A Matheuristic with Fixed-Sequence Reoptimization for a Real-Life Inventory Routing Problem

Yun He, Christian Artigues, Cyril Briand, Nicolas Jozefowiez, Sandra Ulrich Ngueveu

Submitted to *Transportation Science*
manuscript TS-2017-0278.R2

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 solutions 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 $\mathcal{H}$. A set of customers $\mathcal{Z}$ must be visited using a set of trailers $\mathcal{TL}$ driven by a set of drivers $\mathcal{DR}$. 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 $\mathcal{H}$, 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[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 $tl \in \mathcal{TL}$ is defined by initial loaded quantity J^0_{tl}, maximum capacity Q_{tl}, and per-distance-unit usage cost $C^{distance}_{tl}$.

2.2.2. Drivers Each driver $d \in \mathcal{DR}$ can only work during time windows belonging to set $\mathcal{TW}_d$. Each time window $tw \in \mathcal{TW}_d$ 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^{time}_d must be paid for each working minute of driver d. Driver d can only operate a subset of trailers denoted by $\mathcal{TL}_d \subseteq \mathcal{TL}$.

[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 $\mathcal{N}_s$ that starts and ends at the depot (with $|\mathcal{N}_s|$ 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 $\mathcal{N}_s(i)$. Notably, we have $\mathcal{N}_s(1) = \mathcal{N}_s(|\mathcal{N}_s|) = 0$.

Subset $\mathcal{SH}_d^{tw} \subseteq \mathcal{SH}$ 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 $\mathcal{SH}_d^{tw}(i)$. Similarly, subset $\mathcal{SH}^{tl} \subseteq \mathcal{SH}$ is the ordered set of shifts that is assigned to trailer $tl \in \mathcal{TL}$, with $\mathcal{SH}^{tl}(i)$ 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 $i,j \in \{0\} \bigcup \mathcal{Z} \bigcup \mathcal{SO}$, $D_{i,j}$ 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_i^0 and a forecast consumption of R_i^h 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 $[a_i^{tw}, b_i^{tw}]$, where a_i^{tw} and b_i^{tw} 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 $\underline{I}_i$ and maximum tank capacity $\overline{I}_i$. There is also a minimum delivery quantity R_i^{min} 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_i^{od}, delivery flexibility f_i^{od} given as a percentage, and delivery time window $[a^{od}, b^{od}]$ where starting time a^{od} and ending time b^{od} 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 $\mathcal{Z}_{lo}$. A layover customer can be either VMI or call-in, so we have $\mathcal{Z}_{lo} \subseteq \mathcal{Z}_{vmi} \cup \mathcal{Z}_{ci}$. 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 $d \in \mathcal{DR}$, 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 $\mathcal{SH}^l$.

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 $v_i^{s,h}$ is defined for each shift $s \in \mathcal{SH}$ at each hour $h \in \mathcal{H}$ and for each operation $i \in [1, |\mathcal{N}_s|]$. The variable is equal to 1 if and only if site $\mathcal{N}_s(i)$ is visited during hour h. Continuous variable t_i^s is defined for each operation $i \in [1, |\mathcal{N}_s|]$ in each shift $s \in \mathcal{SH}$. This variable is equal to the arrival time in minutes at site $\mathcal{N}_s(i)$. Continuous variable $z_i^s \in [0, 59]$ is defined for each operation $i \in [1, |\mathcal{N}_s|]$ in each shift $s \in \mathcal{SH}$. The variable is equal to the remaining time in minutes in the hour when site $\mathcal{N}_s(i)$ is visited. The variables related to arrival times at sites with their domains induced by the data is recapitulated in (1)–(3).

$$v_i^{s,h} \in \{0,1\} \qquad \forall s \in \mathcal{SH}, \forall i \in [1, |\mathcal{N}_s|], \forall h \in \mathcal{H} \tag{1}$$

$$t_i^s \in [A_d^{tw}, B_d^{tw}] \qquad \forall d \in \mathcal{DR}, \forall tw \in \mathcal{TW}_d, \forall s \in \mathcal{SH}^{tw}, \forall i \in [1, |\mathcal{N}_s|] \tag{2}$$

$$z_i^s \in [0, 59] \qquad \forall s \in \mathcal{SH}, \forall i \in [1, |\mathcal{N}_s|] \tag{3}$$

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

$$J_{tl}^{s,i} \in [0, Q_{tl}] \qquad \forall tl \in \mathcal{TL}, \forall s \in \mathcal{SH}^{tl}, \forall i \in [1, |\mathcal{N}_s|] \tag{4}$$

$$q_i^{s,h} \in \left[0, \overline{I}_{\mathcal{N}_s(i)}\right] \qquad \forall s \in \mathcal{SH}, \forall h \in \mathcal{H}, \forall i \in \{[1, |\mathcal{N}_s|] : \mathcal{N}_s(i) \in \mathcal{Z}_{vmi}\} \tag{5}$$

$$q_i^{s,h} \geq 0 \qquad \forall s \in \mathcal{SH}, \forall h \in \mathcal{H}, \forall i \in \{[1, |\mathcal{N}_s|] : \mathcal{N}_s(i) \in \mathcal{Z}_{ci}\} \tag{6}$$

$$q_i^{s,h} \leq 0 \qquad \forall s \in \mathcal{SH}, \forall h \in \mathcal{H}, \forall i \in \{[1, |\mathcal{N}_s|] : \mathcal{N}_s(i) \in \mathcal{SO}\} \tag{7}$$

$$I_j^h \in [\underline{I}_j, \overline{I}_j] \qquad \forall j \in \mathcal{Z}_{vmi}, \forall h \in \mathcal{H} \tag{8}$$

2.3.3. Layover pauses Binary variable l_i^s is defined for each operation $i \in [2, |\mathcal{N}_s|]$ in each shift $s \in \mathcal{SH}^l$. Recall that $\mathcal{SH}^l$ 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 $\mathcal{N}_s(i)$. 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 $p_i^s \in [0, 1]$ is defined for each operation $i \in [2, |\mathcal{N}_s|]$ of each shift $s \in \mathcal{SH}^l$. Let t_i^s be the arrival time at operation $i \in [2, |\mathcal{N}_s|]$ in shift $s \in \mathcal{SH}^l$ and λ^s the starting time of the layover pause if it is planned before operation i in shift s. Then, variable p_i^s is defined as $p_i^s = \frac{\lambda^s - t_{i-1}^s}{t_i^s - t_{i-1}^s}$.

The variable domains are the following:

$$l_i^s \in \{0, 1\} \qquad \forall s \in \mathcal{SH}^l, \forall i \in [2, |\mathcal{N}_s|] \tag{9}$$

$$p_i^s \in [0, 1] \qquad \forall s \in \mathcal{SH}^l, \forall i \in [2, |\mathcal{N}_s|] \tag{10}$$

2.4. Constraints

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

$$ST_{\mathcal{N}_s(i)} + T_{\mathcal{N}_s(i), \mathcal{N}_s(i+1)} + LOD_d l_{\mathcal{N}_s(i+1)}^s \leq t_{i+1}^s - t_i^s \qquad \forall d \in \mathcal{DR}, \forall s \in \mathcal{SH}^d, \forall i \in [1, |\mathcal{N}_s| - 1] \tag{11}$$

$$MIS_d \leq t_1^{\mathcal{SH}_d^{tw}(i+1)} - t_{|\mathcal{N}_{\mathcal{SH}_d^{tw}(i)}|}^{\mathcal{SH}_d^{tw}(i)} \qquad \forall d \in \mathcal{DR}, \forall tw \in \mathcal{TW}^d, \forall i \in [1, |\mathcal{SH}_d^{tw}| - 1] \tag{12}$$

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

$$t_i^s = 60 \sum_{h \in \mathcal{H}} h \ v_i^{s,h} + z_i^s \qquad \forall s \in \mathcal{SH}, \forall i \in [1, |\mathcal{N}_s|] \tag{13}$$

$$0 \leq z_i^s \leq 59 \qquad \forall s \in \mathcal{SH}, \forall i \in [1, |\mathcal{N}_s|] \tag{14}$$

$$\sum_{h \in \mathcal{H}} v_i^{s,h} = 1 \qquad \forall s \in \mathcal{SH}, \forall i \in [1, |\mathcal{N}_s|] \tag{15}$$

