CIRRELT-2020-12

Mechanisms for Feasibility and Improvement for Inventory-Routing Problems

Thiago A. Guimarães
Cleder M. Schenekemberg
Leandro C. Coelho
Cassius T. Scarpin
José E. Pécora Jr.

April 2020

Document de travail également publié par la Faculté des sciences de l'administration de l'Université Laval, sous le numéro FSA-2020-003.

Mechanisms for Feasibility and Improvement for Inventory-Routing Problems

Thiago A. Guimarães[1,*], Cleder M. Schenekemberg[1,*], Leandro C. Coelho[2,3], Cassius T. Scarpin[1], José E. Pécora Jr.[1,2]

1. Research Group of Technology Applied to Optimization (GTAO), Federal University of Paraná (UFPR), Curitiba, Brazil, Federal Institute of Science and Technology of Paraná (IFPR), Curitiba, Brazil
2. Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)
3. Université Laval, Operations and Decision Systems Department, 2325 de la Terrasse, Québec, Canada G1V 0A6

Abstract. Inventory-routing problems (IRPs) define a class of challenging integrated combinatorial optimization problems, encompassing inventory management and vehicle routing decisions into the same framework. Due to their complexity, exact algorithms can solve only small cases, which is far from practical context which arises with many rich features such as multi-vehicle, multi-depot, and multi-echelon. In this paper, we propose new modular mechanisms that can be embedded into different optimization algorithms, either heuristic or exact ones. We exploit the use of these mechanisms to improve a traditional branch-and-cut scheme. We evaluate our methods by solving three different classes of IRP. In particular, we address the multi-vehicle IRP, the multi-depot IRP, and the multi-depot IRP in a two-echelon supply chain. The results show that our methods are very effective, outperforming other exacts and heuristics approaches from the literature, obtaining 152 new optimal solutions and 353 new best-known solutions on 1712 well-known benchmark instances from the literature.

Keywords. Integer programming; inventory-routing; feasibility; improvement; branch-and-cut.

Acknowledgements. This paper was partly funded by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - grant 1554767, and by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant 2019-00094. Thanks are due to Masoud Chitsaz and Claudia Archetti for making their results available. We also thank the C3SL (Centro de Computação Científica e Software Livre) from Federal University of Paraná for providing parallel computing facilities.

---

* Corresponding authors: thiagoandre@ufpr.br; ledercms@hotmail.com

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2020

1. Introduction

Inventory-routing problems (IRPs) are crucial in logistics operations as they combine into a single framework the joint optimization of inventory management and distribution activities [10]. They have been very well studied by exact approaches such as branch-and-cut (B&C) [1, 4, 6, 7, 11, 12, 18] and branch-and-price-and-cut (B&PC) [14]. A wide range of heuristics and matheuristics have also been proposed [1, 3, 5, 7, 8, 18]. The most challenging set of instances for the single-vehicle IRP was introduced by Archetti et al. [3], where the largest ones have up to 200 customers and a planning horizon with six periods. These instances were adapted to the multi-vehicle IRP (MIRP) by Coelho and Laporte [12], by considering up to 5 vehicles. More recently, Bertazzi et al. [7] and Guimarães et al. [18] have proposed instances for the multi-depot multi-vehicle IRP (MDIRP) and two-echelon MDIRP (2E-MDIRP), with up to 50 customers, six depots, and three vehicles. These problems arise in many practical contexts [2], in which the size of the problem is significantly larger than what current algorithms are capable of solving exactly. State-of-the-art algorithms are capable of proving optimality for instances with up to 50 customers for the single-vehicle IRP and 2E-MDIRP. Regarding the MDIRP, no optimal solution has been proven for instances with more than 15 customers, three periods, and two depots.

Some of these algorithms work by combining different solutions or modifying them to generate new ones. It is common that these new solutions might not be feasible or not yet fully optimized even to a local minimum. On the exact methods, such as B&C and B&PC, infeasible solutions are not explored along the search tree.

As pointed by Talbi [26], feasibility repairing strategies usually operate in a greedy heuristic framework, and are dependent on specific features of the problem at hand. At the same time, the performance of such strategies is closely related to the design and success of these heuristics, especially when the repairing is done by penalizing the objective function [3, 17, 22]. Partition methods, like those based on the fix-and-relax scheme of Escudero and Salmeron [15], are also able to repair infeasibility. However, its constructive nature is not fit to improve a potential solution, requiring a further local search procedure to handle it [25, 27]. On the other hand, enhancing techniques embedded into optimizing algorithms consider only feasible solutions [11, 19].

In this paper, we develop two dedicated strategies for recovering feasibility and improving even partial solutions. The first one, which we call Feasibility and Improvement Procedure (FIP), regains feasibility by allowing the nodes of a route to be reorganized and also improves solutions by optimizing the inventory flow for a given set of delivery routes. Unlike other approaches of this nature, FIP is capable of working on an infeasible solution, making it robust enough for being adapted to varying structural conditions within the scope of the IRP. The second one, called General Improvement Procedure (GIP), is inspired by the local branching technique of Fischetti and Lodi [16], and works by exploring a neighborhood exactly. The way we construct the neighborhood is related to the existing current routes, which is significantly more restricted than the total search space of all possible routes. As a main scientific contribution, both FIP and GIP are flexible enough to be embedded in a wide range of algorithms, being heuristics, matheuristics, or exacts ones, and they can be easily extended to other problems.

The paper is organized as follows. In Section 2, we present a basic formulation of the IRP, while a traditional B&C scheme is provided in Section 3. In Section 4, we describe FIP and GIP and show how to embed them into the B&C. In Section 5, extensive computational experiments on several classes of IRPs from the literature are performed, and the details of the results are discussed. Our conclusions follow in Section 6.

2. A core model for the inventory-routing problem

The most studied variant of the IRP is suitable to represent all needs of our paper. In this sense, we present the classical IRP formulation. This problem is defined over an undirected graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{0, ..., n\}$ is the vertex set, and $\mathcal{E} = \{(i, j) : i, j \in \mathcal{V}, i < j\}$ is the edge set. Vertex 0 represents the plant, and set $\mathcal{C} = \mathcal{V} \backslash \{0\}$ represents customers. A non-negative cost c_{ij} is associated with each edge $(i, j) \in \mathcal{E}$. The planning horizon is defined over a set $\mathcal{T} = \{1, ..., p\}$ of periods. At each period $t \in \mathcal{T}$, a fleet of $|\mathcal{K}|$ homogeneous vehicles of capacity Q and a certain amount r^t of product are made available at plant. Both the plant and customers have a minimum and a maximum inventory level, given by L_i and U_i, respectively, and also incur an inventory holding cost h_i, $i \in \mathcal{V}$, for each unit stocked per period. In $t = 0$, the decision maker knows the initial inventory levels I_i^0, $i \in \mathcal{V}$, and the demands of the customers along the whole planning horizon, given by d_i^t, $i \in \mathcal{C}$.

A vehicle can perform at most one delivery route per period, all routes must start and finish at the plant, and split deliveries are not allowed. The objective of the IRP is to minimize the total inventory and transportation cost, determining when to visit and how much product to deliver for each customer, and how to combine customers deliveries into vehicle routes.

The following variables are used in the mathematical formulation:

- q_i^{kt}: quantity delivered to customer i by vehicle k in period t;
- I_i^t: inventory level at vertex $i \in \mathcal{V}$ at the end of period t;
- $Y_i^{kt} = 1$ if vehicle k visits vertex $i \in \mathcal{V}$ in period t, 0 otherwise;
- $y_{ij}^{kt} = 1$ if vehicle k travels directly between customer i and customer j in period t, 0 otherwise;

- $y_{0i}^{kt} \in \{0,1,2\}$. When $y_{0i}^{kt} = 1$, vehicle k travels from the plant to customer i in period t. If $y_{0i}^{kt} = 2$, a round trip is defined, 0 otherwise;

