A Branch-and-Cut Algorithm for the Multi-Product Multi-Vehicle Inventory-Routing Problem

Leandro C. Coelho
Gilbert Laporte

September 2012

CIRRELT-2012-53
A Branch-and-Cut Algorithm for the Multi-Product Multi-Vehicle Inventory-Routing Problem

Leandro C. Coelho¹,²,*  Gilbert Laporte¹,³

¹ Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)
² Department of Logistics and Operations Management, HEC Montréal, 3000 Côte-Sainte-Catherine, Montréal, Canada H3T 2A7
³ Department of Management Sciences, HEC Montréal, 3000 Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

Abstract. The combined operation of distribution and inventory control achieved through a vendor-managed inventory strategy creates a synergetic interaction which benefits supplier and customers. Inventory-Routing Problems (IRPs) arise when inventory and routing decisions must be taken simultaneously, which yields a difficult combinatorial optimization problem. While most IRP research deals with a single product, there are often several products involved in distribution activities. In this paper, we propose a branch-and-cut algorithm for the solution of IRPs with multiple products and multiple vehicles. We formally define and model the problem, and we solve it exactly. We also consider the inclusion of consistency features which are meaningful in a multi-product environment and help improve the quality of the service offered.

Keywords. Inventory-routing, multi-product, multi-vehicle, exact, branch-and-cut, consistency.

Acknowledgements. This work was partly supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant 39682-10. This support is gratefully acknowledged. We also thank the Réseau québécois de calcul de haute performance (RQCHP) for providing computing facilities.

Results and views expressed in this publication are the sole responsibility of the authors and do not necessarily reflect those of CIRRELT.

Les résultats et opinions contenus dans cette publication ne reflètent pas nécessairement la position du CIRRELT et n'engagent pas sa responsabilité.

* Corresponding author: Leandro.Coeelho@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2012

© Copyright Coelho, Laporte and CIRRELT, 2012
1 Introduction

Inventory-Routing Problems (IRPs) have received increased attention in the last years. Several heuristics [5, 10, 15] and exact algorithms [4, 34] are available for its single vehicle and single product version. Recently the multi-vehicle case (MIRP) was also solved heuristically [16] and exactly [1, 14]. In all these papers, there is only one product, whereas many vendor-managed inventory (VMI) applications are concerned with the distribution of several products. In this paper we model and optimally solve the multi-product multi-vehicle IRP (MMIRP).

Several applications of the MMIRP are well documented. Most of these arise in maritime logistics, namely in the distribution of several types of fuel and gases by compartmentalized ships [6, 8, 27, 8, 23, 29, 31, 37, 38]. Non-maritime applications include the distribution of perishable products [18, 19], the transportation of gases by tanker trucks [7], the automobile components industry [3], and fuel delivery [28].

Like the applications, the assumptions and solution procedures are also diverse. A Lagrangian-based heuristic based on a VRP algorithm was proposed by Chien et al. [12]. A special case of the problem with a single customer and predetermined frequency deliveries was studied by Speranza and Ukovich [35, 36]. Bertazzi et al. [9] later expanded these studies to handle multiple customers, in which each customer is treated individually, and those with the same optimal frequency are aggregated for the computation of routes. Carter et al. [11] have proposed a two-phase heuristic which first solves an allocation problem to determine when and how much to deliver to customers, and then constructs delivery routes. Sindhuchoa et al. [32] assumes that each vehicle always carries the same set of items and then formulates the IRP as a set partitioning problem. An MMIRP with demand uncertainty was studied by Huang and Lin [24] who solved it by means of an ant colony optimization algorithm. A variation of the multi-product version which also considers multiple suppliers but only one customer (many-to-one structure) was analyzed by Moin et al. [25]. The authors derive lower and upper bounds after solving a linear mathematical formulation with a commercial solver and then compute better upper bounds by means of a genetic algorithm. Building upon the previous structure, Ramkumar et al. [30] proposed a MILP for a many-to-many multi-item multi-depot IRP. However, computational results showed the limitations of the method since small instances with only two vehicles, two products, two suppliers, three customers and three periods could not be solved to optimality within eight hours of computing time. Despite the relatively large number of papers in this field, no comparison between algorithms is possible due to the different assumptions made in each paper and to the lack of a common test bed.