Constraints (13)–(14) ensure the coherence between binary variables v and continuous variables t for the scheduling of operations. If t_i^s is inside an hour h, then $v_i^{s,h}$ 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

$$a^{tw}_{\mathcal{N}_s(i)} \leq t^s_i \leq b^{tw}_{\mathcal{N}_s(i)} - ST_{\mathcal{N}_s(i)} \qquad \forall s \in \mathcal{SH}, \forall i \in \{[1, |\mathcal{N}_s|] : \mathcal{N}_s(i) \in \mathcal{Z}_{vmi}\} \tag{16}$$

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

2.4.4. Trailer capacity

$$J^{s,i+1}_{tl} - J^{s,i}_{tl} = -\sum_{h \in \mathcal{H}} q^{s,h}_i \qquad \forall tl \in \mathcal{TL}, \forall s \in \mathcal{SH}^{tl}, \forall i \in [1, |\mathcal{N}_s| - 1] \tag{17}$$

$$J^{\mathcal{SH}^{tl}(i+1),1}_{tl} = J^{\mathcal{SH}^{tl}(i),|\mathcal{N}_{\mathcal{SH}^{tl}(i)}|}_{tl} \qquad \forall tl \in \mathcal{TL}, \forall i \in [1, |\mathcal{SH}^{tl}| - 1] \tag{18}$$

$$-Q_{tl} \leq q^{s,h}_i \leq 0 \qquad \forall tl \in \mathcal{TL}, \forall s \in \mathcal{SH}^{tl}, \forall i \in \{[1, |\mathcal{N}_s|] : \mathcal{N}_s(i) \in \mathcal{SO}\} \tag{19}$$

$$0 \leq J^{s,i}_{tl} \leq Q_{tl} \qquad \forall tl \in \mathcal{TL}, \forall s \in \mathcal{SH}^{tl}, \forall i \in [1, |\mathcal{N}_s|] \tag{20}$$

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

$$I^{h+1}_j = I^h_j + \sum_{s \in \mathcal{SH}} \sum_{i \in \{[2, |\mathcal{N}_s|-1] : \mathcal{N}_s(i) = j\}} q^{s,h}_i - R^h_j \qquad \forall j \in \mathcal{Z}_{vmi}, \forall h \in \mathcal{H} \setminus \{H\} \tag{21}$$

$$R_{\mathcal{N}_s(i)} v^{s,h}_i \leq q^{s,h}_i \leq (\overline{I}_{\mathcal{N}_s(i)} - \underline{I}_{\mathcal{N}_s(i)}) v^{s,h}_i \qquad \forall s \in \mathcal{SH}, \forall h \in \mathcal{H}, \forall i \in \{[1, |\mathcal{N}_s|] : \mathcal{N}_s(i) \in \mathcal{Z}_{vmi}\} \tag{22}$$

$$\underline{I}_j \leq I^h_j \leq \overline{I}_j \qquad \forall j \in \mathcal{Z}_{vmi}, \forall h \in \mathcal{H} \tag{23}$$

$$R^{od}_j f^{od}_j \leq \sum_{s \in \mathcal{SH}} \sum_{i \in \{[2, |\mathcal{N}_s|-1] : \mathcal{N}_s(i) = j\}} \sum_{h \in \left[a^{od}, b^{od}\right]} q^{s,h}_i \leq R^{od}_j \qquad \forall j \in \mathcal{Z}_{ci}, \forall od \in \mathcal{OD}_j \tag{24}$$

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

$$\sum_{i=2}^{|\mathcal{N}_s|-1} l_i^s = 1 \qquad \forall s \in \mathcal{SH}^l \tag{25}$$

$$\sum_{i=2}^{|\mathcal{N}_s|-1} l_i^s = 0 \qquad \forall s \in \mathcal{SH} \backslash \mathcal{SH}^l \tag{26}$$

$$p_i^s \leq l_i^s \qquad \forall s \in \mathcal{SH}, \forall i \in [2, |\mathcal{N}_s|] \tag{27}$$

$$\sum_{j=i}^{|\mathcal{N}_s|} \sum_{k=j}^{|\mathcal{N}_s|} T_{\mathcal{N}_s(j-2),\mathcal{N}_s(j-1)} \ l_k^s + T_{\mathcal{N}_s(i-1),\mathcal{N}_s(i)} \ p_i^s \leq MDD_{d_s} \qquad \forall s \in \mathcal{SH}, \forall i \in [3, |\mathcal{N}_s|] \tag{28}$$

$$\sum_{j=2}^{|\mathcal{N}_s|} T_{\mathcal{N}_s(j-1),\mathcal{N}_s(j)} - (\sum_{j=i}^{|\mathcal{N}_s|} \sum_{k=j}^{|\mathcal{N}_s|} T_{\mathcal{N}_s(j-2),\mathcal{N}_s(j-1)} \ l_k^s + T_{\mathcal{N}_s(i-1),\mathcal{N}_s(i)} \ p_i^s) \leq MDD_{d_s} \qquad \forall s \in \mathcal{SH}, \forall i \in [3, |\mathcal{N}_s|] \tag{29}$$

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 $\mathscr{LR}$ is defined in Equation (30) as the total cost of all the shifts divided by the total quantity delivered in these shifts.

The distribution costs related to each shift include: the distance cost $\mathscr{D}$ induced by the usage of the trailers, the time cost $\mathscr{T}$ related to the total duration of the shift induced by drivers' working hours and the layover cost $\mathscr{L}$ if the shift contains a layover pause.

$$\mathscr{LR} = \frac{\mathscr{T} + \mathscr{D} + \mathscr{L}}{\mathscr{Q}} \tag{30}$$

The detailed computation of theses costs are given by Equations (31)–(33). The total quantity $\mathscr{Q}$ delivered by all shifts is the sum of the quantities in each delivery operation of each shift. It is given by Equation (34).

$$\mathscr{T} = \sum_{d \in \mathcal{DR}} \sum_{tw \in \mathcal{TW}_d} \sum_{s \in \mathcal{SH}^{tw}} C_d^{time} (t^s_{\mathcal{N}_s(|\mathcal{N}_s|)} - t_1^s - \sum_{i=2}^{|\mathcal{N}_s|-1} l_i^s LOD_d) \tag{31}$$

$$\mathscr{D} = \sum_{tl \in \mathcal{TL}} \sum_{s \in \mathcal{SH}^{tl}} \sum_{i=1}^{|\mathcal{N}_s|-1} C_{tl}^{distance} D_{\mathcal{N}_s(i), \mathcal{N}_s(i+1)} \tag{32}$$

$$\mathscr{L} = \sum_{d \in \mathcal{DR}} \sum_{tw \in \mathcal{TW}_d} \sum_{s \in \mathcal{SH}^{tw}} \sum_{i=2}^{|\mathcal{N}_s|-1} l_i^s C_d^{layover} \tag{33}$$

$$\mathscr{Q} = \sum_{s \in \mathcal{SH}} \sum_{h \in \mathcal{H}} \sum_{i \in \{[2, |\mathcal{N}_s|-1] : \mathcal{N}_s(i) \in \mathcal{Z}\}} q_i^{s,h} \tag{34}$$

Note that since the sequences of operations are assumed known, the total distance value $\mathscr{D}$ 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

$$\mathbf{t} = (t_i^s)_{i \in [1, |\mathcal{N}_s|]}^{s \in \mathcal{SH}}, \mathbf{l} = (l_i^s)_{i \in [2, |\mathcal{N}_s|]}^{s \in \mathcal{SH}}, \mathbf{q} = (q_i^{s,h})_{i \in \{[2, |\mathcal{N}_s|-1] : \mathcal{N}_s(i) \in \mathcal{Z}\}}^{s \in \mathcal{SH}, h \in \mathcal{H}}$$

in the feasible space $\mathbb{S}$ defined by the constraints above to minimize the fractional objective $\alpha = \frac{\mathscr{T}(\mathbf{t},\mathbf{l}) + \mathscr{D} + \mathscr{L}(\mathbf{l})}{\mathscr{Q}(\mathbf{q})}$ can be linearized to minimizing $(\mathscr{T} + \mathscr{D} + \mathscr{L}) - \alpha \mathscr{Q}$. 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 α, solves the resulting FS–MILFP, and adjusts the value of α 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.