The IRP can be formulated by (1)−(14):

$$\min \sum_{t\in\mathcal{T}}\sum_{i\in\mathcal{V}} h_i I_i^t + \sum_{t\in\mathcal{T}}\sum_{k\in\mathcal{K}}\sum_{(i,j)\in\mathcal{E}} c_{ij} y_{ij}^{kt} \tag{1}$$

subject to

$$I_0^t = I_0^{t-1} + r^t - \sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{C}} q_i^{kt} \qquad t\in\mathcal{T} \tag{2}$$

$$I_i^t = I_i^{t-1} + \sum_{k\in\mathcal{K}} q_i^{kt} - d_i^t \qquad i\in\mathcal{C}, t\in\mathcal{T} \tag{3}$$

$$\sum_{k\in\mathcal{K}} q_i^{kt} \leq U_i - I_i^{t-1} \qquad i\in\mathcal{C}, t\in\mathcal{T}, \tag{4}$$

$$q_i^{kt} \leq U_i Y_i^{kt} \qquad i\in\mathcal{C}, k\in\mathcal{K}, t\in\mathcal{T} \tag{5}$$

$$\sum_{i\in\mathcal{C}} q_i^{kt} \leq Q Y_0^{kt} \qquad k\in\mathcal{K}, t\in\mathcal{T} \tag{6}$$

$$\sum_{k\in\mathcal{K}} Y_i^{kt} \leq 1 \qquad i\in\mathcal{C}, t\in\mathcal{T} \tag{7}$$

$$\sum_{\substack{j\in\mathcal{V}\\ i<j}} y_{ij}^{kt} + \sum_{\substack{j\in\mathcal{V}\\ j<i}} y_{ji}^{kt} = 2Y_i^{kt} \qquad i\in\mathcal{V}, k\in\mathcal{K}, t\in\mathcal{T} \tag{8}$$

$$\sum_{i\in S}\sum_{\substack{j\in S\\ i<j}} y_{ij}^{kt} \leq \sum_{i\in S} Y_i^{kt} - Y_m^{kt} \qquad S\subseteq\mathcal{C}, |S|\geq 2, m\in S, k\in\mathcal{K}, t\in\mathcal{T} \tag{9}$$

$$q_i^{kt} \geq 0 \qquad i\in\mathcal{C}, k\in\mathcal{K}, t\in\mathcal{T} \tag{10}$$

$$L_i \leq I_i^t \leq U_i \qquad i\in\mathcal{V}, t\in\mathcal{T} \tag{11}$$

$$Y_i^{kt} \in \{0,1\} \qquad i\in\mathcal{V}, k\in\mathcal{K}, t\in\mathcal{T} \tag{12}$$

$$y_{ij}^{kt} \in \{0,1\} \qquad i,j\in\mathcal{C}, i<j, k\in\mathcal{K}, t\in\mathcal{T} \tag{13}$$

$$y_{0i}^{kt} \in \{0,1,2\} \qquad i\in\mathcal{C}, k\in\mathcal{K}, t\in\mathcal{T}. \tag{14}$$

The objective function (1) minimizes the inventory and the transportation costs. Constraints (2) and (3) balance the inventory flow for the plant and customers, while (4) impose inventories policies. Constraints (5) link the quantity delivered with the visit to the customer, while constraints (6) ensure that the total amount delivered does not exceed the vehicle capacity. Split deliveries are avoided by constraints (7). Linking and subtour elimination conditions are imposed by (8) and (9), respectively. The variables domain is defined by constraints (10)−(14).

Several valid inequalities have been proposed to strengthen the formulation, according to the variant of the problem: for the basic IRP and MIRP [4, 13, 14], for the MDIRP [7], and the 2E-MDIRP [18].

3. Branch-and-cut algorithm

The complete enumeration of the subtour elimination constraints (SEC) (9) is impracticable. However, SEC can be dynamically generated along the search process, and added to the problem whenever a subtour is found at the current solution. This strategy can be embedded inside an exact procedure, such as a B&C.

Except for the SEC, at the beginning of the search process, all constraints are generated and added to the root node. Whenever a node of the search tree is solved, we use the CVRPSEP package from Lysgaard et al. [20] to search for subtours. When one is found, its corresponding SEC are added to the search tree. Otherwise, a fractional variable is chosen for branching yielding a new subproblem, and the model is then reoptimized in a new node. The optimization process goes on until a feasible or dominated solution is found, or until there are no more cuts to be added. The pseudocode of our B&C is presented by **Algorithm 1**.

Algorithm 1 Pseudocode of the proposed B&C algorithm

```
At the root node, generate (1)−(14), except (9), and all associated valid inequalities.
while (Processing Time < Time Limit) and (there is node to evaluate) do
    Select one node from the B&C tree.
    Solve the linear programming (LP) relaxation of the node, yielding S_LP.
    while Solution S_LP contains subtours do
        Add violated subtour elimination constraints.
    end while
    if Solution S_LP is integer then
        if f(S_LP) < f(S_BEST) or S_BEST = ∅ then
            S_BEST ← S_LP;
        end if
    else
        Branch on a fractional variable.
    end if
end while
return S_BEST;
```

4. Mechanisms for feasibility and improvement

Although current exact methods are not very efficient in solving large IRP instances, several authors have designed B&C algorithms for this task (see Section 1). In this sense, we now introduce our mechanisms for feasibility and improvement that will be embedded in the B&C algorithm of Section 3.

4.1. Feasibility and improvement procedure

The first stage of FIP operates on the routing structure of any solution of the problem, feasible or not, and works as follows. Whenever a new solution is found, FIP identifies all vertices i visited by vehicle k in period t, in order to generate the set $R_{kt} = \{i \in \mathcal{V} : Y_i^{kt} = 1\}$, $k \in \mathcal{K}$ and $t \in \mathcal{T}$. At this point, one of the following situations can occur:

1. $|R_{kt}| > 0$, $Y_i^{kt} = 1$ with only $i \in \mathcal{C}$. In this case, we generate $R_{kt} \leftarrow R_{kt} \cup \{0\}$ if the problem has only one depot, or $R_{kt} \leftarrow R_{kt} \cup \{u\}$ for the multi-depot case, where u is the depot in which vehicle k is housed.
2. $|R_{kt}| > 1$, $Y_i^{kt} = 1$ with $i \in \mathcal{V}$.
3. $|R_{kt}| = 0$, or $|R_{kt}| = 1$ and the only node is a depot. In this case, the set R_{kt} is disregarded.

For cases 1 and 2, the optimal sequence of visiting the nodes in R_{kt} is determined by applying the B&C procedure of Padberg and Rinaldi [21]. This stage has two essential roles in the search process. First, it recovers feasibility when the solution contains subtours, which necessarily happens in case 1. Second, it minimizes the transportation cost of each vehicle in each period, eliminating all dominated routes from the current solution. Since the set R_{kt} in the case 2 is composed of all nodes visited by vehicle k and its depot, no subtour exist after its optimization, and an infeasible or partial solution becomes feasible.

Figure 1 illustrates the first stage of FIP. In case 1 in a given period t, vehicle $k = 1$ has one subtour, and its associated set $R_{1t} \leftarrow \{1, 2, 3\} \cup \{0\}$ yields the optimal route $\{0 \rightarrow 2 \rightarrow 1 \rightarrow 3 \rightarrow 0\}$. The two subtours of vehicle $k = 2$ generate the set $R_{2t} = \{6, 7, 8, 9, 10\}$. The subtour is eliminated by setting $R_{2t} \leftarrow \{6, 7, 8, 9, 10\} \cup \{0\}$, which produces the optimal route $\{0 \rightarrow 6 \rightarrow 7 \rightarrow 9 \rightarrow 10 \rightarrow 8 \rightarrow 0\}$, and $R_{3t} \leftarrow \{5\} \cup \{0\}$, yielding the roundtrip $\{0 \rightarrow 5 \rightarrow 0\}$. In case 2, the set $R_{1t} = \{0, 2, 3, 4\}$ has a dominating route, which turns into an optimal sequence $\{0 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 0\}$ by applying the B&C of Padberg and Rinaldi [21]. When the set $R_{2t} = \{0, 6, 7, 8, 9, 10\}$ is generated, the subtour of customers $\{7, 9, 10\}$ no longer exists. Thus, the optimal sequence of visits is $\{0 \rightarrow 6 \rightarrow 7 \rightarrow 9 \rightarrow 10 \rightarrow 8 \rightarrow 0\}$. Round trip of set R_{3t} remains.

As the inventory flows remain unchanged, the solution obviously could be suboptimal for the new set of routes generated by the first stage of FIP, even though the routing aspect is optimal. Thus, a second stage

