inventory Routing Problem (FSCIRP). Our experiments show that FSCIRP can be solved efficiently and is able to strictly improve the results obtained by the best methods submitted during the Challenge. We also discuss the promising potential of our column-generation based heuristic.

The paper is organized as follows. Section 2 introduces first the Challenge problem in general, then details FSCIRP, as well as the fractional programming algorithm to solve it. Section 3 describes a general decomposition scheme for solving the Challenge problem. Notably, it presents the greedy algorithms that provide initial solutions for FSCIRP and the column-generation-based matheuristic that takes advantage of the solutions given by FSCIRP. Computational experiments are conducted and analyzed in Section 4 and conclusions are drawn in Section 5.

2. The Challenge Problem and the Fixed-Sequence Continuous Inventory Routing Problem

This section presents the Challenge problem in general and then details FSCIRP and its solution.

2.1. Presentation of the problems

The Challenge problem is defined over a time horizon $H$. A set of customers $\mathcal{Z}$ must be visited using a set of trailers $\mathcal{T L}$ driven by a set of drivers $\mathcal{D R}$. Other sites include the depot, denoted 0 and a set of sources $\mathcal{SO}$ to refill the trailers. Given the hourly consumption forecast of each
customer, the problem is to plan, over time horizon $H$, a set of shifts for each driver and trailer, so as to avoid stock-outs at the customers and to minimize global delivery cost.

A shift is a sequence of timed operations made by a driver using a trailer. An operation is a visit of a driver and a trailer to a site which is either a customer or a product source. An operation must be scheduled with an accuracy of one minute and associated with a delivered (or loaded) quantity. A trailer must be assigned to one single driver during each shift but may be reassigned to another driver in another shift later. The sequence of shifts assigned to a driver is called a route. Shift operations are subject to numerous constraints, including complex driver shift constraints. For a more accurate description of the Challenge problem itself we refer to the Challenge subject\textsuperscript{1}. This paper focuses on a particular subproblem, FSCIRP, that is detailed in the remaining of this section.

FSCIRP is a subproblem of the Challenge problem where the order of operations of each driver is fixed, as well as the assignment of the driver/trailer pair to each shift. The arrival time at each site and the quantity to be delivered or loaded in each operation are the remaining decisions to be made.

The method to solve FSCIRP presented in this section can be used as a subroutine for other methods to solve the Challenge problem, such as the one presented in Section 3. We will show in Section 4 that solving the FSCIRP can improve the solutions obtained by the best algorithms submitted to the Challenge.

2.2. FSCIRP Input

The input data for FSCIRP are similar to those for the Challenge problem with the addition of a set of shifts. These inputs are described in details below. An instance for FSCIRP can easily be obtained from an initial solution to the Challenge problem. This initial solution may not respect all time windows or capacity constraints of the Challenge problem, since only sequences of operations matter in the FSCIRP. In this case, the FSCIRP first decides whether the given sequences of operations is feasible. If the answer is yes, then it optimizes the arrival time and delivered quantity of each operation.

2.2.1. Trailers Each trailer $tl \in TL$ is defined by initial loaded quantity $J_{0tl}$, maximum capacity $Q_{tl}$, and per-distance-unit usage cost $C_{tl}^{distance}$.

2.2.2. Drivers Each driver $d \in DR$ can only work during time windows belonging to set $TW_d$. Each time window $tw \in TW_d$ is defined by interval $[A_{tw}^d, B_{tw}^d]$ with $A_{tw}^d$ and $B_{tw}^d$ the starting and ending times of the time window in minutes. Cost $C_{d}^{time}$ must be paid for each working minute of driver $d$. Driver $d$ can only operate a subset of trailers denoted by $TL_d \subseteq TL$.

\textsuperscript{1}http://www.roadef.org/challenge/2016/files/IRP\_AL\_Model\%20Description\%20for\%20EURO-ROADEF\%20Challenge\%20Version\%202.2.pdf
2.2.3. Shifts The set of shifts is denoted by $\mathcal{SH}$. Each shift $s \in \mathcal{SH}$ is performed by driver $d_s \in \mathcal{DR}$ with one of his compatible trailers $t_s \in \mathcal{TL}_{d_s}$ during one of his time windows $tw_s \in \mathcal{TW}_{d_s}$. Each shift $s \in \mathcal{SH}$ is a sequence of operations $N_s$ that starts and ends at the depot (with $|N_s|$ the total number of operations in the shift including starting and ending depot). Both drivers and trailers are located at the depot at the beginning of the time horizon. In addition to the depot, there are two other types of sites: customers and sources. These different types of sites are presented more in details below. The $i^{th}$ site in shift $s$ is denoted by $N_s(i)$. Notably, we have $N_s(1) = N_s(|N_s|) = 0$.

Subset $\mathcal{SH}_{tw}^d \subseteq \mathcal{SH}$ is the ordered set of shifts that takes place inside time window $tw \in \mathcal{TW}_d$ of driver $d \in \mathcal{DR}$. The $i^{th}$ shift in the set is denoted $\mathcal{SH}_{tw}^d(i)$. Similarly, subset $\mathcal{SH}_{tl}^t \subseteq \mathcal{SH}$ is the ordered set of shifts that is assigned to trailer $tl \in \mathcal{TL}$, with $\mathcal{SH}_{tl}^t(i)$ the $i^{th}$ shift performed by trailer $tl$.

Obviously, a driver or a trailer can only be assigned to at most one shift at each time instant. Driver $d \in \mathcal{DR}$ cannot drive more than a maximum duration denoted by $MDD_d$ (in minutes) in each of his shift. He can perform several shifts consecutively in the same time window but must rest for a minimum inter-shift duration denoted by $MIS_d$ (also in minutes) between two consecutive shifts.

2.2.4. Sources The set of sources $\mathcal{SO}$ contains sites where a trailer can be refilled if needed. Sources are always available and it is always possible to refill a trailer up to its capacity. A setup time (in minutes) $ST_i$ is required to carry out the service at each source $i \in \mathcal{SO}$. Also, only a subset of trailers $\mathcal{TL}_i \subseteq \mathcal{TL}$ can have access to each source $i \in \mathcal{SO}$.

2.2.5. Distance and travelling time For each pair of sites $(i, j)$, with $i, j \in \{0\} \cup \mathcal{Z} \cup \mathcal{SO}$, $D_{i,j}$ and $T_{i,j}$ denote respectively the travel distance in kilometers and the travel time in minutes.

2.2.6. Customers Any delivery operation performed at customer $i \in \mathcal{Z}$ has service time $ST_i$. Each customer is only compatible with a subset of trailers $\mathcal{TL}_i \subseteq \mathcal{TL}$. As mentioned before, the set of customers is divided into the set of VMI customers $\mathcal{Z}_{vmi}$ and the set of call-in customers $\mathcal{Z}_{ci}$.

Each VMI customer $i \in \mathcal{Z}_{vmi}$ has initial inventory level $I^0_i$ and a forecast consumption of $R^h_i$ units of product in each hour $h \in \mathcal{H}$. He can only be visited during one of the time windows in set $\mathcal{TW}_i$. Each time window $tw \in \mathcal{TW}_i$ is defined by interval $[a^{tw}_i, b^{tw}_i]$, where $a^{tw}_i$ and $b^{tw}_i$ are the starting and ending times of the time window in minutes. The inventory level of each VMI customer $i \in \mathcal{Z}_{vmi}$ must always remain between safety level $I_i$ and maximum tank capacity $I_i$. There is also a minimum delivery quantity $R^{min}_i$ for each VMI customer $i \in \mathcal{Z}_{vmi}$.

Each call-in customer $i \in \mathcal{Z}_{ci}$ has a set of orders $\mathcal{OD}_i$. Each order $od \in \mathcal{OD}_i$ is defined by required quantity $R^{od}_i$, delivery flexibility $f^{od}_i$ given as a percentage, and delivery time window $[a^{od}_i, b^{od}_i]$ where starting time $a^{od}_i$ and ending time $b^{od}_i$ are expressed in minutes.
2.2.7. Layover customers Some customers are too far to be reached within the maximum driving duration of the drivers. These customers are referred to as layover customers, denoted by set $Z_{lo}$. A layover customer can be either VMI or call-in, so we have $Z_{lo} \subseteq Z_{vmi} \cup Z_{ci}$. To visit a layover customer during a shift, a layover pause must be inserted between two operations of the shift, provided that the driving times before and after the layover pause do not exceed the maximum driving duration of the driver in this shift. For each driver $d \in DR$, the layover pause duration in minutes is denoted by $LOD_d$. Each layover pause by driver $d$ induces fixed cost $c_d^{layover}$.