Algorithm 1 Algorithm for fractional programming

1: Initialize α with a feasible logistic ratio $\mathscr{LR}_0$
2: Solve the FS-MILFP model with the linearised objective function
3: Recompute the ratio $\alpha' = \frac{\mathscr{T}+\mathscr{D}+\mathscr{L}}{\mathscr{Q}}$ with the current solution
4: **while** $\alpha' < \alpha - \epsilon$ **do** *(ϵ is a tolerance for optimality)*
5: $\quad \alpha \leftarrow \alpha'$
6: $\quad$ 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.

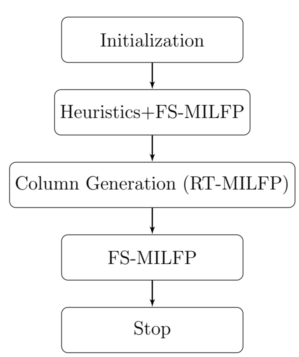

Figure 1 General solution method

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 λ_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.

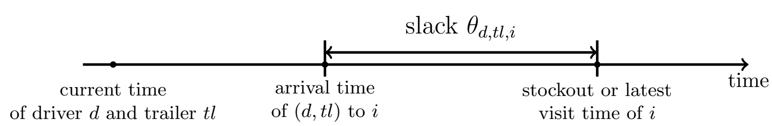

Figure 2 Definition of slack $\theta_{d,tl,i}$ for driver/trailer pair (d, tl) and customer i

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
10:     **else**
11:         remove i from L
12:     **if** i will not have stock out in the planning horizon **then**
13:         remove i from L
14:     update the state of the system
15:     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 $u_{tl}^{r,h}$ equals 1 if and only if route $r \in \mathcal{RO}_d$ uses trailer $tl \in \mathcal{TL}$ at hour $h \in \mathcal{H}$ and parameter $v_i^{r,h} \in \{0,1\}$ equals 1 if and only if route $r \in \mathcal{RO}_d$ visits site $i \in \mathcal{Z} \bigcup \mathcal{SO}$ at hour $h \in \mathcal{H}$.

The RT–MILFP contains the following variables. Binary variable x_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 $q_{i,tl}^h$ is the quantity delivered (or loaded) at hour $h \in \mathcal{H}$ at site $i \in \mathcal{Z} \cup \mathcal{SO}$ by trailer $tl \in \mathcal{TL}$. As for the FS–MILFP, Dinkelbach's algorithm is used to linearize objective (35), but here coefficient α is assumed fixed.

$$\min \quad \sum_{d \in \mathcal{DR}} \sum_{r \in \mathcal{RO}_d} C_{r,d}^{route} x_d^r - \alpha \sum_{i \in \mathcal{Z}} \sum_{tl \in \mathcal{TL}} \sum_{h \in \mathcal{H}} q_{i,tl}^h \tag{35}$$

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

$$[\omega_d] \qquad - \sum_{r \in \mathcal{RO}_d} x_d^r \geq -1 \qquad \forall d \in \mathcal{DR} \tag{36}$$

$$[\psi_{tl}^h] \qquad - \sum_{d \in \mathcal{DR}} \sum_{r \in \mathcal{RO}_d} x_d^r u_{tl}^{r,h} \geq -1 \qquad \forall tl \in \mathcal{TL}, \forall h \in \mathcal{H} \tag{37}$$

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

$$[\chi_{i,tl}^h] \qquad \sum_{d \in \mathcal{DR}} \sum_{r \in \mathcal{RO}_d} x_d^r u_{tl}^{r,h} v_i^{r,h} Q_{tl} \geq -q_{i,tl}^h \qquad \forall i \in \mathcal{SO}, \forall tl \in \mathcal{TL}, \forall h \in \mathcal{H} \tag{38}$$

$$[\chi_{i,tl}^h] \qquad \sum_{d \in \mathcal{DR}} \sum_{r \in \mathcal{RO}_d} x_d^r u_{tl}^{r,h} v_i^{r,h} Q_{tl} \geq q_{i,tl}^h \qquad \forall i \in \mathcal{Z}, \forall tl \in \mathcal{TL}, \forall h \in \mathcal{H} \tag{39}$$

$$[\phi_{i,tl}^h] \qquad - \sum_{d \in \mathcal{DR}} \sum_{r \in \mathcal{RO}_d} x_d^r u_{tl}^{r,h} v_i^{r,h} R_i^{min} \geq -q_{i,tl}^h \qquad \forall i \in \mathcal{Z}, \forall tl \in \mathcal{TL}, \forall h \in \mathcal{H} \tag{40}$$

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

$$J_{tl}^{h} = J_{tl}^{h-1} - \sum_{i \in \mathcal{Z} \bigcup \mathcal{SO}} q_{i,tl}^{h} \qquad \forall tl \in \mathcal{TL}, \forall h \in \mathcal{H} \tag{41}$$

$$0 \leq J_{tl}^{h} \leq Q_{tl}, \qquad \forall tl \in \mathcal{TL}, \forall h \in \mathcal{H} \tag{42}$$

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

$$I_{i}^{h} = I_{i}^{h-1} + \sum_{tl \in \mathcal{TL}_i} q_{i,tl}^{h} - R_{i}^{h}, \qquad \forall i \in \mathcal{Z}_{vmi}, \forall h \in \mathcal{H} \tag{43}$$

$$\sum_{h=a^{od}}^{b^{od}} \sum_{tl \in \mathcal{TL}} q_{i,tl}^{h} \geq f^{od} R^{od} \qquad \forall i \in \mathcal{Z}_{ci}, \forall od \in \mathcal{OD} \tag{44}$$

$$\sum_{h=a^{od}}^{b^{od}} \sum_{tl \in \mathcal{TL}} q_{i,tl}^{h} \leq R^{od} \qquad \forall i \in \mathcal{Z}_{ci}, \forall od \in \mathcal{OD} \tag{45}$$

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

$$\underline{I}_i \leq I_{i}^{h} \leq \overline{I}_i \qquad \forall i \in \mathcal{Z}_{vmi}, \forall h \in \mathcal{H} \tag{46}$$

$$0 \leq J_{tl}^{h} \leq Q_{tl} \qquad \forall tl \in \mathcal{TL}, \forall h \in \mathcal{H} \tag{47}$$

$$-Q_{tl} \leq q_{i,tl}^{h} \leq 0 \qquad \forall i \in \mathcal{SO}, \forall tl \in \mathcal{TL}_i, \forall h \in \mathcal{H} \tag{48}$$

$$0 \leq q_{i,tl}^{h} \leq \overline{I}_i \qquad \forall i \in \mathcal{Z}, \forall tl \in \mathcal{TL}_i, \forall h \in \mathcal{H} \tag{49}$$

$$x_{d}^{r} \in \{0,1\} \qquad \forall d \in \mathcal{DR}, \forall r \in \mathcal{RO}_d \tag{50}$$

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 $\mu_{tl}^{r,h} \in [0, 60]$, 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:

$$-\sum_{d \in \mathcal{DR}} \sum_{r \in \mathcal{RO}_d} x_d^r \mu_{tl}^{r,h} \geq -60 \qquad \forall tl \in \mathcal{TL}, \forall h \in \mathcal{H} \tag{51}$$

The relation between parameters u and μ can be:

$$u_{tl}^{r,h} = \begin{cases} 1 \text{ if } \mu_{tl}^{r,h} > 0 \\ 0 \text{ otherwise} \end{cases} \tag{Relation 1}$$

or

$$u_{tl}^{r,h} = \begin{cases} 1 \text{ if } \mu_{tl}^{r,h} = 60 \\ 0 \text{ otherwise} \end{cases} \tag{Relation 2}$$