We consider a version of the MMIRP in which a supplier is responsible for the distribution of several products to a set of geographically dispersed customers using a fleet of vehicles. Customers and vehicles have a maximum inventory capacity which is shared by all products. This problem arises, for example, in the grocery and beverage industries. We deal with a deterministic version of the problem in which the supplier has full knowledge of future demands, such that no stockout at the customers is allowed to occur. We propose an exact algorithm for this problem. We increase the scope of an exact approach based on a branch-and-cut algorithm put forward for the single-vehicle case by Archetti et al. [4] and extended to the MIRP by Coelho and Laporte [14] and by Adulyasak et al. [1]. Our implementation is shown to obtain optimal or high quality solutions for medium size instances of the problem. Using our model and algorithm we are able to evaluate the impact of each parameter of the problem, namely the number of customers, products, vehicles and periods, in terms of computational difficulty and solution cost. We also evaluate the impact of ensuring stable operations over time through the inclusion of consistency features [16]. We measure the effect of these features in a multi-product context.

The objective of this paper is threefold. First, we formally describe, model and solve the MMIRP exactly by branch-and-cut. We consider a large number of customers with several products, vehicles and planning horizons sizes so that the impact of each parameter can be properly measured. A secondary contribution is the introduction of consistency features in a multi-product environment, which is also assessed in our experiments. Third, we propose a large set of benchmark instances in order to evaluate our algorithm.
are made publicly available to the research community to allow the assessment of future algorithms. The proposed test bed is designed to cover a large set of combinations regarding the number of customers, the number of products, the number of vehicles and the length of the planning horizon, ranging from relatively easy instances to very challenging ones.

The remainder of the paper is organized as follows. In Section 2 we formally describe the MMIRP and the consistency features considered in this paper. In Section 3 we propose mixed-integer linear programming formulations for these problems. In Section 4 we present our branch-and-cut algorithm. We then provide the results of extensive computational experiments in Section 5. Conclusions follow in Section 6.

2 The Multi-Product Multi-Vehicle Inventory-Routing Problem

We now formally introduce the MMIRP and the MMIRP with consistency features. These problems are defined on a graph \( G = (V, A) \) where \( V = \{0, \ldots, n\} \) is the vertex set and \( A \) is the arc set. Vertex 0 represents the supplier and the vertices of \( V' = V \setminus \{0\} \) represent customers. The supplier is responsible for distributing a set \( \mathcal{M} = \{1, \ldots, M\} \) of products to the customers. Both the supplier and customers incur a unit inventory holding cost \( h_{mi} \) per period \( (i \in V, m \in \mathcal{M}) \). The length of the planning horizon is \( p \) and, at each time period \( t \in T = \{1, \ldots, p\} \), the quantity of product \( m \) made available at the supplier is \( r_{mt} \). We assume the supplier has enough inventory to meet all the demand during the planning horizon and that inventories are not allowed to be negative. The variables \( I_{m0}^t \) and \( I_{mi}^t \) are defined as the inventory levels of product \( m \) at the end of period \( t \), respectively at the supplier and at customer \( i \). At the beginning of the planning horizon the decision maker knows the current inventory level of the supplier and of all customers \( (I_{m0}^t \text{ for } i \in V, m \in \mathcal{M}) \), and has full knowledge of the demand \( d_{mt} \) of product \( m \) of each customer \( i \) for each time period \( t \). There is a set \( K = \{1, \ldots, K\} \) of vehicles available. Each vehicle is able to perform one route per time period to deliver products from the supplier to a subset of customers. A routing cost \( c_{ij} \) is associated with arc \( (i, j) \in A \).

The objective of the MMIRP is to minimize the total inventory-distribution cost while meeting the demand for each customer. The replenishment plan is subject to the following constraints:

- the inventory level at each customer can never exceed its maximum capacity;
- inventory levels are not allowed to be negative;
- each of the supplier’s vehicles can perform at most one route per time period; each route starts and ends at the supplier;
- vehicle capacities cannot be exceeded.