The set of shifts containing at least one layover customer is referred to as $SH^l$.

### 2.3. Decision variables

The FSCIRP can be modeled as a mixed-integer program. The decision variables are presented in this section.

2.3.1. Arrival times at sites Binary variable $v_{s,h}^i$ is defined for each shift $s \in SH$ at each hour $h \in H$ and for each operation $i \in [1, |N_s|]$. The variable is equal to 1 if and only if site $N_s(i)$ is visited during hour $h$. Continuous variable $t_s^i$ is defined for each operation $i \in [1, |N_s|]$ in each shift $s \in SH$. This variable is equal to the arrival time in minutes at site $N_s(i)$. Continuous variable $z_s^i \in [0, 59]$ is defined for each operation $i \in [1, |N_s|]$ in each shift $s \in SH$. The variable is equal to the remaining time in minutes in the hour when site $N_s(i)$ is visited.

The variables related to arrival times at sites with their domains induced by the data is recapitulated in (1)–(3).

$$v_{s,h}^i \in \{0, 1\} \quad \forall s \in SH, \forall i \in [1, |N_s|], \forall h \in H \quad (1)$$

$$t_s^i \in [A^s_w, B^s_w] \quad \forall d \in DR, \forall tw \in TW_d, \forall s \in SH^{tw}, \forall i \in [1, |N_s|] \quad (2)$$

$$z_s^i \in [0, 59] \quad \forall s \in SH, \forall i \in [1, |N_s|] \quad (3)$$

2.3.2. Trailer capacity and customer inventory levels Continuous variable $J_{tl}^{s,i}$ is defined for each trailer $tl \in TL$ and for each operation $i \in [1, |N_s|]$ of each shift $s \in SH^{dl}$. The variable is equal to the quantity of product left in trailer $tl$ after the operation at site $N_s(i)$ in shift $s$. Continuous variable $q_{s,h}^{s,i}$ is defined for each operation $i \in [1, |N_s|]$ in each shift $s \in SH$ at each hour $h \in H$. If site $N_s(i)$ is a customer, the variable is equal to the quantity delivered at hour $h$. If $N_s(i)$ is a source, the variable is equal to the quantity loaded to the trailer performing shift $s$ at hour $h$. The inventory level of each VMI customer $j \in Z_{vmi}$ at each hour $h$ is modeled by continuous variable $I_j^h$. The variable domains are given by Constraints (4)–(8).

$$J_{tl}^{s,i} \in [0, Q_{tl}] \quad \forall tl \in TL, \forall s \in SH^{dl}, \forall i \in [1, |N_s|] \quad (4)$$

$$q_{s,h}^{s,i} \in [0, T_{N_s(i)}] \quad \forall s \in SH, \forall h \in H, \forall i \in \{[1, |N_s|] : N_s(i) \in Z_{vmi}\} \quad (5)$$

$$q_{s,h}^{s,i} \geq 0 \quad \forall s \in SH, \forall h \in H, \forall i \in \{[1, |N_s|] : N_s(i) \in Z_{ci}\} \quad (6)$$
2.3. Layover pauses

Binary variable $l^s_i$ is defined for each operation $i \in [2, |N_s|]$ in each shift $s \in S_H$. Recall that $S_H$ is the set of shifts with at least one layover customer. The variable is equal to 1 if and only if a layover pause is scheduled just before visiting site $N_s(i)$. In order to check the time consistency of shifts, the exact timing of each layover pause has to be specified. For this purpose, continuous variable $p^s_i \in [0, 1]$ is defined for each operation $i \in [2, |N_s|]$ of each shift $s \in S_H$. Let $t^s_i$ be the arrival time at operation $i \in [2, |N_s|]$ in shift $s \in S_H$ and $\lambda$ the starting time of the layover pause if it is planned before operation $i$ in shift $s$. Then, variable $p^s_i$ is defined as $p^s_i = \frac{\lambda - t^s_i - 1}{t^s_i - t^s_1}$.

The variable domains are the following:

$$l^s_i \in \{0, 1\} \quad \forall s \in S_H, \forall i \in [2, |N_s|] \quad (9)$$
$$p^s_i \in [0, 1] \quad \forall s \in S_H, \forall i \in [2, |N_s|] \quad (10)$$

2.4. Constraints

2.4.1. Sequence of operations

The following constraints ensure that the sequence of operations defined in shifts are respected.

$$ST_{N_s(i)} + T_{N_s(i), N_s(i+1)} + LOD_{d, N_s(i+1)} I^s_{N_s(i+1)} \leq t^s_{i+1} - t^s_i \quad \forall d \in D_R, \forall s \in S_H, \forall i \in [1, |N_s| - 1] \quad (11)$$
$$MIS_d \leq t^{|S_H^d|}_{l^s_{N_s(i+1)}} - t^{|S_H^d|}_{l^s_{N_s(i)}} \quad \forall d \in D_R, \forall s \in T_W, \forall i \in [1, |S_H^d| - 1] \quad (12)$$

Constraints (11) make sure that, if there is no layover, the duration between any two consecutive operations should be at least the setup time of the previous operation plus the travelling duration between the two sites. If there is a layover planned before the $(i + 1)^{th}$ operation, the layover duration is added to the minimum required transition time. Constraints (12) state that the duration between two consecutive shifts inside the same time window of the same driver should be no less than the minimum inter-shift duration of the driver.

2.4.2. Timing of arrivals at sites in minutes

$$t^s_i = 60 \sum_{h \in H} h v^s_{i,h} + z^s_i \quad \forall s \in S_H, \forall i \in [1, |N_s|] \quad (13)$$
$$0 \leq z^s_i \leq 59 \quad \forall s \in S_H, \forall i \in [1, |N_s|] \quad (14)$$
$$\sum_{h \in H} v^s_{i,h} = 1 \quad \forall s \in S_H, \forall i \in [1, |N_s|] \quad (15)$$

Constraints (13)–(14) ensure the coherence between binary variables $v$ and continuous variables $t$ for the scheduling of operations. If $t^s_i$ is inside an hour $h$, then $v^s_{i,h}$ is set to 1. Constraints (15) ensure that each operation in a shift is scheduled at one specific hour. Note that this kind of constraints tends to be more efficient than the classic big M constraints.
2.4.3. VMI customer time windows

\[
\begin{align*}
    &a_{v_{s}(i)}^a \leq t_{s}^i \leq b_{v_{s}(i)}^a - ST_{v_{s}(i)} & \forall s \in SH, \forall i \in \{1, |N_s| : N_s(i) \in Z_{vmi}\}
\end{align*}
\]  

Constraints (16) make sure that all the operations at a VMI customer occur inside a predefined time window.

2.4.4. Trailer capacity

\[
\begin{align*}
    &J_{tl}^i + J_{tl}^{i+1} - \sum_{h \in H} q_{i}^{s,h} & \forall tl \in TL, \forall s \in SHtl, \forall i \in [1, |N_s| - 1] \\
    &J_{tl}^{SHtl(i)+1} = J_{tl}^{SHtl(i)} & \forall tl \in TL, \forall i \in [1, |SHtl| - 1] \\
    &-Q_{tl} \leq q_{i}^{s,h} \leq 0 & \forall tl \in TL, \forall s \in SHtl, \forall i \in \{1, |N_s| : N_s(i) \in SO\} \\
    &0 \leq J_{tl}^{i} \leq Q_{tl} & \forall tl \in TL, \forall s \in SHtl, \forall i \in [1, |N_s|]
\end{align*}
\]  

Constraints (17) maintain the coherence of the trailer quantity between consecutive operations inside each shift. Constraints (18) enforce the coherence of the trailer quantity among consecutive shifts performed by the same trailer. Constraints (19) ensure that the quantity obtained from sources never exceeds trailer capacity. Finally, constraints (20) state that the quantity in each trailer is positive and never exceeds the capacity of the trailer.