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 $d \in \mathcal{DR}$ associated with constraints (36), $\psi_{tl}^h \in \mathbb{R}_+$ for each trailer $tl \in \mathcal{TL}$ and for each hour $h \in \mathcal{H}$ associated with constraints (37), $\chi_{i,tl}^h \in \mathbb{R}_+$ for each source or customer site $i \in \mathcal{Z} \bigcup \mathcal{SO}$ for each trailer $tl \in \mathcal{TL}$ and for each hour $h \in \mathcal{H}$ associated with constraints (38), and (39) and $\phi_{i,tl}^h$ for each customer site $i \in \mathcal{Z}$ for each trailer $tl \in \mathcal{TL}$ 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:

$$-\omega_d - \sum_{tl \in \mathcal{TL}} \sum_{h \in \mathcal{H}} u_{tl}^{r,h} \psi_{tl}^h + \sum_{tl \in \mathcal{TL}} \sum_{h \in \mathcal{H}} \sum_{i \in \mathcal{Z} \bigcup \mathcal{SO}} u_{tl}^{r,h} v_i^{r,h} (Q_{tl} \chi_{i,tl}^h - R_i^{min} \phi_{i,tl}^h) - C_{r,d}^{route} \leq 0$$
$$\forall r \in \mathcal{RO}_d, \forall d \in \mathcal{DR}$$

with $C_{r,d}^{route}$ the total cost (of distance, time and layover) of all the shifts in route r for driver d. Assuming $\phi_{i,tl}^{h} = 0$ for each source $i \in \mathcal{SO}$, the combined term $Q_{tl}\chi_{i,tl}^{h} - R_i^{min}\phi_{i,tl}^{h}$ represents the total benefit to visit each site $i \in \mathcal{Z} \bigcup \mathcal{SO}$ by trailer $tl \in \mathcal{TL}$ in each hour $h \in \mathcal{H}$.

Let

$$f(r) = -\sum_{tl\in\mathcal{TL}}\sum_{h\in\mathcal{H}} u_{tl}^{r,h}\psi_{tl}^{h} + \sum_{tl\in\mathcal{TL}}\sum_{h\in\mathcal{H}}\sum_{i\in\mathcal{Z}\bigcup\mathcal{SO}} u_{tl}^{r,h} v_i^{r,h}(Q_{tl}\chi_{i,tl}^{h} - R_i^{min}\phi_{i,tl}^{h}) - C_{r,d}^{route} \tag{52}$$

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

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

The parameters $u_{tl}^{r,h}$ 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 $u_{tl}^{r,h}$ 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 $i \in \mathcal{Z}\bigcup\mathcal{SO}$ at each hour $h \in \mathcal{H}$ 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 $u_{tl}^{r,h}$ 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 $i \in \mathcal{Z}$ happens in one of the customer's time windows;
- each customer $i \in \mathcal{Z}$ is visited by compatible trailer $tl \in \mathcal{TL}_i$.

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 $O(|\mathcal{TW}_d|S)$ time, with $|\mathcal{TW}_d|$ 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 $|\mathcal{Z}|$ varies from 32 to 324. The number of call-in customers and layover customers is given by $|\mathcal{Z}_{ci}|$ and $|\mathcal{Z}_{lo}|$, respectively. The size of the horizon $|\mathcal{H}|$ varies from 10 to 35 days (from 240 to 840 hours). The number of drivers $|\mathcal{DR}|$ varies from 4 to 13 and the number of trailers $|\mathcal{TL}|$ 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).

Table 1 Characteristics of instance sets B and X

Map	Instances	$\lvert\mathcal{Z}\rvert$	$\lvert\mathcal{Z}_{ci}\rvert$	$\lvert\mathcal{Z}_{lo}\rvert$	$\lvert\mathcal{SO}\rvert$	$\lvert\mathcal{DR}\rvert$	$\lvert\mathcal{TL}\rvert$	max $\mathcal{H}$
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 H.

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 "$|\mathcal{Z}|$" and "$|\mathcal{H}|$" 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 "$\overline{\text{Gap}}$(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.

$$\text{Gap}(x) = \frac{x_1 - x_0}{x_0}(\%) \tag{53}$$

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 coast 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

Table 2 Re-optimization by the FS-MILFP on Instances H with initial solutions of 30 min of randomized greedy heuristics

Map	$\|\mathcal{Z}\|$	$\|\mathcal{H}\|$	greedy heuristics			re-optimization FS-MILFP			Gaps			
			LR_0	TSC_0	TDQ_0	LR_1	TSC_1	TDQ_1	Gap(LR)	Gap(TSC)	Gap(TDQ)	$\overline{\text{Gap}}$(LR)
I	32	48	0.015926	1013.58	63645.02	0.015799	1013.58	64152.97	-0.80%	0.00%	0.80%	
II	36	48	0.098906	431.20	4359.71	0.070220	350.80	4995.71	-29.00%	-18.65%	14.59%	
III	118	48	0.013250	106.00	8000.00	0.013250	106.00	8000.00	0.00%	0.00%	0.00%	-27.37%
IV	179	48	0.027029	417.50	15446.42	0.026852	417.50	15548.22	-0.65%	0.00%	0.66%	
V	312	48	0.016996	3790.48	223019.28	0.015913	3610.88	226910.78	-6.37%	-4.74%	1.74%	
I	32	72	0.016448	1519.05	92352.97	0.015530	1519.05	97815.41	-5.58%	0.00%	5.91%	
II	36	72	0.048993	1230.50	25116.04	0.048916	1230.50	25155.28	-0.16%	0.00%	0.16%	
III	118	72	0.057257	1350.30	23583.32	0.053809	1344.70	24990.15	-6.02%	-0.41%	5.97%	-6.73%
IV	179	72	0.016960	981.00	57841.82	0.016460	981.00	59598.11	-2.95%	0.00%	3.04%	
V	312	72	0.019341	8381.48	433363.54	0.015682	7651.88	487946.33	-18.92%	-8.70%	12.60%	
I	32	96	0.015832	2607.84	164720.17	0.014795	2605.84	176135.37	-6.55%	-0.08%	6.93%	
II	36	96	0.037481	1419.20	37864.17	0.030279	1338.80	44215.77	-19.22%	-5.67%	16.77%	
III	118	96	0.041602	1551.40	37291.89	0.037742	1483.40	39303.99	-9.28%	-4.38%	5.40%	-11.89%
IV	179	96	0.021804	1651.80	75756.39	0.020452	1562.20	76383.36	-6.20%	-5.42%	0.83%	
V	312	96	0.020706	10434.67	503935.17	0.016933	8982.67	530496.74	-18.22%	-13.92%	5.27%	
I	32	120	-	-	-	-	-	-	-	-	-	
II	36	120	0.049108	2408.20	49038.52	0.045644	2408.20	52760.08	-7.05%	0.00%	7.59%	
III	118	120	0.039229	1935.30	49332.83	0.036292	1935.30	53325.52	-7.49%	0.00%	8.09%	-10.32%
IV	179	120	0.022460	3008.90	133964.89	0.019458	2653.30	136358.30	-13.37%	-11.82%	1.79%	
V	312	120	0.021174	13861.19	654626.30	0.018344	12742.39	694643.26	-13.37%	-8.07%	6.11%	
I	32	144	0.017568	5253.40	299031.71	0.016278	5229.40	321265.27	-7.34%	-0.46%	7.44%	
II	36	144	0.040484	2403.20	59362.02	0.035549	2352.20	66167.77	-12.19%	-2.12%	11.46%	
III	118	144	0.042293	3058.00	72304.30	0.040245	3058.00	75983.90	-4.84%	0.00%	5.09%	-10.37%
IV	179	144	0.022460	3008.90	133964.89	0.019458	2653.30	136358.30	-13.37%	-11.82%	1.79%	
V	312	144	0.021654	19087.54	881463.16	0.018594	17139.54	921795.62	-14.13%	-10.21%	4.58%	
I	32	168	0.018744	6822.22	363963.17	0.017143	6822.22	397950.81	-8.54%	0.00%	9.34%	
II	36	168	0.042915	3793.20	88388.92	0.036796	3741.50	101682.38	-14.26%	-1.36%	15.04%	
III	118	168	0.047242	4003.20	84737.57	0.046880	4003.20	85392.75	-0.77%	0.00%	0.77%	-9.35%
IV	179	168	0.023041	4039.87	175337.63	0.020357	3625.07	178076.06	-11.65%	-10.27%	1.56%	
V	312	168	0.023472	27575.40	1174799.47	0.020762	24709.30	1190094.78	-11.55%	-10.39%	1.30%	
I	32	192	-	-	-	0.016267	7595.53	466931.37	*	*	*	
II	36	192	0.048119	4101.40	85234.54	0.041343	4101.40	99205.41	-14.08%	0.00%	16.39%	
III	118	192	0.045484	4311.50	94791.66	0.044816	4253.90	94920.16	-1.47%	-1.34%	0.14%	-9.16%
IV	179	192	0.022286	5461.17	245049.01	0.020379	5043.57	247491.69	-8.56%	-7.65%	1.00%	
V	312	192	0.022840	30954.57	1355282.97	0.019975	27543.07	1378887.12	-12.54%	-11.02%	1.74%	
I	32	216	0.019666	9581.09	487199.83	0.017209	9215.49	535514.28	-12.49%	-3.82%	9.92%	
II	36	216	0.042419	5197.70	122531.16	0.038402	5197.70	135350.05	-9.47%	0.00%	10.46%	
III	118	216	0.047733	5563.70	116558.95	0.047664	5563.70	116727.92	-0.14%	0.00%	0.14%	-8.66%
IV	179	216	0.022540	6706.27	297526.05	0.020785	6267.47	301538.60	-7.79%	-6.54%	1.35%	
V	312	216	0.021940	35028.80	1596578.46	0.019004	30594.63	1609944.00	-13.38%	-12.66%	0.84%	
I	32	240	-	-	-	0.016530	9578.29	579434.16	*	*	*	
II	36	240	0.040642	6303.90	155109.52	0.034466	6158.40	178681.11	-15.20%	-2.31%	15.20%	
III	118	240	0.048196	6664.70	138283.83	0.047726	6612.30	138548.01	-0.98%	-0.79%	0.19%	-9.60%
IV	179	240	0.023689	9033.67	381343.93	0.021776	8340.87	383037.34	-8.08%	-7.67%	0.44%	
V	312	240	0.021994	39236.84	1783940.27	0.018884	34335.97	1818282.65	-14.14%	-12.49%	1.93%	
								average gap	-11.62%	-7.02%	2.91%	