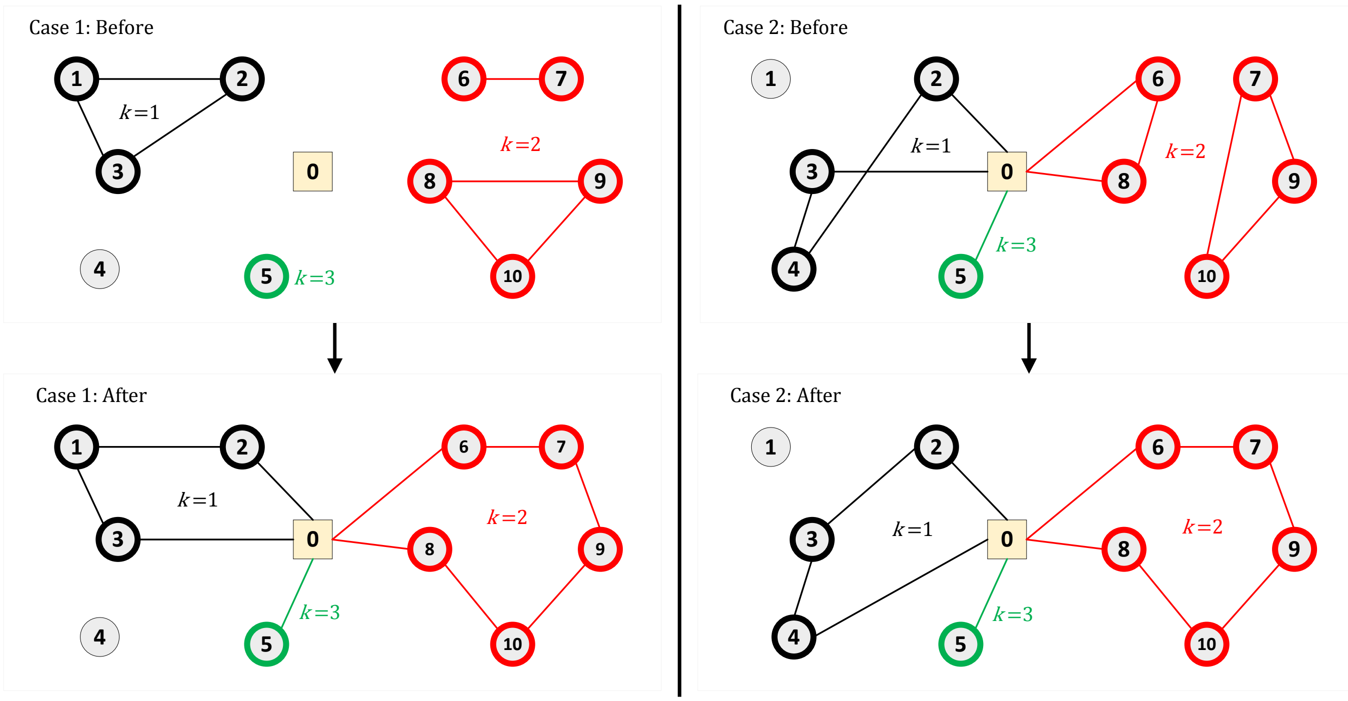


Figure 1: First stage of FIP

is required to determine the delivery quantities which optimizes the inventory cost for the set of routes established in the first stage. We formulate it as a mixed-integer programming (MIP) (see [9]), which uses the binary parameter:

- ψ_i^{kt}: equal to 1 if vertex i is visited in the current route of vehicle k in period t, where $Y_i^{kt} = 1$, 0 otherwise.

The model is formulated as:

$$\min \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{V}} h_i I_i^t \tag{15}$$

subject to (2)–(4), (10), (11), and to:

$$q_i^{kt} \leq U_i \psi_i^{kt} \qquad i \in \mathcal{C}, k \in \mathcal{K}, t \in \mathcal{T} \tag{16}$$

$$\sum_{i \in \mathcal{C}} q_i^{kt} \leq Q \psi_0^{kt} \qquad k \in \mathcal{K}, t \in \mathcal{T}. \tag{17}$$

The objective function (15) minimizes the inventory costs. Constraints (16) link the quantity delivered with the visit to the customer, while constraints (17) ensure that the total amount delivered does not exceed the vehicle capacity.

After solving the second stage of FIP, we identify all customers i visited by vehicle k in period t and define the following set $B_{kt} = \{i \in \mathcal{C} : q_i^{kt} > 0\}$, for all $k \in \mathcal{K}$ and for all $t \in \mathcal{T}$. After this, the optimal sequence of visiting between depot and each $|B_{kt}| > 0$ is given by the B&C of Padberg and Rinaldi [21], where $k \in \mathcal{K}$ and $t \in \mathcal{T}$. This removes unserved customers, minimizing the transportation cost.

4.2. General improvement procedure

The General Improvement Procedure (GIP) was first introduced by Schenekemberg et al. [24] for a specific case of IRP with fleet management, and has also been adapted to the two-echelon production-routing problem (2E-PRP) by Schenekemberg et al. [23]. In this paper, we generalize GIP to handle several variants of IRPs. It improves a given solution $\bar{s}$ by performing removals, insertions, and customers swaps only on established routes. By preventing customers from being served by new routes, GIP explores a much smaller search space than the original model.

In this sense, let $\mathcal{A}_{kt}$ be the set of customers $i \in C$ served by vehicle k in period t. A subset of edges adjacent to the depot is also introduced, given by $\mathcal{E}_{kt} \subseteq \mathcal{E}$, with $\mathcal{E}_{kt} = \{(i,j) : i,j \in \{0\} \cup \mathcal{A}_{kt} \cup \mathcal{A}_{kt}^c\}$, and $\mathcal{A}_{kt}^c = \mathcal{C} \backslash \mathcal{A}_{kt}$. Formally, GIP sets free q_i^{kt}, Y_i^{kt} and y_{ij}^{kt} to be optimized if and only if their associated routes exist, where $(i,j) \in \mathcal{E}_{kt}$ and $|\mathcal{A}_{kt}| > 0$. Otherwise, when $\mathcal{A}_{kt} = \emptyset$, the associated variables q_i^{kt}, Y_i^{kt} and y_{ij}^{kt} are set to zero, and all other decision variables are also free to be optimized. Due to this large search space, it can still be difficult to improve solutions in an acceptable time. To overcome this limitation, GIP limits the number of changes to each route, by adding the following constraints inspired by the local branching scheme [16].

$$\sum_{i \in \mathcal{A}_{kt}} \left(1 - Y_i^{kt}\right) + \sum_{i \in \mathcal{A}_{kt}^c} Y_i^{kt} \leq \mathcal{B} \qquad k \in \mathcal{K}, t \in \mathcal{T}, |\mathcal{A}_{kt}| > 0. \tag{18}$$

The positive integer parameter $\mathcal{B}$ limits the number of binary variables switching their value with respect to each existing route from a solution $\bar{s}$, either from 1 to 0 or from 0 to 1. The set of solutions satisfying constraints (18) define the $\mathcal{B}$-OPT neighborhood $\mathcal{N}(\bar{s}, \mathcal{B})$ of $\bar{s}$. The parameter $\mathcal{B}$ must be chosen appropriately since when $\mathcal{B}$ is too small, the probability of finding solutions better than $\bar{s}$ is low. Otherwise, big values for $\mathcal{B}$ do not allow a thorough exploration of the neighborhood in a suitable time.

Figure 2 illustrates a situation with 14 customers, vehicle capacity $Q = 400$, before (a) and after (b) applying GIP for two consecutive periods, $[t, t+1]$. The transportation cost is calculated as the sum of distances on each route, represented by D, while $\sum_{t' \in [t,t+1]} \sum_{i \in \mathcal{V}} h_i I_i^{t'}$ gives the inventory cost. The total amount delivered by a route is L. The routes of vehicle $k = 1$ in t and $t+1$ before GIP define $\mathcal{A}_{1,t} = \{7, 8, 9, 10, 11\}$ and $\mathcal{A}_{1,t+1} = \{7, 13\}$, while for vehicle $k = 2$ they are $\mathcal{A}_{2,t} = \{1, 2, 3, 4, 5, 6\}$ and $\mathcal{A}_{2,t+1} = \{2, 12, 14\}$, respectively. Therefore, their associated complementary sets are given by $\mathcal{A}^c_{1,t} = \{1, 2, 3, 4, 5, 6, 12, 13, 14\}$, $\mathcal{A}^c_{1,t+1} = \{1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 14\}$, $\mathcal{A}^c_{2,t} = \{7, 8, 9, 10, 11, 12, 13, 14\}$, and $\mathcal{A}^c_{2,t+1} = \{1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13\}$, respectively. After applying GIP by considering $\mathcal{B} = 4$ as maximum number of movements, we obtain one removal (customer 8) and two insertions (customers 3 and 13) for $k = 1$, while for $k = 2$ two removals (customers 2 and 3) and two insertions (customers 12 and 14) are performed in t. In period $t + 1$, one removal (customer 13) and three insertions (customers 2, 8 and 12) occur for $k = 1$. These changes imply the removals of all customers served by $k = 2$, eliminating the route of that vehicle. At the same time, by optimizing q_i^{kt}, a new inventory flow is obtained. The transportation cost decreases from 104.0 to 92.0, while the inventory cost changes from 69.1 to 63.5. Thus, the total cost after applying GIP is 155.5 against 173.1 before.