2.4.5. Customer inventory management

\[
\begin{align*}
    &I_{j}^{h+1} = I_{j}^{h} + \sum_{s \in SH \ i \in \{2, |N_s| - 1 : N_s(i) = j\}} q_{i}^{s,h} - R_{j}^{h} & \forall j \in Z_{vmi}, \forall h \in H \setminus \{H\} \\
    &R_{N_{s}(i)} v_{i}^{s,h} \leq q_{i}^{s,h} \leq (I_{N_{s}(i)} - L_{N_{s}(i)}) v_{i}^{s,h} & \forall s \in SH, \forall h \in H, \forall i \in \{1, |N_s| : N_s(i) \in Z_{vmi}\} \\
    &I_{j}^{h} \leq I_{j}^{h} \leq \bar{I}_{j} & \forall j \in Z_{vmi}, \forall h \in H \\
    &R_{j}^{od} f_{j}^{od} \leq \sum_{s \in SH \ i \in \{2, |N_s| - 1 : N_s(i) = j\}} \sum_{h \in \{a^{od}, b^{od}\}} q_{i}^{s,h} \leq R_{j}^{od} & \forall j \in Z_{ci}, \forall od \in OD_j
\end{align*}
\]  

Constraints (21) enforce the inventory balance of each VMI customer from hour to hour. They say that the inventory level in the next hour equals the current inventory level plus the delivered quantity in the current hour minus the forecast demand faced by the customer in the current hour. Constraints (22) make sure that the delivered quantity stays in the allowed limits of VMI customers. Constraints (23) ensure that the inventory level of each VMI customer never exceeds its capacity and always stays above its safety level. Constraints (24) are for the fulfillment of call-in orders. They impose the delivered quantity to be at least the minimum percentage needed to satisfy an order, without exceeding the maximum deliverable level.
2.4.6. Layover pauses

\[ \sum_{i=2}^{N_s-1} l_i^s = 1 \quad \forall s \in SH^i \] (25)

\[ \sum_{i=2}^{N_s-1} l_i^s = 0 \quad \forall s \in SH \setminus SH^i \] (26)

\[ p_i^s \leq l_i^s \quad \forall s \in SH, \forall i \in [2, |N_s|] \] (27)

\[ \sum_{j=1}^{N_s} \sum_{k=j}^{N_s} T_{N_s(j-2), N_s(j-1)} l_k^s + T_{N_s(i-1), N_s(i)} p_i^s \leq MDD_{d_s} \quad \forall s \in SH, \forall i \in [3, |N_s|] \] (28)

\[ \sum_{j=2}^{N_s} T_{N_s(j-1), N_s(j)} - \left( \sum_{j=i}^{N_s} \sum_{k=j}^{N_s} T_{N_s(j-2), N_s(j-1)} l_k^s + T_{N_s(i-1), N_s(i)} p_i^s \right) \leq MDD_{d_s} \quad \forall s \in SH, \forall i \in [3, |N_s|] \] (29)

The fact that a shift with a layover customer can have one and only one layover pause is modelled by Constraints (25). Conversely, Constraints (26) ensure that there is no layover pause inside a shift without a layover customer. Constraints (27) set variable \( p_i^s \) to 0 if there is no layover before the \( i \)-th operation of each shift \( s \) (\( l_i^s = 0 \)). Constraints (28)–(29) enforce the driving duration inside each shift to never exceed the maximum driving duration of the driver performing this shift. If the layover pause exists (i.e., \( l_i^s = 1 \)), Constraints (28) and (29) allow the computation of the total driving times before and after the layover pause, respectively; otherwise, Constraints (28) become \( 0 \leq MDD_d \) and Constraints (29) state that the total driving time of the shift is bounded by the maximum driving duration, which is always satisfied in the case of a feasible initial solution.

2.5. Objective Function

The goal is to minimize the distribution costs in the long term. To achieve this goal, the logistic ratio \( LR \) is defined in Equation (30) as the total cost of all the shifts divided by the total quantity delivered in these shifts.

\[ LR = \frac{J + D + L}{Q} \] (30)

The detailed computation of these costs are given by Equations (31)–(33). The total quantity \( Q \) delivered by all shifts is the sum of the quantities in each delivery operation of each shift. It is given by Equation (34).
\[ T = \sum_{d \in DR} \sum_{t_w \in TW} \sum_{s \in SH} C_{td}^{\text{time}} (t_s^d(|N_s|) - t_1^s - \sum_{i=2}^{\vert N_s \vert - 1} l_i^s LOD_d) \] (31)

\[ D = \sum_{t_l \in TL} \sum_{s \in SH} \sum_{i=1}^{\vert N_s \vert - 1} t_{l}^{s,i} |N_s(i)| - 1 \sum_{i=1}^{\vert N_s \vert - 1} C_{tl}^{\text{distance}} D_{N_s(i), N_s(i+1)} \] (32)

\[ L = \sum_{d \in DR} \sum_{t_w \in TW} \sum_{s \in SH} t_{l}^{s,i} |N_s(i)| - 1 \sum_{i=2}^{\vert N_s \vert - 1} \sum_{i=1}^{\vert N_s \vert - 1} C_{l}^{\text{layover}} \] (33)

\[ Q = \sum_{s \in SH} \sum_{h \in H} \sum_{i \in \{2, \ldots, \vert N_s \vert - 1\}: N_s(i) \in Z} q_{i,h} \] (34)

Note that since the sequences of operations are assumed known, the total distance value \( D \) is a constant in the FSCIRP.

2.6. Fractional Programming Algorithm

Because the objective function is not linear, one cannot solve the model directly by a Mixed Integer Linear Programming (MILP) solver. However, it is possible to employ Dinkelbach’s algorithm in Dinkelbach (1967) as illustrated in Algorithm 1 for the FSCIRP case. In this algorithm, the problem of finding vectors

\[ t = (t_s^{s,i})_{s \in SH, i \in [1, \vert N_s \vert]}, 1 = (l_{i}^{s,i})_{s \in SH, i \in [2, \vert N_s \vert]}, q = (q_{i,h})_{s \in SH, h \in H} \]

in the feasible space \( S \) defined by the constraints above to minimize the fractional objective \( \alpha = \frac{T(t, 1, Q(q))}{D(q)} \) can be linearized to minimizing \( (T + D + L) - \alpha Q \). In this way, the FSCIRP turns into a MILP that can be solved iteratively. In the sequel, we refer to this MILP as Fixed-Sequence Mixed Integer Linear Fractional Programming (FS–MILFP). The algorithm starts from an initial value of \( \alpha \), solves the resulting FS–MILFP, and adjusts the value of \( \alpha \) until a convergence criterion is met. This algorithm is shown to converge superlinearly for MILFPs, by providing a sequence of monotonely decreasing upper bounds on the optimal fractional objective as proved by You et al. (2009). If the last MILP is solved to optimality, the corresponding upper bound is optimal. Note that, provided that the input fixed sequence of shifts is feasible, the algorithm allows to obtain upper bounds for the Challenge problem.

It should be noted that, after solving an instance of the FS–MILFP model, one can face the situation where some of the customers in the fixed sequences are visited without any delivered quantity. In that case, these unnecessary operations are removed from the sequences. The new sequences without the unnecessary operations are given back to the FS–MILFP solver for re-optimization.
He et al.: A Matheuristic for a Real-life IRP
Article submitted to Transportation Science; manuscript no. TS-2017-0278.R2

Algorithm 1 Algorithm for fractional programming

1: Initialize $\alpha$ with a feasible logistic ratio $LR_0$
2: Solve the FS-MILFP model with the linearised objective function
3: Recompute the ratio $\alpha' = \frac{2 + Q}{2 + D}$ with the current solution
4: while $\alpha' < \alpha - \epsilon$ do (\(\epsilon\) is a tolerance for optimality)
5: \hspace{1em} $\alpha \leftarrow \alpha'$
6: Repeat step 2 and 3

This model can be easily adapted to the case with an infeasible initial solution by adding slack variables for missed orders or stock-outs and by changing the objective to minimizing the total number of missed orders and the total missed quantity. In that case, if a feasible solution exists after re-optimization, the logistic ratio objective function is restored.