minutes. We also use the solutions returned by the algorithms of the winner of the Challenge and the winners of the scientific prize.

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

Map	$\lvert\mathcal{Z}\rvert$	$\lvert\mathcal{H}\rvert$	greedy heuristics			re-optimization FS-MILFP			Gaps			
			LR_0	TSC_0	TDQ_0	LR_1	TSC_1	TDQ_1	Gap(LR)	Gap(TSC)	Gap(TDQ)	$\overline{\text{Gap}}$(LR)
I	5	360	0.071938	417.25	5800.17	0.071656	417.25	5822.98	-0.39%	0.00%	0.39%	
II	5	360	0.026883	952.50	35431.70	0.024109	952.50	39507.28	-11.51%	0.00%	11.50%	
III	5	360	0.047094	398.70	8466.08	0.046143	398.70	8640.58	-2.06%	0.00%	2.06%	-8.82%
IV	5	360	0.101030	551.90	5462.76	0.092756	518.80	5593.18	-8.92%	-6.00%	2.39%	
V	5	360	0.044637	557.10	12480.68	0.036819	460.20	12498.90	-21.23%	-17.39%	0.15%	
I	10	360	0.021447	3810.42	177665.96	0.019487	3785.66	194266.98	-10.06%	-0.65%	9.34%	
II	10	360	0.025840	1429.50	55320.32	0.021702	1429.50	65870.50	-19.07%	0.00%	19.07%	
III	10	360	0.062965	919.20	14598.52	0.052108	919.20	17640.12	-20.84%	0.00%	20.83%	-15.63%
IV	10	360	0.025329	1068.40	42180.98	0.024243	1068.00	44053.45	-4.48%	-0.04%	4.44%	
V	10	360	0.029531	2753.10	93226.96	0.023871	2340.90	98065.11	-23.71%	-14.97%	5.19%	
II	15	360	0.034749	3313.00	95341.54	0.026277	3174.20	120799.65	-32.24%	-4.19%	26.70%	
III	15	360	0.075392	2681.40	35566.15	0.065481	2412.50	36842.90	-15.14%	-10.03%	3.59%	-23.60%
IV	15	360	0.029864	1339.10	44839.75	0.024568	1162.00	47297.60	-21.56%	-13.23%	5.48%	
V	15	360	0.030955	6471.09	209050.20	0.024670	5433.94	220264.97	-25.48%	-16.03%	5.36%	
II	20	360	0.037180	4505.90	121192.29	0.030804	4325.30	140412.16	-20.70%	-4.01%	15.86%	
IV	20	360	0.030914	2161.30	69913.40	0.026797	2024.79	75560.13	-15.36%	-6.32%	8.08%	-20.04%
V	20	360	0.031425	6392.04	203403.62	0.025331	5483.89	216486.17	-24.06%	-14.21%	6.43%	
I	25	360	0.017661	6195.31	350785.86	0.016217	6194.11	381957.98	-8.90%	-0.02%	8.89%	
II	25	360	0.042188	7021.70	166437.69	0.037203	6918.90	185975.36	-13.40%	-1.46%	11.74%	-13.07%
IV	25	360	0.029093	2430.30	83534.26	0.024929	2244.50	90034.05	-16.70%	-7.65%	7.78%	
V	25	360	0.028982	7425.48	256206.88	0.025589	6847.23	267583.16	-13.26%	-7.79%	4.44%	
II	30	360	0.046598	8061.10	172990.69	0.039237	7579.40	193167.48	-18.76%	-5.98%	11.66%	
IV	30	360	0.030096	2711.00	90078.05	0.027500	2711.00	98581.30	-9.44%	0.00%	9.44%	-14.10%
V	30	360	-	-	-	0.026492	7089.94	267621.39	*	*	*	
II	35	360	0.044720	9064.00	202682.79	0.039928	9012.60	225719.39	-12.00%	-0.57%	11.37%	
IV	35	360	0.028667	3529.20	123110.89	0.025625	3318.90	129520.36	-11.87%	-5.96%	5.21%	-15.91%
V	35	360	0.023280	10800.93	463957.66	0.018798	9437.13	502019.17	-23.84%	-12.63%	8.20%	
IV	40	360	0.028206	4339.90	153863.08	0.024177	4107.20	169880.03	-16.66%	-5.36%	10.41%	-16.66%
V	40	360	-	-	-	0.017956	8666.89	482666.21	*	*	*	
IV	45	360	0.026564	4315.80	162469.13	0.025488	4249.00	166704.09	-4.22%	-1.55%	2.61%	-9.55%
V	45	360	0.024165	12181.72	504114.42	0.021034	11194.37	532193.08	-14.89%	-8.11%	5.57%	
IV	50	360	0.025970	4371.40	168327.57	0.024617	4313.00	175203.06	-5.50%	-1.34%	4.08%	-17.86%
V	50	360	0.023685	11729.53	495233.73	0.018189	9405.79	517119.80	-30.22%	-19.81%	4.42%	
								average gap	-15.52%	-5.98%	8.15%	

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 intances of the Challenge

Map	inst	solution by 30 min greedy heuristics			reoptimized solution by FS-MILFP			Gaps				time (s)
		LR_0	TSC_0	TDQ_0	LR_1	TSC_1	TDQ_1	Gap(LR)	Gap(TSC)	Gap(TDQ)	$\overline{\text{Gap}}$(LR)	
I	2.24	0.019026	10041.34	527773.57445264	0.017279	9386.07	543214.28721764	-9.18%	-6.53%	2.93%		9
	2.25	-	-	-	-	-	-	-	-	-	-3,06%	
	2.26	0.054344	62404.48	1148330.77055505	0,054344	62404,48	1148330,77055505	0.0%	0.0%	0.0%		1803
II	2.13	0.055549	10534	189633.71192296	0.053881	10380.4	192655.62192093	-3.00%	-1.46%	1.59%		6
	2.14	0.077024	53361.6	692790.81930113	0.074027	52130.4	704211.81311074	-3.89%	-2.31%	1.65%	-3,12%	117
	2.19	0.082765	54279.4	655821.872372527	0.080723	53611.8	664144.56999617	-2.47%	-1.23%	1.27%		118
III	2.15	0.063036	8521.2	135179.73513544	0.060052	8214.8	136795.33513374	-4.73%	-3.60%	1.20%		1
	2.17	-	-	-	-	-	-	-	-	-	-1,18%	
	2.18	-	-	-	-	-	-	-	-	-		
	X3	-	-	-	-	-	-	-	-	-		
IV	2.16	0.026034	10478.97	402504.05752936	0.024898	10048.17	403573.64368827	-4.36%	-4.11%	0.27%		1
	2.20	0.025649	49589.43	1933391.3971696	0.024292	47400.23	1951253.35263731	-5.29%	-4.41%	0.92%	-4,28%	44
	2.21	0.026353	50449.77	1914386.44922121	0.025536	48998.57	1918805.39998709	-3.10%	-2.88%	0.23%		40
	X2	0.026038	10480.57	402504.05752936	0.024902	10049.77	403573.64368838	-4.36%	-4.11%	0.27%		1
V	2.12	0.02142	37940.99	1771254.51440135	0.017763	31662.29	1782440.91730439	-17.07%	-16.55%	0.63%		74
	2.22	-	-	-	-	-	-	-	-	-	-5,69%	
	2.23	-	-	-	-	-	-	-	-	-		
	X1	0.02142	37940.99	1771254.51440135	0.017763	31662.29	1782440.91730439	-17.07%	-16.55%	0.63%		75
	X4	-	-	-	-	-	-	-	-	-		
	X5	-	-	-	-	-	-	-	-	-		
							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