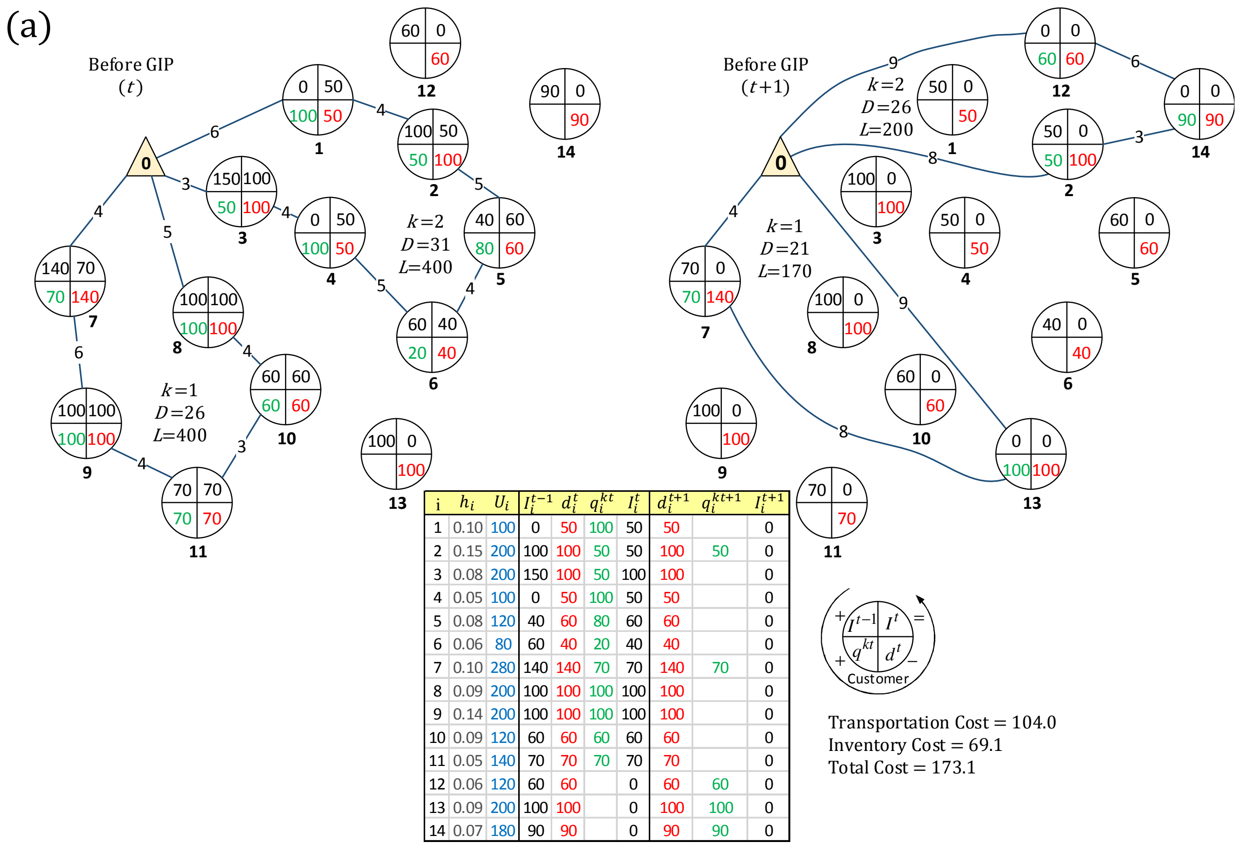
4.3. General optimization framework: improved B&C algorithm

In this section, we describe how to embed the FIP and GIP into the B&C algorithm presented in Section 3. Due to its limitation to handle large instances in the IRP scope, our purpose is to help B&C not only to find better solutions; in many situations, B&C is not even capable of finding a first feasible solution. Whenever a new best solution is found, either by FIP or by the B&C, GIP is applied in order to improve it by performing insertions, removals, and swaps of customers on existing routes. After preliminary experiments, we apply GIP only once a new best solution found, since the computational time required for GIP can significantly impact the global processing time. We provide a pseudocode for the improved B&C in **Algorithm 2**.

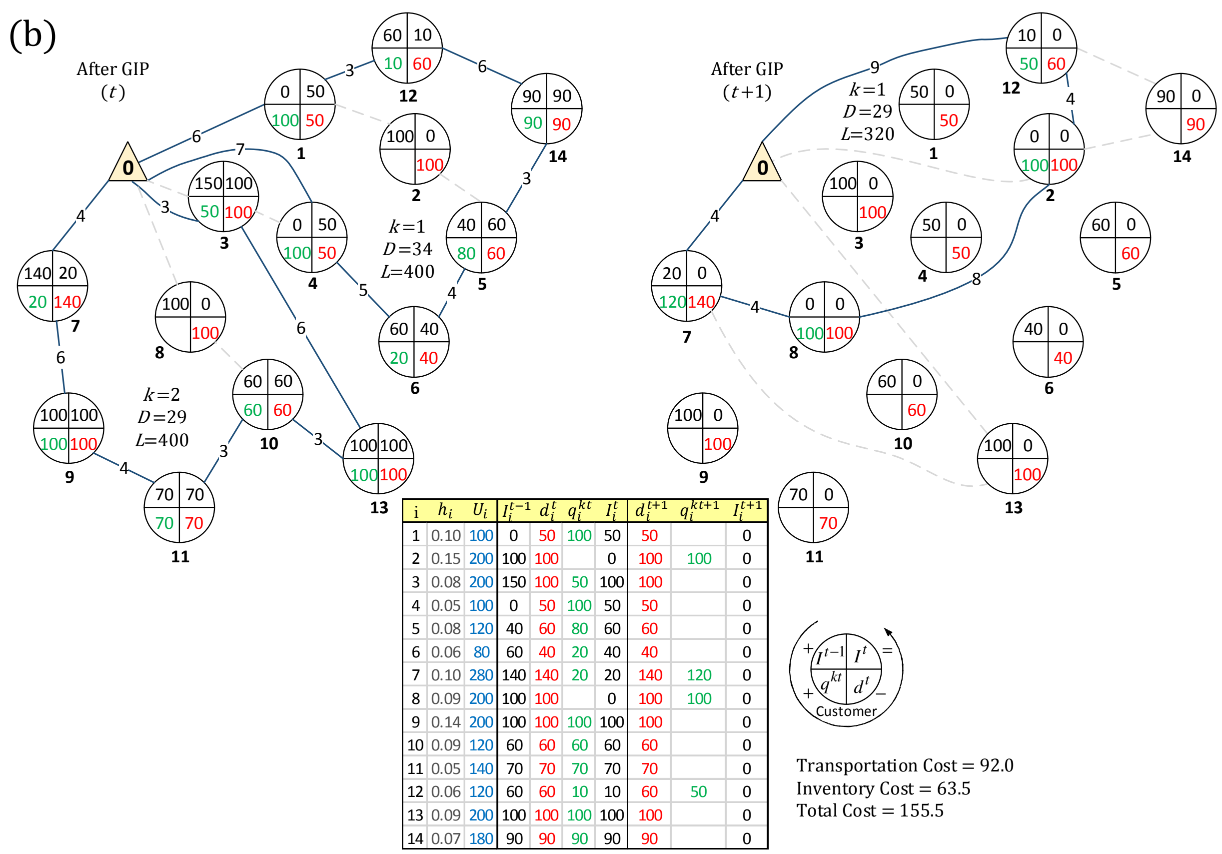
5. Computational experiments

In this section, we provide an extensive analysis of our algorithms by solving three different IRPs from the literature, which still have many open instances. First, we solve the classical IRP and its multi-vehicle (MIRP) case. Due to the flexibility of our method, we also evaluate its performance over the MDIRP, introduced by Bertazzi et al. [7], and the 2E-MDIRP, proposed by Guimarães et al. [18]. Our algorithms were coded in C++ using Gurobi 8.1.0, running in CentOS Linux operating system. After a tuning phase, we set the MIP model for inventory flow on the second stage of FIP to be solved up to optimality, while GIP is executed according to the number of customers as follows:

- IRP and MIRP: 360 s if $|\mathcal{C}| \leq 50$, 600 s otherwise.



i	h_i	U_i	I_i^{t-1}	d_i^t	q_i^{kt}	I_i^t	d_i^{t+1}	q_i^{kt+1}	I_i^{t+1}
1	0.10	100	0	50	100	50	50		0
2	0.15	200	100	100	50	50	100	50	0
3	0.08	200	150	100	50	100	100		0
4	0.05	100	0	50	100	50	50		0
5	0.08	120	40	60	80	60	60		0
6	0.06	80	60	40	20	40	40		0
7	0.10	280	140	140	70	70	140	70	0
8	0.09	200	100	100	100	100	100		0
9	0.14	200	100	100	100	100	100		0
10	0.09	120	60	60	60	60	60		0
11	0.05	140	70	70	70	70	70		0
12	0.06	120	60	60		0	60	60	0
13	0.09	200	100	100		0	100	100	0
14	0.07	180	90	90		0	90	90	0



i	h_i	U_i	I_i^{t-1}	d_i^t	q_i^{kt}	I_i^t	d_i^{t+1}	q_i^{kt+1}	I_i^{t+1}
1	0.10	100	0	50	100	50	50		0
2	0.15	200	100	100		0	100	100	0
3	0.08	200	150	100	50	100	100		0
4	0.05	100	0	50	100	50	50		0
5	0.08	120	40	60	80	60	60		0
6	0.06	80	60	40	20	40	40		0
7	0.10	280	140	140	20	20	140	120	0
8	0.09	200	100	100		0	100	100	0
9	0.14	200	100	100	100	100	100		0
10	0.09	120	60	60	60	60	60		0
11	0.05	140	70	70	70	70	70		0
12	0.06	120	60	60	10	10	60	50	0
13	0.09	200	100	100	100	100	100		0
14	0.07	180	90	90	90	90	90		0