3. Decomposition algorithm for the Challenge Problem

In this section, we present a general decomposition scheme for the solution of the Challenge problem. Recall that a route is a sequence of shifts assigned to a driver. Based on the notion of routes, the problem can be decomposed into two parts: (D1) the generation of routes and their assignment to drivers; (D2) the optimization of visiting time and delivered quantity for each operation, once a route has been selected for each driver.

To find the sequence of operations and the assignment of shifts to drivers (decisions D1), two greedy heuristics and a matheuristic based on column generation are proposed. The FS–MILFP defined in Section 2 is used for the optimization of delivery quantity and timing of operations in each shift (decisions D2). The complete solution method integrates these components as shown in Figure 1.

Figure 1 General solution method

In the following, the heuristics for finding the best sequences of operations are presented. We start with the randomized greedy algorithms and then present the column-generation based matheuristic.
3.1. Randomized Greedy Algorithms

In order to obtain starting solutions for the Challenge problem, two greedy heuristics were designed. The first one is a novel approach based on the notion of state. The second one is based on the urgency of the customers and is adapted from the heuristic proposed by Benoist et al. (2011). Only the state-based heuristic is detailed in the following.

Given an inventory routing system, its state at a given instant can be defined by: the position of each trailer and each driver at this time; the driving duration spent so far by each driver; the quantity left in each trailer and at each customer; the time of the last visit to each customer. The algorithm starts with a list of customers to serve inside the time horizon. Then, it randomly chooses a customer \( i \) from the list, with respect to the latest visit time \( \lambda_i \), which is computed according to the state of the customers and their time windows.

To add a new visit to customer \( i \), the greedy needs to find a pair of compatible driver/trailer \((d, tl)\). Actually, given the current state of the system and considering a compatible pair \((d, tl)\) that would visit a customer \( i \), there are 5 possible actions: (1) changing the time window of the driver; (2) starting a new shift (without changing the time window of the driver); (3) making a layover pause (without changing the shift); (4) visiting a source (to refill the trailer); (5) waiting (for the customer to open). A procedure enumerates all the possible combinations of these 5 actions, which gives a set of valid arrival times for the driver/trailer pair \((d, tl)\) to visit the customer \( i \), together with the corresponding maximum quantity that can be delivered at each valid arrival time. The slack time \( \theta_{d,tl,i} \) is then computed for each triplet of driver/trailer/customer \((d, tl, i)\) as explained in Figure 2. The arrival time is then chosen randomly from the set of valid arrival times computed during the previous enumeration procedure.

![Figure 2: Definition of slack \( \theta_{d,tl,i} \) for driver/trailer pair \((d, tl)\) and customer \( i \)](image)

According to the slack time \( \theta_{d,tl,i} \) associated with each compatible triplet of driver/trailer/customer \((d, tl, i)\), a driver/trailer pair \((d, tl)\) with positive slack is randomly chosen to visit customer \( i \). Since the arrival time is known (thanks to the previous phase), the visit of customer \( i \) is added to the solution and a delivered quantity is randomly chosen between the minimum and the maximum deliverable quantity. If nothing can be done for visiting customer \( i \) in time (before a stockout occurs), then the customer is considered lost and is removed from the list.

The state of the system is then updated. The iteration goes on until no customer will run out of stock or no order exists by the end of the time horizon. The complete algorithm is summarized in Algorithm 2.
Algorithm 2 State-based greedy heuristic

1: initialize the state of the system
2: construct the list $L$ of customers to serve
3: for all compatible triplet of driver/trailer/customer $(d, tl, i)$ do
   4: compute the slack time for the driver/trailer pair $(d, tl)$ to visit customer $i$
   5: while $L \neq \emptyset$ do
      6: choose randomly a customer $i$ from $L$
      7: choose randomly a valid driver/trailer pair $(d, tl)$ such that $\theta_{d, tl, i} > 0$
      8: if such a pair $(d, tl)$ exists then
         9: add operation to $i$ by $(d, tl)$ following the actions computed in Step 4
      else
      10: remove $i$ from $L$
      11: if $i$ will not have stock out in the planning horizon then
         12: remove $i$ from $L$
      13: update the state of the system
      repeat 3–4

3.2. Column Generation Based Heuristic

To select a promising route for each driver (decision D1), a Mixed Integer Linear Fractional Programming with Timed Route and Aggregated Time Units (RT–MILFP) formulation is proposed. To reduce the number of variables in this formulation, we assume that the trailer inventory balance is aggregated to the hourly level, which could possibly yield infeasible solutions. However, as explained in section 2, feasible solutions may be derived by solving FS–MILFP to re-schedule the operations.

In the RT–MILFP, a timed route is defined as a sequence of shifts with partially decided operations inside the whole planning horizon. Only the quantity loaded or delivered by the operations in the route is unspecified. One timed route has to be selected for each driver. Since the total number of timed routes is exponential, each of such variables is considered as a column in the master problem and is generated by a pricing sub-problem.

In the column generation algorithm, the Master Problem (MP) is defined as the linear relaxation of the RT–MILFP formulation with the complete route set $\mathcal{RO}$. The Restricted Master Problem (RMP) is the restriction of the MP to subset $\mathcal{RO}_1 \subset \mathcal{RO}$. The column generation approach begins with an RMP defined on subset $\mathcal{RO}_1$ containing only a few initial routes found by the greedy heuristics presented in Section 3.1. Then, it looks for beneficial columns with respect to the reduced cost by solving a pricing sub-problem. If such a column exists, it is added to the RMP and the RMP is solved again. Otherwise, the optimal solution for the RMP is also the optimal solution for the MP and RT–MILFP is solved to obtain an integer solution.
In our case, the master problem deals with the decisions of timed route selection and inventory management. The sub-problem manages dual values of the timed route selection variables (computed in the master problem). Its role is to generate beneficial new routes (in terms of reduced cost), while satisfying constraints concerning working time of drivers and opening time windows of customers. The sub-problem can be naturally decomposed, since a route can be decomposed as a sequence of shifts.

In the following, we first present the RT–MILFP extended formulation with different aggregation levels. Then, the pricing sub-problem is presented, as well as its further decomposition.

3.2.1. Mathematical formulation Let \( \mathcal{RO}_d \) denote the set of all possible timed routes for each driver \( d \in \mathcal{DR} \). Binary parameter \( u_{r,h}^{r,t} \) equals 1 if and only if route \( r \in \mathcal{RO}_d \) uses trailer \( t_l \in \mathcal{T} \) at hour \( h \in \mathcal{H} \) and parameter \( v_{r,h}^{r,i} \in \{0,1\} \) equals 1 if and only if route \( r \in \mathcal{RO}_d \) visits site \( i \in \mathcal{Z} \cup \mathcal{SO} \) at hour \( h \in \mathcal{H} \).

The RT–MILFP contains the following variables. Binary variable \( x_{r,d}^r \) is equal to 1 if and only if route \( r \in \mathcal{RO}_d \) is selected for driver \( d \in \mathcal{DR} \). Continuous variable \( q_{h,i}^{r,t} \) is the quantity delivered (or loaded) at hour \( h \in \mathcal{H} \) at site \( i \in \mathcal{Z} \cup \mathcal{SO} \) by trailer \( t_l \in \mathcal{T} \). As for the FS–MILFP, Dinkelbach’s algorithm is used to linearize objective (35), but here coefficient \( \alpha \) is assumed fixed.

\[
\min \sum_{d \in \mathcal{DR}} \sum_{r \in \mathcal{RO}_d} C_{r,d}^r x_{r,d}^r - \alpha \sum_{i \in \mathcal{Z} \cup \mathcal{SO}} \sum_{t_l \in \mathcal{T}} \sum_{h \in \mathcal{H}} q_{h,i}^{r,t} \tag{35}
\]

The constraints are presented by categories. The dual variables needed by the pricing sub-problem presented in Section 3.2.2 are introduced in the paragraphs below. They are marked on the left side of the constraints in square brackets.