Map	inst	best known upper bound			reoptimized solution by FS-MILFP			Gaps				time (s)
		LR_0	TSC_0	TDQ_0	LR_1	TSC_1	TDQ_1	Gap(LR)	Gap(TSC)	Gap(TDQ)	$\overline{\text{Gap}}$(LR)	
	2.24	0.013033	8016.33	615067.038590813	0.012523	7716.23	616165.20257248	-3.91%	-3.74%	0.18%		1
I	2.25	0.012411	14652.95	1180611.05703992	0.012137	14383.12	1185081.9329844	-2.21%	-1.84%	0.38%	-2.49%	1800
	2.26	0.012866	15324.29	1191049.15202624	0.012691	15167.15	1195076.95009976	-1.36%	-1.03%	0.34%		1800
	2.13	0.030768	8313.30	270192.576592003	0.029789	8075.20	271076.95696855	-3.18%	-2.86%	0.33%		7
II	2.14	0.037582	28449.60	757005.958032781	0.036996	28181.50	761748.38802664	-1.56%	-0.94%	0.63%	-2.35%	1800
	2.19	0.036018	26467.70	734838.073116171	0.035183	26021.70	739616.77311049	-2.32%	-1.69%	0.65%		1800
	2.15	0.026608	7067.36	265613.54769265	0.025894	6905.06	266668.81134	-2.68%	-2.30%	0.40%		1
III	2.17	0.031538	26934.65	854030.680519699	0.031260	26753.15	855818.6127211	-0.88%	-0.67%	0.21%	-1.38%	31
	2.18	0.033018	27510.50	833208.309782362	0.032700	27311.80	835220.97765779	-0.96%	-0.72%	0.24%		16
	X3	0.031905	27435.56	859927.474862186	0.031584	27253.36	862892.0718163	-1.01%	-0.66%	0.34%		21
	2.16	0.012420	7800.43	628047.868720703	0.012207	7684.93	629548.20316877	-1.71%	-1.48%	0.24%		1
IV	2.20	0.018656	37635.23	2017275.41060711	0.018489	37472.93	2026820.10946348	-0.90%	-0.43%	0.47%	-1.28%	64
	2.21	0.017210	34639.86	2012811.83820039	0.017051	34473.76	2021817.89608588	-0.92%	-0.48%	0.45%		80
	X2	0.01241	7286.53	587140.465024078	0.012214	7199.23	589405.27195929	-1.58%	-1.20%	0.39%		1
	2.12	0.010266	22398.88	2181947.2958854	0.010046	22009.03	2190826.15713757	-2.14%	-1.74%	0.41%		31
	2.22	0.012992	59681.85	4593776.09874866	0.012667	58585.65	4624907.60109552	-2.50%	-1.84%	0.68%		211
V	2.23	0.013311	62221.93	4674428.86039284	0.013003	61133.83	4701647.11295406	-2.31%	-1.75%	0.58%	-2.31%	378
	X1	0.010234	22504.99	2199123.24909978	0.01001	22116.54	2209352.44682696	-2.19%	-1.73%	0.47%		24
	X4	0.013015	61765.63	4745719.20105509	0.01269	60625.04	4777226.26343734	-2.50%	-1.85%	0.66%		207
	X5	0.013994	64727.03	4625451.62636039	0.013681	63669.68	4653731.91491192	-2.24%	-1.63%	0.61%		211
							average gap	-1.95%	-1.53%	0.43%		

Table 6 Improvement of the solutions of the scientific prize winner

Map	inst	initial solution			reoptimized solution by FS-MILFP			Gaps				time (s)
		LR_0	TSC_0	TDQ_0	LR_1	TSC_1	TDQ_1	Gap(LR)	Gap(TSC)	Gap(TDQ)	$\overline{\text{Gap}}$(LR)	
	2.24	0.014175	8143.82	574523.940152157	0.013847	8101.02	585039.64024015	-2,31%	-0,53%	1,83%		5
I	2.25	0.013194	15692.57	1189341.3641055	0.013113	15658.17	1194120.22416276	-0,61%	-0,22%	0,40%	-1.55%	1799
	2.26	0.013580	16854.30	1241118.04644541	0.013347	16629.55	1245979.89916813	-1,72%	-1,33%	0,39%		1801
	2.13	0.034803	8536.40	245277.562111547	0.034538	8475.60	245402.16215033	-0,76%	-0,71%	0,05%		27
II	2.14	0.046737	34286.40	733604.336278942	0.045963	33988.40	739469.71159599	-1,66%	-0,87%	0,80%	-1.21%	1800
	2.19	0.041487	31597.50	761629.455928689	0.041487	31597.50	761629.455928689	0.0%	0.0%	0.0%		1800
	2.15	0.034528	9023.50	261337.526339042	0.034155	8950.70	262060.6640195	-1,08%	-0,81%	0,28%		1
III	2.17	0.049813	40549.83	814049.140787618	0.048749	39813.03	816696.30925331	-2,14%	-1,82%	0,33%	-1.44%	48
	2.18	0.047547	40215.74	845813.993534017	0.047019	39960.11	849878.794481201	-1,11%	-0,64%	0,48%		255
	X3	-	-	-	-	-	-	-	-	-		
	2.16	0.016826	10368.04	616206.44733943	0.016288	10136.84	622349.85815969	-3,20%	-2,23%	1,00%		2
IV	2.20	0.026095	53143.61	2036507.95301131	0.024284	49593.41	2042195.11844444	-6,94%	-6,68%	0,28%	-4.18%	128
	2.21	0.025374	52100.23	2053266.23046444	0.024116	49600.43	2056785.73344054	-4,96%	-4,80%	0,17%		408
	X2	0.016886	10829.14	641323.36793108	0.016613	10696.74	643872.46635999	-1,62%	-1,22%	0,40%		2
	2.12	0.016290	35787.24	2196878.31889949	0.015061	33351.33	2214429.58537573	-7,54%	-6,81%	0,80%		318
	2.22	0.018819	81176.89	4313490.37037945	0.017725	76933.54	4340497.37396281	-5,81%	-5,23%	0,63%		1803
	2.23	-	-	-	-	-	-	-	-	-		
V	X1	0.015974	33695.34	2109356.28342815	0.015017	31949.59	2127535.2895777	-5,99%	-5,18%	0,86%	-6.23%	119
	X4	0.018325	80909.33	4415134.05755929	0.017198	76392.73	4441915.33785093	-6,15%	-5,58%	0,61%		1800
	X5	0.018101	81914.75	4525508.58019449	0.017077	77786.85	4555170.33325599	-5,66%	-5,04%	0,66%		1800
							average gap	-3.49%	-2.92%	0.59%		

improvement brought by RT-MILFP with the FS-MILFP reoptimization can be as high as 62.35%. For instance C on map V with 5 customers and 360 hours' horizon, the improvement is also as high as 67.60%. However, when the instances get larger, the improvement is not so effective.