Figure 2: A numerical example for GIP

Algorithm 2 Pseudocode of the improved B&C algorithm

At the root node, generate (1)−(14), except (9), and all associated valid inequalities.
while (Processing Time < Time Limit) and (there is node to evaluate) **do**
    Select one node from the B&C tree.
    Solve the LP relaxation of the node, yielding S_{LP}.
    while Solution S_{LP} contains subtours **do**
        Add violated subtour elimination constraints.
    end while
    if Solution S_{LP} is integer **then**
        Apply FIP to S_{LP}, yielding S_{FIP};
        if $f(S_{FIP}) < f(S_{BEST})$ or $S_{BEST} = \emptyset$ **then**
            $S_{BEST} \leftarrow S_{FIP}$;
            Apply GIP to S_{FIP}, yielding S_{GIP}
            if $f(S_{GIP}) < f(S_{BEST})$ **then**
                $S_{BEST} \leftarrow S_{GIP}$;
            end if
        end if
    else
        Branch on a fractional variable.
    end if
end while
return S_{BEST};

- MDIRP: 360 s if $|\mathcal{C}| \leq 30$, 480 s otherwise.
- 2E-MDIRP: 360 s if $|\mathcal{C}| \leq 50$, 480 s otherwise.

5.1. Instance sets

We have considered a small and a large set of instances from the IRP for the single-vehicle case, introduced by Archetti et al. [4] and Archetti et al. [3], respectively, and adapted to the MIRP by Coelho and Laporte [11]. The first set considers a total of 160 small size instances, in which the number of customers ranges from $\mathcal{C} = \{5, 10, \ldots, 50\}$ when $|\mathcal{T}| = 3$, and $\mathcal{C} = \{5, 10, ..., 30\}$ when $|\mathcal{T}| = 6$. The second one is more challenging having a total of 60 large size instances with $\mathcal{C} = \{50, 100, 200\}$ and $|\mathcal{T}| = 6$. Both sets consider two classes of inventory costs, low and high. The fleet of homogeneous vehicles $\mathcal{K} = \{1, 2, ..., 5\}$, totaling $(160 + 60) \times 5 = 1100$ instances.

Regarding to the MDIRP, we have evaluated the set introduced by Bertazzi et al. [7], consisting of $\mathcal{C} = \{5, 10, \ldots, 50\}$, a fleet of $|\mathcal{K}| = 3$, with $|\mathcal{T}| = \{3, 6\}$. The number of depots ranges from 2 to 6, according to the number of customers, for a total of 100 instances.

Finally, we also have performed an analysis of our methods over the instances of a new variant of IRP in a more complex structure. The 2E-MDIRP [18] is defined over a two-echelon supply chain, formed by input suppliers and plants on the first echelon, and by plants and customers on the second one. The problem takes into account input pickups, delivery decisions, and inventory management over a planning horizon. The set of instances for this problem consists of four combinations on the first echelon, being one supplier-one plant, two suppliers-two plants, three supplier-two plants, and two suppliers-three plants. The set of customers is $\mathcal{C} = \{5, 10, 25, 50\}$, with two combinations of fleet, $|\mathcal{K}| = 1$ and $|\mathcal{K}| = 3$, two classes of inventory costs, and two inventory policies, totaling 512 instances.

All instances and detailed results are available from `https://www.leandro-coelho.com/local-search-for-feasibility-and-improvement-for-irps/`.

5.2. Benchmark algorithms

We have compared our improved B&C with state-of-the-art algorithms, both exact and heuristics, that have also been evaluated on the same datasets. The first exact algorithm for the IRP was a B&C introduced by Archetti et al. [4]. Later, Coelho and Laporte [12] present a B&C for the MIRP, which incorporates a solution improvement mechanism. In Adulyasak et al. [1], another B&C and new valid inequalities are presented, addressed to the multi-vehicle production-routing problem, where the MIRP is a particular case. In the same work, the authors also developed a matheuristic algorithm based on the adaptive large neighborhood search

(ALNS) mechanism. Desaulniers et al. [14] proposed a B&PC with a new set of cutting planes, and evaluated the algorithm over the set of small instances. More recently, Avella et al. [6] presented an IRP reformulation derived from a single-period substructure. The authors defined a generic family of valid inequalities and performed computational experiments only for 50 customers with three periods and 30 customers with six periods from the small set. Regarding heuristic and matheuristic procedures, Archetti et al. [3] designed a hybrid algorithm for the single-vehicle IRP, that combines a tabu search (TS) scheme with MIP models. Later, Archetti et al. [5] extended the matheuristic to the MIRP case. The most recent heuristic for the IRP was proposed by Chitsaz et al. [8], under a unified formulation for the assembly routing problem, where the MIRP is a related case. The authors proposed a three-phase decomposition matheuristic. Experiments on large-scale multi-vehicle instances outperform state-of-the-art heuristics for the MIRP.

Regarding the MDIRP, Bertazzi et al. [7] proposed a B&C algorithm and a three-phase matheuristic based on clustering, routing, and optimization. For the 2E-MDIRP, Guimarães et al. [18] compare a B&C against a matheuristic where ALNS handles the delivery routes, while delivery and pickup quantities, and improvements are solved exactly via MIP subproblems. In both problems, the matheuristics outperformed the B&C algorithms.

Table 1 provides the benchmark algorithms with their respective hardware specifications, solver versions, problems solved, and running times.

Table 1: Benchmark algorithms

Reference	Problem	Method	Algorithm	CPU	Threads	T(s)	Solver
Archetti et al. [4]	IRP	Exact	B&C	Pentium IV 2.8 GHz	Default	7200	CPLEX 9.0
Archetti et al. [3]	IRP	Heuristic	TS + MIP	Intel Dual Core 1.86 GHz	Default	3600	CPLEX 10.1
Archetti et al. [5]	IRP	Matheuristic	Multi-phase MIP + TS	Xeon W3680 3.33 GHz	8	3000	CPLEX 12.5
Chitsaz et al. [8]	IRP	Matheuristic	3-phase decomp.	Xeon X5650 2.67 GHz	1	43200	CPLEX 12.6
Coelho and Laporte [12]	IRP	Exact	B&C	Xeon 2.66 GHz	6	43200* 86400**	CPLEX 12.3
Desaulniers et al. [14]	IRP	Exact	B&PC	Intel Core i7-2600 3.4 GHz	1	7200	CPLEX 12.2
Avella et al. [6]	IRP	Exact	B&C	Intel Core i7-2620 2.7 GHz	1	3600	Xpress 7.6
Adulyasak et al. [1]	IRP	Exact Heuristic	B&C ALNS	Duo CPU PC 2.10 GHz	Default	43200	CPLEX 12.3
Bertazzi et al. [7]	MDIRP	Exact Matheuristic	B&C 3-phase matheur.	Intel Core i7-6500U 2.50 GHz	Default	21700	CPLEX 12.6.1
Guimarães et al. [18]	2E-MDIRP	Exact Matheuristic	B&C ALNS + MIP	Intel Core i5-6200U 2.40 GHz	Default	7200	Gurobi 7.0.2
This paper	IRP MDIRP 2E-MDIRP	Exact	Improved B&C	Xeon CPU E5-2630 v2 2.60 GHz	6	7200	Gurobi 8.1.0

Notes: * for small instances of Archetti et al. [4], ** for large instances of Archetti et al. [3].

5.3. Preliminary results

We start our analysis by assessing the impact of the FIP and GIP on the B&C. To this end, we consider a subset of 15 instances from each problem, being five small, five medium and five large ones, totaling 45 instances. Table 2 presents the number of solutions found by the algorithms (SF), optimals proven (OPT), and the average results for: upper bounds (UB), lower bounds (LB), optimality gap, and processing time. The impact of the FIP and GIP jointly embedded into the B&C is highlighted by SF, OPT, the quality of the UBs, and average gaps, without compromising the processing time.

Table 2: General comparison of methods for the MIRP with $|\mathcal{K}| = 5$, MDIRP and 2E-MDIRP

Method	SF	OPT	$\overline{UB}$	$\overline{UB}^*$	$\overline{LB}$	$\overline{GAP}(\%)$	$\overline{T(s)}$
B&C	31	19	-	7212.3	**10714.5**	-	4630.1
B&C + FIP	**45**	19	12901.4	7114.5	10697.7	10.2	4638.7
B&C + GIP	**45**	20	12146.3	7117.0	10689.9	7.1	4591.9
B&C + FIP + GIP	**45**	**22**	**12052.3**	**7066.6**	10668.6	**6.4**	**4551.1**

* Where B&C found a solution.