Assignment of a driver/trailer pair to each route

\[
[\omega_d] - \sum_{r \in \mathcal{RO}_d} x_{r,d}^r \geq -1 \quad \forall d \in \mathcal{DR} \tag{36}
\]

\[
[\psi_{h,t_l}^r] - \sum_{d \in \mathcal{DR}} \sum_{r \in \mathcal{RO}_d} x_{r,d}^r u_{r,h}^{r,t} v_{r,h}^{r,i} \geq -1 \quad \forall t_l \in \mathcal{T}, \forall h \in \mathcal{H} \tag{37}
\]

Constraints (36) ensure that at most one hourly-timed route is assigned to each driver. Constraints (37) make sure that at most one trailer is used at each hour in a driver route. At the end of this section, we discuss how this constraint can be adjusted to different aggregation levels.

Quantity limits

\[
[X_{i,t_l}^h] \sum_{d \in \mathcal{DR}} \sum_{r \in \mathcal{RO}_d} x_{r,d}^r u_{r,t_l}^{r,i} v_{r,h}^{r,i} Q_{t_l} \geq -q_{i,t_l}^h \quad \forall i \in \mathcal{SO}, \forall t_l \in \mathcal{T}, \forall h \in \mathcal{H} \tag{38}
\]

\[
[X_{i,t_l}^h] \sum_{d \in \mathcal{DR}} \sum_{r \in \mathcal{RO}_d} x_{r,d}^r u_{r,t_l}^{r,i} v_{r,h}^{r,i} Q_{t_l} \geq q_{i,t_l}^h \quad \forall i \in \mathcal{Z}, \forall t_l \in \mathcal{T}, \forall h \in \mathcal{H} \tag{39}
\]

\[
[\phi_{i,t_l}^h] - \sum_{d \in \mathcal{DR}} \sum_{r \in \mathcal{RO}_d} x_{r,d}^r u_{r,t_l}^{r,i} v_{r,h}^{r,i} R_{i}^{min} \geq -q_{i,t_l}^h \quad \forall i \in \mathcal{Z}, \forall t_l \in \mathcal{T}, \forall h \in \mathcal{H} \tag{40}
\]
Constraints (38) and (39) ensure that the quantity loaded at a source or delivered to a customer never exceeds the trailer capacity. Note that due to aggregation of time units to hours, these two constraints do not guarantee feasibility of the delivery. For instance, in the case where a source and a customer are delivered during the same hour, these constraints might allow an empty trailer to visit a customer first before visiting a source. Constraints (40) make sure that the quantity delivered to a customer is at least the minimum delivery quantity required by the customer.

**Inventory of trailers**

\[
J^h_{tl} = J^{h-1}_{tl} - \sum_{i \in Z \cup SO} q^h_{i,tl} \quad \forall tl \in \mathcal{T}, \forall h \in \mathcal{H} \tag{41}
\]

\[
0 \leq J^h_{tl} \leq Q_{tl} \quad \forall tl \in \mathcal{T}, \forall h \in \mathcal{H} \tag{42}
\]

Constraints (41) are for the inventory balance of each trailer from one hour to the other. Constraints (42) make sure that the tank level in the trailer is never negative and never exceeds the trailer capacity. Time aggregation to hour makes these constraints an approximation of the actual trailer inventory balance constraints.

**Customer inventory levels or demands**

\[
I^h_i = I^{h-1}_i + \sum_{tl \in \mathcal{T}_i} q^h_{i,tl} - R^h_i \quad \forall i \in \mathcal{Z}_{vmi}, \forall h \in \mathcal{H} \tag{43}
\]

\[
\sum_{h = \text{od}} \sum_{tl \in \mathcal{T}} q^h_{i,tl} \geq f^{\text{od}} R^{\text{od}} \quad \forall i \in \mathcal{Z}_{ci}, \forall \text{od} \in \mathcal{OD} \tag{44}
\]

\[
\sum_{h = \text{od}} \sum_{tl \in \mathcal{T}} q^h_{i,tl} \leq R^{\text{od}} \quad \forall i \in \mathcal{Z}_{ci}, \forall \text{od} \in \mathcal{OD} \tag{45}
\]

Constraints (43) ensure the inventory balance of each VMI customer from one period to the next. Constraints (44) check whether the quantity delivered inside the time limits of an order satisfies the demand. Constraints (45) set limits on the quantity delivered to call-in customers.

**Variable domains**

\[
L_i \leq I^h_i \leq T_i \quad \forall i \in \mathcal{Z}_{vmi}, \forall h \in \mathcal{H} \tag{46}
\]

\[
0 \leq J^h_{tl} \leq Q_{tl} \quad \forall tl \in \mathcal{T}, \forall h \in \mathcal{H} \tag{47}
\]

\[
-Q_{tl} \leq q^h_{i,tl} \leq 0 \quad \forall i \in \mathcal{SO}, \forall tl \in \mathcal{T}, \forall h \in \mathcal{H} \tag{48}
\]

\[
0 \leq q^h_{i,tl} \leq T_i \quad \forall i \in \mathcal{Z}, \forall tl \in \mathcal{T}_i, \forall h \in \mathcal{H} \tag{49}
\]

\[
x^d_r \in \{0,1\} \quad \forall d \in \mathcal{DR}, \forall r \in \mathcal{RO}_d \tag{50}
\]

Constraints (46)—(50) define the variable domains.
Discussion on the aggregation level The time unit in RT–MILFP can be made (approximately) accurate to the minute by introducing parameter \( \mu_{r,h}^{t,l} \in [0,60] \), which is defined as the number of minutes that trailer \( tl \) is used in hour \( h \) on route \( r \). The aggregated trailer usage constraints (37) can then be rewritten as:

\[
- \sum_{d \in \mathcal{D}R} \sum_{r \in \mathcal{R}O_d} x_{d}^{r} \mu_{r,h}^{t,l} \geq -60 \quad \forall t,l \in \mathcal{T}L, \forall h \in \mathcal{H} 
\] (51)

The relation between parameters \( u \) and \( \mu \) can be:

\[
u_{r,h}^{t,l} = \begin{cases} 1 & \text{if } \mu_{r,h}^{t,l} > 0 \\ 0 & \text{otherwise} \end{cases} \quad \text{(Relation 1)}
\]

or

\[
u_{r,h}^{t,l} = \begin{cases} 1 & \text{if } \mu_{r,h}^{t,l} = 60 \\ 0 & \text{otherwise} \end{cases} \quad \text{(Relation 2)}
\]

In the RT–MILFP formulation, trailer inventory balance constraints are relaxed to an hourly level instead of minutes. Consequently, depending on whether Relation 1 or 2 is used, the formulation has different properties. If Relation 1 is used, then the RT–MILFP formulation is over-constrained in terms of trailer usage, because Relation 1 forbids one trailer to be shared by two or more shifts in the same hour, which should be allowed in reality as long as the shifts do not overlap in time. Therefore, the formulation is neither a relaxation nor an over-constrained formulation of the original Challenge problem. On the other hand, if Relation 2 is applied, then the trailer usage constraints are relaxed in the sense that a trailer is considered to be occupied in an hour only if it is fully used in this hour. In our experiments, Relation 1 is preferred.

3.2.2. Pricing sub-problem The dual problem contains dual variables \( \omega_d \in \mathbb{R}_+ \) for each driver \( d \in \mathcal{D}R \) associated with constraints (36), \( \psi_{i,l}^h \in \mathbb{R}_+ \) for each trailer \( tl \in T\mathcal{L} \) and for each hour \( h \in \mathcal{H} \) associated with constraints (37), \( \chi_{i,l}^h \in \mathbb{R}_+ \) for each source or customer site \( i \in \mathcal{Z} \cup \mathcal{SO} \) for each trailer \( tl \in T\mathcal{L} \) and for each hour \( h \in \mathcal{H} \) associated with constraints (38), and (39) and \( \phi_{i,l}^h \) for each customer site \( i \in \mathcal{Z} \) for each trailer \( tl \in T\mathcal{L} \) and for each hour \( h \in \mathcal{H} \) associated with constraints (40). The value of the dual variables can be considered as additional costs for the usage of drivers or trailers, or as benefits brought by operations to customers.