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-

Table 7 Improvements brought by the RT-MILFP on Instances H

Map	Z	H	greedy heuristics + FS-MILFP			RT-MILFP + FS-MILFP			Gaps			
			LR_0	TSC_0	TDQ_0	LR_1	TSC_1	TDQ_1	Gap(LR)	Gap(TSC)	Gap(TDQ)	$\overline{\text{Gap}}$(LR)
I	32	48	0.015795	1012.02	64072.97	0.015433	988.82	64073.0	-2.29%	-2.29%	0.00%	
II	36	48	0.070220	350.80	4995.71	0.026436	2694.00	101905.8	-62.35%	667.96%	1939.87%	
III	118	48	0.013250	106.00	8000.00	0.013250	106.00	8000.00	0.00%	0.00%	0.00%	-23.25%
IV	179	48	0.026852	417.50	15548.22	0.013204	1886.55	142878.1	-50.83%	351.87%	818.94%	
V	312	48	0.014503	4260.95	293798.39	0.014389	4227.35	293798.4	-0.79%	-0.79%	0.00%	
I	32	72	0.015577	1560.90	100208.11	0.015577	1560.90	100208.1	0.00%	0.00%	0.00%	
II	36	72	0.038247	1033.80	27029.76	0.025423	2906.80	114337.8	-33.53%	181.18%	323.01%	
III	118	72	0.045632	1126.30	24682.02	0.043445	1072.30	24682.0	-4.79%	-4.79%	0.00%	-8.80%
IV	179	72	0.014880	815.90	54830.81	0.014156	2434.00	171944.1	-4.87%	198.32%	213.59%	
V	312	72	0.016827	8225.09	488813.36	0.016693	8156.55	488629.0	-0.80%	-0.83%	-0.04%	
I	32	96	0.015433	2797.77	181286.27	0.015047	2728.97	181369.1	-2.50%	-2.46%	0.05%	
II	36	96	0.030020	1339.10	44606.73	0.025041	2473.90	98795.9	-16.59%	84.74%	121.48%	
III	118	96	0.036421	1449.80	39806.57	0.035105	1397.40	39806.6	-3.61%	-3.61%	0.00%	-4.88%
IV	179	96	0.018955	1477.80	77965.41	0.018955	1477.80	77965.4	0.00%	0.00%	0.00%	
V	312	96	0.017628	10754.47	610081.29	0.017324	10571.97	610250.4	-1.72%	-1.70%	0.03%	
I	32	120	0.015482	3856.08	249062.01	0.015100	3742.14	247827.6	-2.47%	-2.95%	-0.50%	
II	36	120	0.041657	2380.10	57135.46	0.031711	4446.80	140227.4	-23.88%	86.83%	145.43%	
III	118	120	0.034757	1848.60	53186.72	0.033772	1796.20	53186.7	-2.83%	-2.83%	0.00%	-5.89%
IV	179	120	0.019355	2596.50	134148.91	0.019355	2596.50	134148.9	0.00%	0.00%	0.00%	
V	312	120	0.017835	13149.15	737274.97	0.017784	13112.75	737342.2	-0.29%	-0.28%	0.01%	
I	32	144	0.016928	5531.96	326784.77	0.016336	5340.76	326930.6	-3.50%	-3.46%	0.04%	
II	36	144	0.033790	2623.10	77630.48	0.031853	4092.60	128482.9	-5.73%	56.02%	65.51%	
III	118	144	0.038552	3209.70	83255.59	0.038552	3209.70	83255.59	0.00%	0.00%	0.00%	-2.06%
IV	179	144	0.019714	2716.40	137791.71	0.019714	2716.40	137791.71	0.00%	0.00%	0.00%	
V	312	144	0.019355	17467.08	902461.94	0.019152	17301.73	903398.5	-1.05%	-0.95%	0.10%	
I	32	168	0.017842	7280.21	408035.53	0.017842	7280.21	408035.5	0.00%	0.00%	0.00%	
II	36	168	0.037818	3563.00	94214.48	0.037818	3563.00	94214.5	0.00%	0.00%	0.00%	
III	118	168	0.046383	4233.40	91270.89	0.046383	4233.40	91270.89	0.00%	0.00%	0.00%	-0.20%
IV	179	168	0.020346	3623.07	178076.06	0.020346	3623.07	178076.06	0.00%	0.00%	0.00%	
V	312	168	0.019726	23552.32	1194002.72	0.019528	23318.32	1194125.3	-1.00%	-0.99%	0.01%	
I	32	192	0.017262	8646.57	500906.12	0.017109	8570.17	500906.1	-0.89%	-0.88%	0.00%	
II	36	192	0.041032	4385.00	106867.37	0.036483	4942.60	135477.3	-11.09%	12.72%	26.77%	
III	118	192	0.045229	4198.70	92832.33	0.045220	4197.90	92832.3	-0.02%	-0.02%	0.00%	-2.53%
IV	179	192	0.020379	5043.57	247491.69	0.020379	5043.57	247491.69	0.00%	0.00%	0.00%	
V	312	192	0.018794	26470.76	1408467.59	0.018673	26308.36	1408931.3	-0.64%	-0.61%	0.03%	
I	32	216	0.017894	9635.22	538458.58	0.017852	9612.42	538458.6	-0.23%	-0.24%	0.00%	
II	36	216	0.035895	5022.10	139910.67	0.035895	5022.10	139910.67	0.00%	0.00%	0.00%	
III	118	216	0.047664	5563.70	116727.92	0.047664	5563.70	116727.92	0.00%	0.00%	0.00%	-0.17%
IV	179	216	0.020717	6176.97	298165.90	0.020717	6176.97	298165.90	0.00%	0.00%	0.00%	
V	312	216	0.018321	30438.99	1661449.97	0.018208	30252.29	1661515.1	-0.62%	-0.61%	0.00%	
I	32	240	0.018038	10120.46	561066.94	0.017787	9979.66	561066.9	-1.39%	-1.39%	-0.00%	
II	36	240	0.038095	6624.30	173890.71	0.037570	6535.10	173945.5	-1.38%	-1.35%	0.03%	
III	118	240	0.048735	6680.10	137070.26	0.048735	6680.10	137070.26	0.00%	0.00%	0.00%	-0.67%
IV	179	240	0.021776	8340.87	383037.34	0.021776	8340.87	383037.34	0.00%	0.00%	0.00%	
V	312	240	0.019038	34173.88	1795023.41	0.018930	33981.44	1795132.9	-0.57%	-0.56%	0.01%	

sic 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

Table 8 Improvements brought by the RT-MILFP on Instances C with 360 hours horizon