5.4. Detailed results for the IRP and the MIRP

We first present in Table 3 the list of papers, instances, and configurations used so far in the literature. Then, in Table 4 we present the results for the small instances of Archetti et al. [4]. We provide the number of instances (#), the number of times that the corresponding algorithm found the best-known solution (BKS), the number of times that the BKS was exclusively found by the algorithm (EBKS) (among all exact algorithms), the number of optimal solutions found (OPT), and the average upper bound $\overline{UB}$. We highlight that our algorithm dominates the B&C and the B&PC up to $|\mathcal{K}| = 4$. Although the B&PC was more effective in proving optimality when $|\mathcal{K}| = 5$, our algorithm obtained more EBKS, and decreased the $\overline{UB}$ by almost 10% compared to the B&PC. In general, our improved B&C has found 129 new BKSs for the whole set of 800 instances.

Since the B&C algorithms of Adulyasak et al. [1] and Avella et al. [6] have performed experiments on only a subset small instances of Archetti et al. [4] (see Table 3), we present the comparison on Tables 5 and 6. Among these subsets, 80 instances have been solved by Avella et al. [6] and Adulyasak et al. [1]. Besides that, the exact algorithms of Coelho and Laporte [12] and Desaulniers et al. [14] also have solved these subsets. All of these results have been considered in the BKS and EBKS computation. Specifically, Table 5 shows the comparison of results against Adulyasak et al. [1], in which our improved B&C dominates the results in terms of $\overline{UB}$, obtains 31 exclusive BKS against only 7, and provides two new optimal solutions.

Table 3: Instance set for the IRP and the MIRP

Reference	**Type**	$\|\mathcal{C}\|$	$\|\mathcal{K}\|$	$\|\mathcal{T}\|$
Archetti et al. [4]	Small	5 to 50	1	3, 6
Archetti et al. [3]	Small	5 to 50	1	3, 6
	Large	50, 100, 200	1	6
Archetti et al. [5]	Small	5 to 50	2 to 5	3, 6
	Large	50, 100, 200	2 to 5	6
Coelho and Laporte [12]	Small	5 to 50	1 to 5	3, 6
	Large	50, 100	1, 2, 3	6
	Large	200	1	6
Desaulniers et al. [14]	Small	5 to 50	2 to 5	3, 6
Adulyasak et al. [1]	Small	5 to 25	2, 3	3, 6
		30 to 50	3, 4	3, 6
Avella et al. [6]	Small	50	2 to 5	3
		15 to 30	2 to 5	6
Chitsaz et al. [8]	Small	5 to 50	1 to 5	3, 6
	Large	50, 100, 200	1 to 5	6
This paper	Small	5 to 50	1 to 5	3, 6
	Large	50, 100, 200	1 to 5	6

Table 4: Comparison with exact algorithms over small instances Archetti et al. [4]

Reference	**Statistics**	$\|\mathcal{K}\| = 1$	$\|\mathcal{K}\| = 2$	$\|\mathcal{K}\| = 3$	$\|\mathcal{K}\| = 4$	$\|\mathcal{K}\| = 5$	**Total**
Archetti et al. [4] - B&C	#	160	-	-	-	-	160
	BKS*	160	-	-	-	-	160
	EBKS*	0	-	-	-	-	0
	OPT	160	-	-	-	-	160
	$\overline{UB}$	7319.0	-	-	-	-	-
Coelho and Laporte [12] - B&C	#	160	160	160	160	160	800
	BKS*	160	156	122	88	58	584
	EBKS*	0	0	4	9	1	14
	OPT	160	152	112	76	52	552
	$\overline{UB}$	7319.0	7875.9	8641.5	10148.0	11478.5	-
Desaulniers et al. [14] - B&PC	#	-	160	160	160	160	640
	BKS*	-	140	108	93	98	439
	EBKS*	-	0	1	9	27	37
	OPT	-	75	77	84	90	326
	$\overline{UB}$	-	7969.5	8705.8	9793.4	10972.9	-
This paper	#	160	160	160	160	160	**800**
	BKS	160	160	147	132	127	**726**
	EBKS	0	3	27	43	56	**129**
	OPT	160	155	119	82	61	**577**
	$\overline{UB}$	7319.0	7875.1	8601.9	9382.0	10092.2	-

* Among Adulyasak et al. [1], Archetti et al. [4], Avella et al. [6], Coelho and Laporte [12], Desaulniers et al. [14].

It is important to mention that Adulyasak et al. [1] have run their B&C for 43200s, six times more than us. Table 6 presents the comparison against Avella et al. [6], in which our algorithm obtains 112 exclusive BKS and proved 60 optimal solutions, against only 3 and 14, respectively.

Table 5: Comparison with Adulyasak et al. [1] over a subset of small instances Archetti et al. [4]

Reference	Statistics	$\lvert\mathcal{K}\rvert = 2$	$\lvert\mathcal{K}\rvert = 3$	$\lvert\mathcal{K}\rvert = 4$	Total
Adulyasak et al. [1] - B&C	#	100	150	50	300
	BKS*	100	124	22	244
	EBKS*	0	4	3	7
	OPT	100	117	20	237
	$\overline{\boldsymbol{UB}}$	6833.6	8125.3	9604.5	-
This paper	#	100	150	50	300
	BKS*	100	138	35	273
	EBKS*	0	18	13	31
	OPT	100	119	18	237
	$\overline{\boldsymbol{UB}}$	6833.6	8112.7	9525.4	-

* Among Avella et al. [6], Coelho and Laporte [12], Desaulniers et al. [14].

Table 6: Comparison with Avella et al. [6] over a subset of small instances Archetti et al. [4]

Reference	Statistics	$\lvert\mathcal{K}\rvert = 2$	$\lvert\mathcal{K}\rvert = 3$	$\lvert\mathcal{K}\rvert = 4$	$\lvert\mathcal{K}\rvert = 5$	Total
Avella et al. [6] - B&C	#	50	50	50	50	200
	BKS*	22	2	2	1	27
	EBKS*	0	0	2	1	3
	OPT	14	0	0	0	14
	$\overline{\boldsymbol{UB}}$	11927.6	13249.3	14675.1	16151.4	-
This paper	#	50	50	50	50	200
	BKS*	50	41	40	46	177
	EBKS*	3	26	38	45	112
	OPT	45	15	0	0	60
	$\overline{\boldsymbol{UB}}$	11801.2	12901.0	14126.4	15411.6	-

* Among Adulyasak et al. [1], Coelho and Laporte [12], Desaulniers et al. [14].

To the best of our knowledge, only Coelho and Laporte [12] have solved the large-scale instances of Archetti et al. [3] exactly, by running their B&C for up to 86400s. In this sense, Table 7 presents the results, in which we report the same instances evaluated by the authors (40 instances when $\lvert\mathcal{K}\rvert = 2$ and $\lvert\mathcal{K}\rvert = 3$). It is possible to observe that our algorithm outperformed the competition in all statistics, even though the running time of our method was only 7200s, or 8.3% of theirs. Our improved B&C has substantially improved the $\overline{UB}$, especially for $\lvert\mathcal{K}\rvert = 2$ and $\lvert\mathcal{K}\rvert = 3$. We also highlight that our method has proven optimality for instance absH6high5n100. It is the first time that an instance with 100 customers and six periods has a proven optimal solution.

Furthermore, our algorithm was the first method to solve instances for the classes with $|\mathcal{K}| = 4$ and $|\mathcal{K}| = 5$ exactly, having provided LB for all of them, even though the results have not been compared due to the absence of any competing method. These results appear online.

Table 7: Comparison with exact algorithms over large instances Archetti et al. [3]

Reference	Statistics	$\lvert\mathcal{K}\rvert = 1$	$\lvert\mathcal{K}\rvert = 2$	$\lvert\mathcal{K}\rvert = 3$	Total
Coelho and Laporte [12] - B&C	#	60	40	40	140
	BKS	20	1	0	21
	EBKS	1	1	0	2
	OPT	17	0	0	17
	$\overline{UB}$	45994.7	45774.0	63651.3	-
This paper	#	60	40	40	140
	BKS	59	39	40	138
	EBKS	40	39	40	119
	OPT	21	0	0	21
	$\overline{UB}$	40482.6	29186.5	32100.0	-

Overall, we have proven nine and four new optimal solutions among 181 and 283 open small and large instances, respectively. Table 8 details these results, and also highlights the limitation of the exact methods in proving optimal solutions for the IRP.