The constraints corresponding to variables \( x \) in the dual problem can then be written as:

\[
-\omega_d - \sum_{d \in \mathcal{D}R} \sum_{r \in \mathcal{R}O_d} u_{r}^{t,l} \psi_{i,l}^h + \sum_{d \in \mathcal{D}R} \sum_{r \in \mathcal{R}O_d} \sum_{i \in \mathcal{Z} \cup \mathcal{SO}} u_{r}^{t,l} \nu_{r}^{t,l} \nu_{i}^{r,h} (Q_{i} \chi_{i,l}^h - R_{i}^{\min} \phi_{i,l}^h) - C_{r,d}^{\text{route}} \leq 0 \\
\forall r \in \mathcal{R}O_d, \forall d \in \mathcal{D}R
\]
with $C_{r,d}$ the total cost (of distance, time and layover) of all the shifts in route $r$ for driver $d$. Assuming $\phi^h_{i,t,l} = 0$ for each source $i \in SO$, the combined term $Q_{i,t,l} \chi^h_{i,t,l} - R_{i}^{\min} \phi^h_{i,t,l}$ represents the total benefit to visit each site $i \in Z \cup SO$ by trailer $tl \in TL$ in each hour $h \in H$.

Let

$$f(r) = - \sum_{t \in T} \sum_{h \in H} u_{r,h}^{r,h} \psi_{t,l}^{r,h} + \sum_{t \in T} \sum_{h \in H} \sum_{i \in Z \cup SO} v_{r,h}^{r,h} \psi_{i}^{r,h} (Q_{i,t,l} \psi_{t,l}^{r,h} - R_{i}^{\min} \phi^h_{i,t,l}) - C_{r,d}^{route}$$

(52)

Finding a new column amounts to finding the route $r^*$ for driver $d$ such that $f(r^*)$ is strictly larger than $\omega_d$. Mathematically, this is equivalent to finding route $r^*$ for driver $d$ such that

$$r^* \in \arg\max_{r \in RO} f(r)$$

$$f(r^*) > \omega_d$$

The parameters $u_{i,t}^{r,h}$ and $v_{i}^{r,h}$ used in the master problem become two sets of decision variables for a certain route $r$ in the sub-problem. Binary variable $u_{i,t}^{r,h}$ equals 1 if trailer $tl$ is used in route $r$ at hour $h$. Binary variable $v_{i}^{r,h}$ for each customer or source site $i \in Z \cup SO$ at each hour $h \in H$ equals 1 if site $i$ is visited at hour $h$ in route $r$.

The objective is to maximize the profit minus the cost of the route defined by $f(r)$ in (52). It is non-linear with respect to variables $u_{i,t}^{r,h}$ and $v_{i}^{r,h}$. Constraints related to driver/trailer assignments and timing of delivery/loading activities need to be satisfied in the subproblem. Namely,

- the trailers used in the route can all be driven by driver $d$;
- the driving duration of $d$ is respected;
- driver $d$ only works in his time windows;
- each visit to customer $i \in Z$ happens in one of the customer’s time windows;
- each customer $i \in Z$ is visited by compatible trailer $tl \in TL_i$.

Since a timed route can be defined as a sequence of timed shifts, the sub-problem can be further decomposed. Promising timed shifts are discovered and combined to form a complete route. In this way, the problem of finding promising timed shifts is equivalent to a Shortest Path Problem with Resource Constraints (SPPRC) in a time-space graph with time as a resource. It can be solved with a labeling algorithm adapted from Irnich and Desaulniers (2005). The problem of combining shifts to form a route can be considered as a Shortest Path Problem (SPP) in a Directed Acyclic Graph (DAG) of shifts. It can thus be solved in $O(|TW_d|S)$ time, with $|TW_d|$ the number of time windows of the driver $d$ and $S$ the number of most profitable shifts generated in each time window.

In our implementation, in each iteration of the column generation, the maximum number of promising shifts generated for each compatible pair of driver and trailer is 200 and at most one route is added for each driver. The column generation stops if no route is found, the total number of iterations is more than 1000 or the predefined time limit is exceeded.
4. Experimentation and Discussion

In this section, we present the experimentation and the results with some discussions about the effectiveness of solving FS-MILFP as post-optimization.

4.1. Instances and General Settings

The experimentation is based on the instance sets B and X from the Challenge. Also, additional smaller instances have been generated to better illustrate the properties of our algorithm.

Let us first analyse the instances B and X (Table 1). The number of customers $|Z|$ varies from 32 to 324. The number of call-in customers and layover customers is given by $|Z_{ci}|$ and $|Z_{lo}|$, respectively. The size of the horizon $|H|$ varies from 10 to 35 days (from 240 to 840 hours). The number of drivers $|DR|$ varies from 4 to 13 and the number of trailers $|TL|$ is from 3 to 15. There are at most 2 sources. According to the number of customers, the instances can be categorized into 5 types (called “map” in the following part). Each map corresponds to a set of sites with identical (or nearly identical for instances X) distance and time matrices. The column “Map” in Table 1 identifies the type of the instances listed in the column “Instances”. All of these numbers go beyond existing benchmarks proposed by Archetti et al. (2007) and Coelho et al. (2012).

| Map | Instances               | $|Z|$ | $|Z_{ci}|$ | $|Z_{lo}|$ | $|SO|$ | $|DR|$ | $|TL|$ | max $H$ |
|-----|-------------------------|------|----------|----------|-------|-------|-------|---------|
| I   | V2.24, V2.25, V2.26     | 32   | 9        | 0        | 2     | 5     | 6     | 840     |
| II  | V2.13, V2.14, V2.19     | 53   | 0        | 14       | 1     | 5     | 5     | 840     |
| III | V2.15, V2.17, V2.18, X3 | 134  | 3        | 16       | 1     | 4     | 3     | 840     |
| IV  | V2.16, V2.20, V2.21, X2 | 184  | 1        | 5        | 1     | 7     | 4     | 840     |
| V   | V2.12, V2.22, V2.23, X1, X4, X5 | 324  | 23       | 12       | 1     | 13    | 15    | 504     |

Additional smaller instances for each type of map are generated using the following parameters. The size of the horizon ranges from 48 hours (2 days) to 240 hours (10 days) with a step of 24 hours. The numbers of customers remain the same as in the initial instances (from 32 to 324 according to the corresponding map). Five instances are generated for each value of the size of the horizon for a total of 45 instances. These instances form the set of instances II.

Finally, to test the influence of the number of customers, we have generated instances with a fixed horizon of 360 hours (15 days) and the number of customers varying from 5 to 50 with a step of 5. The customers are randomly chosen from the initial instances. There are 5 instances for each category with the number of customers between 5 and 30. There are 4 instances with 35 customers and 3 instances for each type of map with 40, 45 or 50 customers. In total, 43 instances are generated. They form the set of instances C.
The mathematical models are coded in C++ and solved by Cplex 12.7.0 with one single thread. All the tests were performed on a computing platform composed of Xeon E5-2695 v4 CPU and 16 Gb memory.

4.2. Effectiveness of FS-MILFP on sets H and C

FS-MILFP is solved using initial solutions returned by the two randomized greedy heuristics after 30 minutes. Tables 2 and 3 report the performance of the FS-MILFP re-optimization on instances H and C, respectively. In these two tables, the column “Map” indicates the type of each instance. Columns “|Z|” and “|H|” indicate the number of customers and the length of time horizon in each instance. Columns “LR”, “TSC” and “TDQ” report the value of the logistic ratio, the total shift cost and the total delivered quantity of each solution. Values indexed by 0 are obtained by the heuristics only and those indexed by 1 are obtained after the FS-MILFP re-optimization. When no feasible solution has been found, the corresponding logistic ratio, shift cost and delivery quantity are marked by “-”. Columns under the label “Gaps” show improvements of solution values obtained after solving FS-MILFP. The gap is computed by formula (53) and expressed in percentage. Columns “Gap(LR)”, “Gap(TSC)”, “Gap(TDQ)” indicate the gaps for the logistic ratio, the total shift cost and the total delivered quantity, respectively. Column “\(\overline{Gap}(LR)\)” is the average gap of the logistic ratio over all the instances in the same category. If the initial solution of an instance is not feasible, but after the FS-MILFP re-optimization, a feasible solution is found, then “*” is marked in the gap columns.

\[
\text{Gap}(x) = \frac{x_1 - x_0}{x_0} (\%)
\] (53)

For both instance sets H and C, the FS-MILFP component can largely improve the solutions of the heuristics. The average reduction rate of the logistic ratio brought by FS-MILFP over all instances H is 11.62%, which results from 7.02% decrease of the total shift cost and 2.91% increase of the total delivery quantity. In particular, the algorithm for solving FS-MILFP is able to repair two instances solved by the heuristics.