Map	Z	greedy heuristics + FS-MILFP			RT-MILFP + FS-MILFP			Gaps			
		LR_0	TSC_0	TDQ_0	LR_1	TSC_1	TDQ_1	Gap(LR)	Gap(TSC)	Gap(TDQ)	$\overline{\text{Gap}}$(LR)
I	5	0.071656	417.25	5822.98	0.071656	417.25	5822.98	0.00%	0.00%	0.00%	
II	5	0.016458	973.20	59130.98	0.016458	973.20	59130.98	0.00%	0.00%	0.00%	
III	5	0.046143	398.70	8640.58	0.039757	479.90	12070.88	-13.84%	20.37%	39.70%	-23.13%
IV	5	0.098191	549.20	5593.18	0.063720	329.50	5171.08	-35.11%	-40.00%	-7.55%	
V	5	0.036672	442.20	12058.20	0.011880	264.90	22297.70	-67.60%	-40.09%	84.92%	
I	10	0.020793	4009.07	192810.10	0.020793	4009.07	192810.10	0.00%	0.00%	0.00%	
II	10	0.019350	1959.40	101259.80	0.018951	1919.00	101259.80	-2.06%	-2.06%	0.00%	
III	10	0.052108	919.20	17640.12	0.052108	919.20	17640.12	0.00%	0.00%	0.00%	-3.83%
IV	10	0.023874	1070.60	44843.35	0.023874	1070.60	44843.35	0.00%	0.00%	0.00%	
V	10	0.030155	2899.80	96163.57	0.024996	2402.60	96118.07	-17.11%	-17.15%	-0.05%	
I	15	0.020428	5702.37	279138.53	0.020427	5701.97	279138.53	-0.00%	-0.01%	0.00%	
II	15	0.031386	3829.00	121995.92	0.031386	3829.00	121995.92	0.00%	0.00%	0.00%	
III	15	0.061985	2399.80	38715.76	0.061985	2399.80	38715.76	0.00%	0.00%	0.00%	-3.72%
IV	15	0.026928	1360.60	50526.51	0.025725	1299.80	50526.51	-4.47%	-4.47%	0.00%	
V	15	0.030866	6102.99	197727.26	0.026509	5724.58	215945.57	-14.12%	-6.20%	9.21%	
I	20	0.018295	7066.90	386277.84	0.017201	6578.20	382433.54	-5.98%	-6.92%	-1.00%	
II	20	0.034675	5129.70	147937.14	0.033769	4995.70	147938.04	-2.61%	-2.61%	0.00%	
III	20	-	-	-	-	-	-	-	-	-	-4.60%
IV	20	0.026679	2008.99	75301.53	0.026679	2008.99	75301.53	0.00%	0.00%	0.00%	
V	20	0.033434	6989.09	209041.51	0.028618	5887.99	205747.44	-14.40%	-15.75%	-1.58%	
I	25	0.017212	6751.55	392247.86	0.017116	6713.15	392217.66	-0.56%	-0.57%	-0.01%	
II	25	0.039908	7748.40	194154.67	0.039678	7703.60	194154.67	-0.58%	-0.58%	0.00%	
III	25	-	-	-	-	-	-	-	-	-	-3.56%
IV	25	0.027023	2359.20	87303.95	0.025980	2172.80	83633.34	-3.86%	-7.90%	-4.20%	
V	25	0.024505	6428.33	262327.23	0.021370	5560.73	260209.91	-12.79%	-13.50%	-0.81%	
I	30	0.017869	7419.56	415218.14	0.017619	7317.61	415318.54	-1.40%	-1.37%	0.02%	
II	30	0.044091	8715.50	197671.45	0.041911	8292.70	197865.92	-4.94%	-4.85%	0.10%	
III	30	-	-	-	-	-	-	-	-	-	-2.67%
IV	30	0.027312	2896.60	106056.10	0.027312	2896.60	106056.10	0.00%	0.00%	0.00%	
V	30	0.027452	7783.03	283512.80	0.025525	7236.73	283512.80	-7.02%	-7.02%	0.00%	
II	35	0.038618	8520.30	220632.12	0.038455	8486.70	220690.42	-0.42%	-0.39%	0.03%	
III	35	-	-	-	-	-	-	-	-	-	
IV	35	0.026425	2951.50	111691.87	0.025861	2887.90	111669.37	-2.13%	-2.15%	-0.02%	-1.83%
V	35	0.024206	11989.43	495302.28	0.023050	11416.83	495302.28	-4.78%	-4.78%	0.00%	
III	40	-	-	-	-	-	-	-	-	-	
IV	40	0.024447	3901.30	159578.80	0.024447	3901.30	159578.80	0.00%	0.00%	0.00%	
V	40	0.022981	11710.78	509582.77	0.022113	11268.28	509574.47	-3.78%	-3.78%	-0.00%	-1.26%
III	45	-	-	-	-	-	-	-	-	-	
IV	45	0.024601	3921.10	159384.90	0.024601	3921.10	159384.90	0.00%	0.00%	0.00%	
V	45	0.022738	10988.47	483275.05	0.022738	10988.47	483275.05	0.00%	0.00%	0.00%	0.00%
III	50	-	-	-	-	-	-	-	-	-	
IV	50	0.025674	4491.10	174931.19	0.025664	4489.50	174931.19	-0.04%	-0.04%	0.00%	
V	50	0.024910	12540.14	503415.59	0.024573	12370.24	503415.59	-1.35%	-1.35%	0.00%	-0.46%

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 α 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

Andersson, H, M Christiansen, G Desaulniers. 2016. A new decomposition algorithm for a liquefied natural gas inventory routing problem. *International Journal of Production Research* **54**(2) 564–578.

Andersson, Henrik, Arild Hoff, Marielle Christiansen, Geir Hasle, Arne Løkketangen. 2010. Industrial aspects and literature survey: Combined inventory management and routing. *Computers and Operations Research* **37**(9) 1515–1536.

Archetti, Claudia, Luca Bertazzi, Alain Hertz, M. Grazia Speranza. 2012. A Hybrid Heuristic for an Inventory Routing Problem. *INFORMS Journal on Computing* **24**(1) 101–116. doi:10.1287/ijoc.1100.0439.

Archetti, Claudia, Luca Bertazzi, Gilbert Laporte, M. Grazia Speranza. 2007. A Branch-and-Cut Algorithm for a Vendor-Managed Inventory-Routing Problem. *Transportation Science* **41**(3) 382–391.

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.

Archetti, Claudia, Guy Desaulniers, M. Grazia Speranza. 2017. Minimizing the logistic ratio in the inventory routing problem. *EURO Journal on Transportation and Logistics* **6**(4) 289–306.

Bell, Walter J., Louis M. Dalberton, Marshall L. Fisher, Arnold J. Greenfield, R. Jaikumar, Pradeep Kedia, Robert G. Mack, Paul J. Prutzman. 1983. Improving the Distribution of Industrial Gases with and On-Line Computerized Routing and Scheduling Optimizer. *Interfaces* **13**(6) 4–23.

Benoist, Thierry, Frédéric Gardi, Antoine Jeanjean, Bertrand Estellon. 2011. Randomized Local Search for Real-Life Inventory Routing. *Transportation Science* **45**(3) 381–398.

Bertazzi, Luca, M. Grazia Speranza. 2012. Inventory routing problems: an introduction. *EURO Journal on Transportation and Logistics* **1**(4) 307–326. doi:10.1007/s13676-012-0016-7.

Bertazzi, Luca, M. Grazia Speranza. 2013. Inventory routing problems with multiple customers. *EURO Journal on Transportation and Logistics* **2**(3) 255–275. doi:10.1007/s13676-013-0027-z.

Campbell, Ann M., Lloyd W. Clarke, Martin W. P. Savelsbergh. 2001. Inventory Routing in Practice. Paolo Toth, Daniele Vigo, eds., *The Vehicle Routing Problem*, chap. Inventory Routing in Practice. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 309–330.

Campbell, Ann Melissa, Martin W. P. Savelsbergh. 2004. A Decomposition Approach for the Inventory-Routing Problem. *Transportation Science* **38**(4) 488–502.

Coelho, Leandro C., Jean-François Cordeau, Gilbert Laporte. 2012. Consistency in multi-vehicle inventory-routing. *Transportation Research Part C: Emerging Technologies* **24** 270 – 287.

Coelho, Leandro C, Jean-François Cordeau, Gilbert Laporte. 2013. Thirty Years of Inventory Routing. *Transportation Science* **48**(1) 1–19. doi:10.1287/trsc.2013.0472.

Cordeau, Jean François, Demetrio Laganà, Roberto Musmanno, Francesca Vocaturo. 2015. A decomposition-based heuristic for the multiple-product inventory-routing problem. *Computers and Operations Research* **55** 153–166.

Desaulniers, Guy, Jørgen G. Rakke, Leandro C. Coelho. 2016. A Branch-Price-and-Cut Algorithm for the Inventory-Routing Problem. *Transportation Science* **50**(3) 1060–1076.

Dinkelbach, Werner. 1967. On Nonlinear Fractional Programming. *Management Science* **13**(7) 492–498.

Grønhaug, Roar, Marielle Christiansen, Guy Desaulniers, Jacques Desrosiers. 2010. A Branch-and-Price Method for a Liquefied Natural Gas Inventory Routing Problem. *Transportation Science* **44**(3) 400–415.

Hewitt, Mike, George Nemhauser, Martin Savelsbergh, Jin Hwa Song. 2013. A branch-and-price guided search approach to maritime inventory routing. *Computers and Operations Research* **40**(5) 1410–1419.

Irnich, Stefan, Guy Desaulniers. 2005. Shortest Path Problems with Resource Constraints. *Column Generation*. Springer US, Boston, MA, 33–65.

Lagos, Felipe, Natashia Boland, Martin Savelsbergh. 2018. The continuous time inventory routing problem. *Optimization Online* **6439**.

Laporte, Gilbert. 2016. Scheduling issues in vehicle routing. *Annals of Operations Research* **236**(2) 463–474.

You, Fengqi, Pedro M. Castro, Ignacio E. Rossmann. 2009. Dinkelbachs Algorithm as an Efficient Method for Solving a Class of MINLP Models for Large-Scale Cyclic Scheduling Problems. *Computers and Chemical Engineering* **33**(11) 1879–1889.