Table 8: Total of new optimal solutions found by our algorithm for instances of the IRP

$\lvert\mathcal{K}\rvert$	**Small set** [4]		**Large set** [3]	
	Previously Open	**New Optimal**	**Previously Open**	**New Optimal**
1	0	0	43	4
2	7	3	60	0
3	41	4	60	0
4	64	1	60	0
5	69	1	60	0
Total	**181**	**9**	**283**	**4**

Since FIP and GIP operate fundamentally at the UB level, we compare our improved B&C with the best heuristic and matheuristic algorithms available in the literature [1, 3, 5, 8]. Tables 9 and 10 detail the results for small and large size instances, respectively. We highlight that whenever one of the exact algorithms [1, 4, 6, 12, 14] has solved an instance, its result is considered in the BKS, UB, and EBKS computation. Regarding approximate methods, the BKS row shows the number of times that at least one of them was able to achieve the best-known solution, while exclusive BKS highlights when BKS was provided by one of the approximated algorithms and not reached by our improved B&C. The results on Table 9 show that our algorithm is very competitive for the set of small size instances, equivalent or better in 704 out of 800 instances, and has provided 108 new BKS. Regarding large size set, the results on Table 10 show that our method was very competitive up to $|\mathcal{K}| = 2$. Even in more complex cases, with $|\mathcal{K}| = 5$, our method was able to promote improvements, finding one new BKS. In general, our algorithm has improved the solutions for 70 instances of a set of 300, i.e., 23.3 % of the large size cases.

Table 9: Comparison with exact, heuristic, and matheuristic algorithms over small instances Archetti et al. [4]

Reference	Statistics	$\|\mathcal{K}\| = 1$	$\|\mathcal{K}\| = 2$	$\|\mathcal{K}\| = 3$	$\|\mathcal{K}\| = 4$	$\|\mathcal{K}\| = 5$	Total
All methods	#	160	160	160	160	160	800
	BKS	160	157	136	126	113	692
	EBKS	0	0	16	38	40	94
	Best $\overline{UB}$	7319.0	7875.6	8613.4	9389.0	10105.5	-
This paper	#	160	160	160	160	160	**800**
	BKS	160	160	144	122	118	**704**
	EBKS	0	3	24	34	47	**108**
	$\overline{UB}$	7319.0	**7875.1**	**8601.9**	**9382.0**	**10092.2**	-

Table 10: Comparison with exact, heuristic, and matheuristic algorithms over large instances of Archetti et al. [3]

Reference	Statistics	$\|\mathcal{K}\| = 1$	$\|\mathcal{K}\| = 2$	$\|\mathcal{K}\| = 3$	$\|\mathcal{K}\| = 4$	$\|\mathcal{K}\| = 5$	Total
All methods	#	60	60	60	60	60	300
	BKS	35	28	51	57	59	230
	EBKS	16	28	51	57	59	211
	Best $\overline{UB}$	40448.1	42007.1	43207.4	44698.5	46312.0	-
This paper	#	60	60	60	60	60	**300**
	BKS	44	32	9	3	1	89
	EBKS	25	32	9	3	1	**70**
	$\overline{UB}$	40482.6	42845.3	46370.0	51157.1	55392.7	-

5.5. Results for the MDIRP

In the second part of the experiments, we provide new exact solutions for the MDIRP. Since the problem was recently introduced by Bertazzi et al. [7], there is only one B&C and one matheuristic available in the literature. In Table 11, we present the average results, grouped according to the number of customers, totaling 10 instances per row. For all of groups, our method obtains significant improvements, having proven 38 optimal against 11 from the literature in much shorter computational time. Our method was able to improve the solution values for 73 instances, outperforming by 7.18% the best UB obtained between the B&C and matheuristic available for this problem.

Table 11: Results for the MDIRP, $|\mathcal{T}| = 3$ and $|\mathcal{T}| = 6$

$\|\mathcal{C}\|$	B&C [7]				Matheuristic [7]			Best	This Paper						Δ% Best
	BKS	OPT	$\overline{UB}$	$\overline{T(s)}$	BKS	$\overline{UB}$	$\overline{T(s)}$	$\overline{UB}$ [7]	BKS	New BKS	OPT	$\overline{UB}$	$\overline{GAP}(\%)$	$\overline{T(s)}$	$\overline{UB}$ [7]
5	8	6	2426.1	9796.8	9	2425.7	9.8	2425.7	10	1	10	2422.0	0.0	917.8	**-0.15**
10	3	3	3783.7	15648.3	4	3625.0	168.3	3625.0	9	6	5	3468.3	1.6	3637.8	**-4.32**
15	2	2	6275.0	20115.8	3	5853.1	1316.3	5841.7	8	6	5	5521.6	1.6	4288.5	**-5.48**
20	0	0	7774.4	21700.0	2	6599.0	1134.3	6590.2	8	8	5	6110.2	5.4	4293.3	**-7.28**
25	0	0	7768.4	21700.0	2	6839.0	1406.4	6839.0	8	8	6	6153.6	3.0	5419.8	**-10.02**
30	0	0	8375.7	21700.0	0	7455.9	1851.7	7455.9	10	10	3	6659.4	5.8	6140.2	**-10.68**
35	0	0	9442.9	21700.0	3	8344.3	294.3	8344.3	7	7	2	7697.6	6.4	6208.2	**-7.75**
40	0	0	9450.7	21700.0	1	8853.5	1358.6	8853.5	9	9	0	8005.2	10.1	7200.0	**-9.58**
45	0	0	9223.2	21700.0	1	8805.1	709.1	8786.8	9	9	2	8004.9	11.0	6929.8	**-8.90**
50	0	0	10644.2	21700.0	1	9868.7	2017.2	9868.7	9	9	0	9112.5	18.4	7200.0	**-7.66**
Avg			**7516.4**	**19746.1**		**6866.9**	**1026.6**	**6863.1**				**6315.5**	**6.3**	**5223.5**	**-7.18**
Total	**13**	**11**			**26**				**87**	**73**	**38**				

5.6. Results for the 2E-MDIRP

Finally, we extend the experiments to one of the newest problems on the IRP context. Table 12 summarizes the results for the 2E-MDIRP with the instances grouped according to the inventory policy and the number of customers, totaling 32 instances per row. Considering only the instances where the B&C [18] found a solution, our method was substantially better, reducing the average UB from 4964.5 to 4780.1, or 3.73%. It is also interesting to note that our improved B&C is very robust when the deliveries are more strict, as in the OU policy. Especially for these cases when OU is imposed on deliveries (ML-OU and OU-OU), our algorithm was able to prove optimal solutions for instances with up to 50 customers, outperforming the ALNS matheuristic proposed by Guimarães et al. [18] in several cases. In general, our exact method provides 102 new BKS among 512 instances, i.e., for almost 20% of the instances.

5.7. Contribution of FIP and GIP to the quality of solutions

Here we assess the contribution of FIP and GIP concerning the quality of the solutions obtained by our algorithm. Since our improved B&C has a mechanism that provides an initial solution, we measure, for each instance tested, the improvement generated by each procedure. We also measure the computational time taken by the FIP and GIP routines. Table 13 shows these results. Of all the improvement obtained by our improved B&C with respect to its initial solution, FIP accounts for most of the improvement (between 37% and 59%), while the processing time required corresponds to an average of 8% of the total, corroborating the initial idea that infeasible solutions are potentially good and can streamline the search strategy. As GIP depends on existing routes, its contribution to improvement ends up being more significant in problems with more vehicles. Finally, we highlight that the role of B&C in the overall optimization accounts for 25% of

Table 12: Results for the 2E-MDIRP

Policy	$\vert\mathcal{C}\vert$	B&C [18]					Matheuristic [18]			Best	This paper								Δ% Best
		BKS	**OPT**	$\overline{UB}^*$	$\overline{GAP}^*$	$\overline{T(s)}$	**BKS**	$\overline{UB}$	$\overline{T(s)}$	$\overline{UB}$ [18]	**BKS**	**New BKS**	**OPT**	$\overline{UB}^*$	$\overline{GAP}^*$ (%)	$\overline{UB}$	$\overline{GAP}$	$\overline{T(s)}$	$\overline{UB}$ [18]
ML-ML	5	32	32	3475.2	0.0	57.5	32	3475.2	142.6	3475.2	32	0	32	3475.2	0.0	3475.2	0.0	3.8	0.00
	10	26	26	4704.5	4.0	1430.7	28	4659.4	444.8	4658.9	31	2	30	4655.8	0.4	4655.8	0.4	657.9	**-0.07**
	25	17	14	4743.8	3.4	4502.2	23	5434.7	481.1	5433.5	31	7	26	4687.8	0.0	5401.5	2.2	1602.5	**-0.59**
	50	8	8	6098.9	6.4	5519.8	18	7853.3	766.0	7853.3	28	14	18	5732.9	0.0	7817.8	4.8	4224.6	**-0.45**
ML-OU	5	32	32	3648.2	0.0	228.3	32	3648.2	158.2	3648.2	32	0	32	3648.2	0.0	3648.2	0.0	3.5	0.00
	10	26	26	5179.3	6.1	1472.4	31	4986.7	676.8	4986.4	30	0	30	4990.1	0.6	4990.1	0.6	781.4	0.07
	25	12	11	5262.6	11.2	4854.7	17	5968.4	587.1	5968.1	29	14	26	4934.6	0.0	5954.5	3.1	2102.2	**-0.23**
	50	0	0	6392.9	11.9	7200.0	21	8786.3	1382.7	8786.3	19	11	12	5861.3	0.0	8885.3	8.8	5164.5	1.13
OU-ML	5	32	32	3534.8	0.0	53.3	31	3535.0	51.0	3534.8	32	0	32	3534.8	0.0	3534.8	0.0	4.0	0.00
	10	27	26	4812.8	3.9	1417.3	27	4775.0	334.8	4774.7	32	3	30	4769.0	0.3	4769.0	0.3	642.8	**-0.12**
	25	13	14	4957.1	4.3	4550.7	23	5641.3	522.6	5641.0	30	8	26	4884.1	0.0	5592.6	2.0	1594.3	**-0.86**
	50	9	8	6034.0	3.1	5497.3	18	8358.6	798.8	8358.6	27	14	17	5930.7	0.0	8344.8	6.1	4303.3	**-0.17**
OU-OU	5	32	32	3708.0	0.0	174.7	32	3708.0	44.3	3708.0	32	0	32	3708.0	0.0	3708.0	0.0	3.1	0.00
	10	26	26	5311.7	6.1	1500.7	29	5109.6	461.4	5108.6	32	1	30	5107.2	1.1	5107.2	1.1	774.2	**-0.03**
	25	11	11	5538.1	12.1	4877.5	19	6159.8	769.6	6159.1	30	11	26	5097.7	0.0	6132.2	3.1	2060.7	**-0.44**
	50	0	0	6029.8	14.9	7200.0	15	9278.0	1584.8	9278.0	23	17	11	5464.4	0.0	9355.7	9.3	5266.1	0.84
Avg				**4964.5**	**5.5**	**3158.5**		**5711.1**	**575.4**	**5710.8**				**4780.1**	**0.1**	**5710.8**	**2.6**	**1824.3**	**-0.06**
Total		**303**	**298**				**396**				**470**	**102**	**410**						