For instances C, under 360 hours’ time horizon, the FS-MILFP can be solved with instances of 50 customers and it can also repair some of the initial infeasible solutions. The average improvement of the logistic ratio is 15.52% which results from 5.98% decrease of total shift cost and 8.15% increase of the total delivery quantity.

4.3. Effectiveness of FS-MILFP on the Challenge instances

To test the limit of the FS-MILFP, we apply the FS-MILFP to the instances of the Challenge and we use as initial solutions the results obtained by the randomized greedy heuristics after 30
The results show that the FS-MILFP reoptimization is effective and can be applied to any solution methods that provide feasible sequences of operations. Table 4 is the improvement after solving FS-MILFP using the initial solutions given by our randomized heuristics. Table 5 shows the improvement of the best know upper bound obtained by Ahmed Kheiri (Cardiff University, U.K.), the winner of the Challenge. Table 6 shows the improvement of the solutions obtained by the team.
Table 3  Re-optimization by the FS-MILFP on Instances C with initial solutions of 30 min of randomized greedy heuristics

<table>
<thead>
<tr>
<th>Map</th>
<th></th>
<th>Set</th>
<th>greedy heuristics</th>
<th>re-optimization FS-MILFP</th>
<th>Gaps</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>LR&lt;sub&gt;0&lt;/sub&gt;</td>
<td>TSC&lt;sub&gt;0&lt;/sub&gt;</td>
<td>TDQ&lt;sub&gt;0&lt;/sub&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>5  300</td>
<td>0.071938</td>
<td>427.25</td>
<td>5800.17</td>
<td>0.071656</td>
</tr>
<tr>
<td>II</td>
<td>5  300</td>
<td>0.028683</td>
<td>942.60</td>
<td>39431.70</td>
<td>0.024109</td>
</tr>
<tr>
<td>III</td>
<td>5  300</td>
<td>0.047094</td>
<td>398.70</td>
<td>38566.08</td>
<td>0.046143</td>
</tr>
<tr>
<td>IV</td>
<td>5  300</td>
<td>0.101030</td>
<td>551.90</td>
<td>5462.76</td>
<td>0.092756</td>
</tr>
<tr>
<td>V</td>
<td>5  300</td>
<td>0.044637</td>
<td>557.10</td>
<td>12480.68</td>
<td>0.036819</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>10 300</td>
<td>0.021147</td>
<td>3810.42</td>
<td>177665.96</td>
<td>0.019487</td>
</tr>
<tr>
<td>II</td>
<td>10 300</td>
<td>0.025840</td>
<td>1429.50</td>
<td>55320.32</td>
<td>0.021702</td>
</tr>
<tr>
<td>III</td>
<td>10 300</td>
<td>0.029864</td>
<td>1339.10</td>
<td>44839.75</td>
<td>0.024568</td>
</tr>
<tr>
<td>IV</td>
<td>10 300</td>
<td>0.039555</td>
<td>6471.09</td>
<td>209050.20</td>
<td>0.024670</td>
</tr>
<tr>
<td>V</td>
<td>10 300</td>
<td>0.037180</td>
<td>4505.90</td>
<td>121192.29</td>
<td>0.030804</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>20 300</td>
<td>0.046598</td>
<td>8061.10</td>
<td>172990.69</td>
<td>0.039237</td>
</tr>
<tr>
<td>II</td>
<td>20 300</td>
<td>0.030914</td>
<td>2711.00</td>
<td>90078.05</td>
<td>0.026797</td>
</tr>
<tr>
<td>III</td>
<td>20 300</td>
<td>0.031425</td>
<td>6392.04</td>
<td>203403.62</td>
<td>0.024670</td>
</tr>
<tr>
<td>IV</td>
<td>20 300</td>
<td>0.046598</td>
<td>8061.10</td>
<td>172990.69</td>
<td>0.039237</td>
</tr>
<tr>
<td>V</td>
<td>20 300</td>
<td>0.030914</td>
<td>2711.00</td>
<td>90078.05</td>
<td>0.026797</td>
</tr>
</tbody>
</table>

He et al. A Matheuristic for a Real-life IRP

22 Article submitted to Transportation Science; manuscript no. TS-2017-0278.R2

who wins the scientific prize by Nabil Absi, Diego Cattaruzza, Dominique Feillet, Frédéric Semet, Maxime Ogier (Ecole des Mines Saint Etienne, Ecole Centrale Lille, France). In these tables, the instances are categorized by the map type. Columns labelled by “LR”, “TSC”, and “TDQ” are for values of the logistic ratio, the total shift cost and the total delivery quantity, as before. Values labelled by 0 are in the initial solution and values labelled by 1 are obtained by the FS-MILFP reoptimization.

We set a 30 minutes time limit for the solution of the FS-MILFP model. For most of the instances, the time spent for re-optimization is less than 400 seconds. On average, the FS-MILFP re-optimization is able to improve the logistic ratio of our randomized heuristics by 6.78%. The average improvement of the solutions obtained by the scientific prize team is 3.49%. The average improvement of the best upper bound is 1.95%. The improvement brought by FS-MILFP is larger for instances of map V having a larger number of customers. This could be explained by the fact
Table 4: Improvement of the solution of 30 min randomized heuristics by the FS-MILFP re-optimization on instances of the Challenge

<table>
<thead>
<tr>
<th>Map</th>
<th>LR</th>
<th>TSC</th>
<th>TDQ</th>
<th>Gaps</th>
<th>time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>0.01797</td>
<td>9386.07</td>
<td>543214.28721764</td>
<td>-9.18%</td>
<td>-6.53%</td>
</tr>
<tr>
<td>II</td>
<td>0.05444</td>
<td>62404.48</td>
<td>1148330.70555505</td>
<td>0.0%</td>
<td>0.0%</td>
</tr>
<tr>
<td>III</td>
<td>0.06052</td>
<td>8214.8</td>
<td>136795.351374</td>
<td>-4.73%</td>
<td>-3.60%</td>
</tr>
<tr>
<td>IV</td>
<td>0.024898</td>
<td>10048.17</td>
<td>403573.64368827</td>
<td>-4.36%</td>
<td>-4.11%</td>
</tr>
<tr>
<td>V</td>
<td>0.024902</td>
<td>10049.77</td>
<td>403573.64368838</td>
<td>-4.36%</td>
<td>-4.11%</td>
</tr>
</tbody>
</table>

Average gap: -6.78% -5.79% 1.05%

that the methods proposed during the challenge have difficulties in managing continuous quantities such as the driver working times and the inventory levels. From Tables 4–6, one can also see that the influence of the instance type to the gap of the logistic ratio is not very obvious. For some small instances on map I or II, the solution of FS-MILFP is not able to converge to optimality after 30 minutes. By analyzing the solution files, we found that the improvement of the logistic ratio is indeed mainly due to adjustments of arrival times and delivery quantities. The layover pauses are also successfully re-positioned by FT-MILFP such that the overall performance of each shift is improved.

4.4. Complete decomposition method and performance of the RT-MILFP on sets H and C

In the Challenge context, the complete decomposition method was able to obtain 13 feasible solutions out of 20 instances. The infeasibility is mainly due to the fact that the greedy heuristics could not always find a feasible solution and that RT-MILFP does not scale well on large Challenge instances. Nonetheless, our method was ranked 6 out of the 9 finalists, obtaining slightly better results than the winners of the scientific prize. In what follows, we present the results of the complete matheuristic on smaller instance sets, which illustrate the potential of RT-MILFP.