* Where B&C found a solution.

the average improvement, while the processing time required approximately 75% of the entire computational experiment.

Table 13: Impact % of GIP, FIP, and B&C in improving the initial solution

Instances	$\%UB_{FIP}$	$\%UB_{GIP}$	$\%UB_{B\&C}$	$\%T_{FIP}(s)$	$\%T_{GIP}(s)$	$\%T_{B\&C}(s)$
IRP, $\vert\mathcal{K}\vert = 1$	59.52	12.72	27.76	24.17	17.37	58.46
IRP, $\vert\mathcal{K}\vert = 2$	48.85	23.47	27.68	8.2	22.13	69.67
IRP, $\vert\mathcal{K}\vert = 3$	38.36	36.78	24.86	3.14	17.07	79.79
IRP, $\vert\mathcal{K}\vert = 4$	41.39	39.62	18.99	2.53	17.33	80.14
IRP, $\vert\mathcal{K}\vert = 5$	39.25	41.95	18.80	2.11	14.45	83.44
MDIRP	37.34	23.20	39.46	1.88	6.42	91.70
2E-MDIRP	58.19	18.19	23.62	13.85	19.26	66.89
Avg.	**46.13**	**27.99**	**25.88**	**7.98**	**16.29**	**75.73**

6. Conclusions

In this paper, we have presented two mechanisms to promote feasibility and improvement for the IRP and its richer variants. The first of these mechanisms, which we call FIP, allows to recover or complete a partial solution, turning it into a feasible one. Moreover, it improves a solution by disregarding dominated routes while it optimizes the inventory flow for the associated set of new routes. The second mechanism, called GIP, aims to explore a given solution, building a neighborhood that depends on the existing routes, generating a search space much smaller than of all possible routes, and fully exploring it exactly. These mechanisms are flexible enough to be adapted to other optimization methods, such as heuristics and matheuristics, and were evaluated within the framework of a traditional B&C, which we call improved B&C.

Our algorithm was tested on the classic instances of the single- and multi-vehicle IRP and two richer variants, the MDIRP and the 2E-MDIRP. On the small benchmark set of Archetti et al. [4], consisting of 800 instances, our improved B&C obtained 726 BKS and 129 exclusive ones (not found by any other exact algorithm from the literature), in addition to proving 577 optimal solutions, the largest number among all the exact methods compared. Similarly, when comparing our algorithm on the 300 large size instances of Archetti et al. [3], the results of our method greatly outperform the competition in only a fraction of the run time. Regarding heuristic and matheuristic algorithms, our improved B&C was able to reach 704 BKS and 108 exclusive ones on the small instances of Archetti et al. [4], in addition to obtaining 70 new BKS on the large instances of Archetti et al. [3].

Our improved B&C was also compared on the 100 instances of MDIRP proposed by Bertazzi et al. [7] against a B&C and a three-phase matheuristic, obtaining about 73% new BKS, and a 7.18 % reduction in the value of the objective function. We also have solved the 512 instances of 2E-MDIRP introduced by Guimarães et al. [18], obtaining 470 BKS, 102 new BKS and 410 optimal solutions, significantly outperforming the existing B&C and the matheuristics.

Our methods proved to be competitive and flexible enough to address the IRP and some of its variants, highlighting the remarkable contribution of the two mechanisms proposed in this paper. As future research, one can evaluate the performance of these mechanisms within other methods, such as heuristics and matheuristics. In addition, one can develop strategies for making fractional solutions feasible within the traditional B&C framework, in order to explore and exploit each node solved within the search tree.

References

[1] Adulyasak, Y., Cordeau, J.F., Jans, R., 2014. Formulations and branch-and-cut algorithms for multi-vehicle production and inventory routing problems. INFORMS Journal on Computing 26, 103–120.

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

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

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

[5] Archetti, C., Boland, N., Speranza, M.G., 2017. A matheuristic for the multivehicle inventory routing problem. INFORMS Journal on Computing 29, 377–387.

[6] Avella, P., Boccia, M., Wolsey, L.A., 2018. Single-period cutting planes for inventory routing problems. Transportation Science 52, 497–508.

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

[8] Chitsaz, M., Cordeau, J.F., Jans, R., 2019. A unified decomposition matheuristic for assembly, production, and inventory routing. INFORMS Journal on Computing 31, 134–152.

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

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

[11] Coelho, L.C., Laporte, G., 2013a. A branch-and-cut algorithm for the multi-product multi-vehicle inventory-routing problem. International Journal of Production Research 51, 7156–7169.

[12] Coelho, L.C., Laporte, G., 2013b. The exact solution of several classes of inventory-routing problems. Computers & Operations Research 40, 558–565.

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

[14] Desaulniers, G., Rakke, J.G., Coelho, L.C., 2015. A branch-price-and-cut algorithm for the inventory-routing problem. Transportation Science 1655, 1–17.

[15] Escudero, L.F., Salmeron, J., 2005. On a fix-and-relax framework for a class of project scheduling problems. Annals of Operations Research 21, 140–163.

[16] Fischetti, M., Lodi, A., 2003. Local branching. Mathematical Programming 98, 23–47.

[17] Gendreau, M., Iori, M., Laporte, G., Martello, S., 2006. A tabu search algorithm for a routing and container loading problem. Transportation Science 40, 342–350.

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

[19] Larrain, H., Coelho, L.C., Cataldo, C., 2017. A variable MIP neighborhood descent algorithm for managing inventory and distribution of cash in automated teller machines. Computers & Operations Research 85, 22 – 31.

[20] Lysgaard, J., Letchford, A.N., Eglese, R.W., 2004. A new branch-and-cut algorithm for the capacitated vehicle routing problem. Mathematical Programming 100, 423–445.

[21] Padberg, M., Rinaldi, G., 1991. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. SIAM Review 33, 60–100.

[22] Qiu, M., Fu, Z., Eglese, R., Tang, Q., 2018. A tabu search algorithm for the vehicle routing problem with discrete split deliveries and pickups. Computers & Operations Research 100, 102 – 116.

[23] Schenekemberg, C.M., Scarpin, C.T., Pecora Jr., J.E., Guimarães, T.A., Coelho, L.C., 2019. The two-echelon production-routing problem. Technical Report CIRRELT-2019-56. Montréal.

[24] Schenekemberg, C.M., Scarpin, C.T., Pecora Jr., J.E., Guimarães, T.A., Coelho, L.C., 2020. The two-echelon inventory-routing problem with fleet management. Computers & Operations Research forthcoming.

[25] Schimidt, T.M.P., Scarpin, C.T., Loch, G.V., Schenekemberg, C.M., 2019. Heuristic approaches to solve a two-stage lot sizing and scheduling problem. IEEE Latin America Transactions 17, 434–443.

[26] Talbi, E.G., 2009. Metaheuristics: From Design to Implementation. Wiley Publishing, New Jersey.

[27] Toledo, C., Arantes, M., Hossomi, M., França, P., Akartunalı, K., 2015. A relax-and-fix with fix-and-optimize heuristic applied to multi-level lot-sizing problems. Journal of Heuristics 21, 1–31.