Tables 7 and 8 report the improvements brought by RT-MILFP on Instances H and C. In these experiments, we start with the solutions of the combined randomized heuristics, which have been re-optimized by FS-MILFP, and show how much RT-MILFP can improve them (after a final re-optimization). As shown by Tables 7 and 8, RT-MILFP is also able to find solutions of good quality, especially for smaller instances. For example, for instance H on map II with 48 hours horizon, the
Table 5 Improvement of the best known upper bound

<table>
<thead>
<tr>
<th>Map</th>
<th>inst</th>
<th>best known upper bound</th>
<th>reoptimized solution by FS-MILFP</th>
<th>Gap</th>
<th>time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>LR_i</td>
<td>TSC_i</td>
<td>TDQ_i</td>
<td>LR_i</td>
</tr>
<tr>
<td>I</td>
<td>2.24</td>
<td>0.013033</td>
<td>8016.35</td>
<td>615067.03590813</td>
<td>0.012523</td>
</tr>
<tr>
<td></td>
<td>2.25</td>
<td>0.014177</td>
<td>8143.82</td>
<td>574523.94015215</td>
<td>0.013847</td>
</tr>
<tr>
<td></td>
<td>2.26</td>
<td>0.015885</td>
<td>8197.32</td>
<td>58740.14040184</td>
<td>0.015266</td>
</tr>
<tr>
<td>II</td>
<td>2.13</td>
<td>0.026608</td>
<td>7807.36</td>
<td>264611.47692655</td>
<td>0.025894</td>
</tr>
<tr>
<td></td>
<td>2.14</td>
<td>0.037582</td>
<td>28449.60</td>
<td>757005.95803728</td>
<td>0.036996</td>
</tr>
<tr>
<td></td>
<td>2.19</td>
<td>0.036018</td>
<td>26476.70</td>
<td>734388.07316117</td>
<td>0.035183</td>
</tr>
<tr>
<td>III</td>
<td>2.15</td>
<td>0.027508</td>
<td>7807.36</td>
<td>264611.47692655</td>
<td>0.026891</td>
</tr>
<tr>
<td></td>
<td>2.16</td>
<td>0.030183</td>
<td>28449.60</td>
<td>757005.95803728</td>
<td>0.029497</td>
</tr>
<tr>
<td></td>
<td>2.17</td>
<td>0.038107</td>
<td>27510.50</td>
<td>833282.95793682</td>
<td>0.037400</td>
</tr>
<tr>
<td>X3</td>
<td>2.19</td>
<td>0.039505</td>
<td>27435.46</td>
<td>84997.47486218</td>
<td>0.038798</td>
</tr>
</tbody>
</table>

Table 6 Improvement of the solutions of the scientific prize winner

<table>
<thead>
<tr>
<th>Map</th>
<th>inst</th>
<th>initial solution</th>
<th>reoptimized solution by FS-MILFP</th>
<th>Gap</th>
<th>time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>LR_i</td>
<td>TSC_i</td>
<td>TDQ_i</td>
<td>LR_i</td>
</tr>
<tr>
<td>I</td>
<td>2.12</td>
<td>0.012066</td>
<td>22938.88</td>
<td>81497.20</td>
<td>585039.64024105</td>
</tr>
<tr>
<td></td>
<td>2.13</td>
<td>0.034830</td>
<td>8536.40</td>
<td>24526.572611</td>
<td>254021.62153</td>
</tr>
<tr>
<td></td>
<td>2.14</td>
<td>0.046736</td>
<td>34200.90</td>
<td>736004.36372894</td>
<td>0.045963</td>
</tr>
<tr>
<td></td>
<td>2.15</td>
<td>0.031311</td>
<td>22213.93</td>
<td>467428.480928</td>
<td>0.030700</td>
</tr>
<tr>
<td></td>
<td>2.16</td>
<td>0.031241</td>
<td>7236.53</td>
<td>587401.46022478</td>
<td>0.031241</td>
</tr>
<tr>
<td>II</td>
<td>2.21</td>
<td>0.012992</td>
<td>7807.36</td>
<td>264611.47692655</td>
<td>0.012566</td>
</tr>
<tr>
<td></td>
<td>2.22</td>
<td>0.016826</td>
<td>10368.04</td>
<td>616206.44739343</td>
<td>0.016288</td>
</tr>
<tr>
<td></td>
<td>2.23</td>
<td>0.014177</td>
<td>8143.82</td>
<td>574523.94015215</td>
<td>0.013847</td>
</tr>
<tr>
<td></td>
<td>2.24</td>
<td>0.015885</td>
<td>8197.32</td>
<td>58740.14040184</td>
<td>0.015266</td>
</tr>
<tr>
<td>III</td>
<td>2.25</td>
<td>0.026608</td>
<td>7807.36</td>
<td>264611.47692655</td>
<td>0.025894</td>
</tr>
<tr>
<td></td>
<td>2.26</td>
<td>0.029497</td>
<td>8181.74</td>
<td>848513.99351407</td>
<td>0.028798</td>
</tr>
<tr>
<td></td>
<td>2.27</td>
<td>0.030183</td>
<td>28449.60</td>
<td>757005.95803728</td>
<td>0.029497</td>
</tr>
<tr>
<td></td>
<td>2.28</td>
<td>0.038107</td>
<td>27510.50</td>
<td>833282.95793682</td>
<td>0.037400</td>
</tr>
<tr>
<td>X3</td>
<td>2.29</td>
<td>0.039505</td>
<td>27435.46</td>
<td>84997.47486218</td>
<td>0.038798</td>
</tr>
</tbody>
</table>

5. Conclusion and Perspectives

In this paper, a real-life IRP proposed by Air Liquide is presented. This problem includes features such as the planning of driver activities in continuous time, different levels of time discretisation, continuous management of trailer quantity and the non-linear objective of the logistic ratio, together with other business related constraints. It becomes much more complicated than clas-
This matheuristic method is very efficient for the solution of instances of IRPs studied in the literature. A matheuristic method is proposed to solve this problem. In particular, a fixed-sequence sub-problem denoted FSCIRP is identified and an FS-MILFP model is proposed to solve it with an algorithm for dealing with the non-linear objective. Given a fixed sequence, the FS-MILFP checks whether there exists a feasible planning of delivery or not. Given a feasible sequence, this model is experimentally proved efficient for re-optimizing the timing and quantity of the operations. It was even able to improve the best solutions obtained so far during the Challenge. Greedy heuristics and a mathematical model with column generation are also proposed for sequence generation. This matheuristic method is very efficient for the solution of instances.
with a horizon shorter than 240 hours (10 days) and with the number of customers smaller than 35. Even though this method does not scale-up very well with large-size Challenge instances, it inspires future research activities.

FSCIRP is worth further studying. First, its complexity is still undetermined. Second, it could be interesting to determine, given a fixed feasible sequence of shifts, the best logistic-ratio improvement that can be expected. Moreover, the reason of the slow convergence of the logistic ratio in practice should be analyzed more in details. Other algorithms for accelerating the convergence of the fractional objective need to be developed.
The RT-MILFP model can be a start for many studies. First, other decomposition methods with aggregation could be interesting to study. For instance, using dynamic solution techniques such as rolling horizon or customer clustering to smaller delivery regions might be a good way to improve the effectiveness of a column generation approach with mixed integer programming. The relation between the linearized objective and the logistic ratio could also be further studied. In the column generation scheme, coefficient $\alpha$ can influence dual values of route selection variables and hence influence the construction of new routes. An algorithm combining column generation and non-linear fractional programming could thus be relevant. Moreover, to solve the integer master problem, the column generation scheme can be integrated into a Branch-and-Price method. Valid inequalities similar to those proposed in Desaulniers et al. (2016) might also be applied.

Since the logistic ratio implies the minimization of inventory routing costs and the maximization of delivery quantities, the problem can be studied under multiple objectives. Therefore, it is useful to look at the maximum quantities one can expect to deliver without increasing the costs. One can also find an efficient way to deliver, such that the minimum increase in shift costs brings the maximum increase in delivery quantities.

Acknowledgments

The authors would like to thank Mikaël Capelle for the technical support at an early stage of this work. The authors would also like to thank the members of two Challenge teams, winners of the best method and scientific prizes, for kindly providing us with the solutions obtained by their methods. Their help was really valuable to show that our re-optimization method is able to work on their solutions to exhibit more efficient solutions: Ahmed Kheiri, Nabil Absi, Diego Cattaruzza, Dominique Feillet, Maxime Ogier and Frédéric Semet. Finally, the authors would like to thank the two reviewers for their valuable suggestions which contributed to improve a previous version of this paper.

References


Archetti, Claudia, Leandro C. Coelho, Maria Grazia Speranza. 2018. *An Exact Algorithm for the Inventory Routing Problem with Logistic Ratio*. CIRRELT, Centre interuniversitaire de recherche sur les réseaux d’entreprise.