The solution to the problem should determine which customers to serve in each time period using which of the supplier’s vehicles, how much to deliver of each product to each visited customer as well as which routes to use.

In order to increase quality of service, additional features may be added to reflect a number of concerns, e.g., workforce management [21, 33, 39] and regularity of service [16]. To this end, Coelho et al. [16] have incorporated several consistency features in order to increase the quality of service while remaining cost effective. Some of them are particularly meaningful when distributing several products. We incorporate the following two consistency features within the MMIRP:

- Driver partial consistency. Extending the work of Groër et al. [22] to the MMIRP, the standard driver consistency feature requires that each customer be assigned to only one driver. We implement a relaxation of this rule to allow some of the deliveries not to be subject to it, that is, most deliveries should be performed by the same driver to the same customer.
- Visit spacing. This feature imposes a temporal space between consecutive visits to the same customer.
### 3 Mathematical Programming Formulations

In this section we propose mixed-integer linear programming formulations for the MMIRP and for its counterpart with consistency features.

Assuming that the transportation cost matrix is symmetric, we work with an undirected formulation in order to reduce the number of variables. Thus, the model uses variables $x_{ij}^{kt}$ equal to the number of times edge $(i, j)$ with $i < j$ is used on the route of vehicle $k$ in period $t$. We also introduce binary variables $y_{i}^{kt}$ equal to one if and only if node $i$ (the supplier or a customer) is visited by vehicle $k$ in period $t$. Let $I_{i}^{mt}$ denote the inventory level of product $m$ at vertex $i$ at the end of period $t$ and $I_{j}^{mt}$ denote the inventory level of product $m$ at vertex $j$. Let $I_{m}^{mt}$ denote the inventory level of product $m$ at vertex $i$ at the end of period $t$. Each customer $i$ in $V'$ has a common inventory holding capacity $C_{i}$ which is shared for all products $m \in M$. The capacity of each vehicle is $Q_{k}$, which is also shared for all products being carried. We denote by $q_{i}^{mkt}$ the quantity of product $m$ delivered from the supplier using vehicle $k$ to customer $i$ in time period $t$. Note that with respect to the single product problem [14], this model requires exactly the same number of binary variables. The only difference lies in the number of continuous variables $I_{i}^{mt}$ and $q_{i}^{mkt}$. The problem can then be formulated as follows:

\[
\begin{align*}
\text{minimize} & \quad \sum_{i \in V} \sum_{m \in M} \sum_{t \in T} h_{i}^{m} I_{i}^{mt} + \sum_{i \in V} \sum_{j \in V, i < j} \sum_{k \in K} \sum_{t \in T} c_{ij} x_{ij}^{kt}, \\
\text{subject to} & \quad I_{i}^{mt} = I_{0}^{i, t-1} + I_{i}^{mt} - \sum_{k \in K} I_{i}^{mt} - \sum_{m \in M} I_{m}^{mt} - \sum_{m \in M} q_{i}^{mt} + \sum_{m \in M} r_{i}^{m, t-1}, \quad i \in V, \quad m \in M, \quad t \in T \\
& \quad I_{m}^{mt} \geq 0, \quad i \in V, \quad m \in M, \quad t \in T \\
& \quad \sum_{m \in M} I_{i}^{mt} \leq C_{i}, \quad i \in V', \quad t \in T \\
& \quad \sum_{m \in M} \sum_{k \in K} q_{i}^{mkt} \leq C_{i} - \sum_{m \in M} I_{m}^{mt, t-1}, \quad i \in V', \quad t \in T \\
& \quad q_{i}^{mkt} \leq C_{i} y_{i}^{kt}, \quad i \in V', \quad m \in M, \quad k \in K, \quad t \in T \\
& \quad \sum_{i \in V'} \sum_{m \in M} q_{i}^{mkt} \leq Q_{k}, \quad k \in K, \quad t \in T \\
& \quad \sum_{j \in V, i < j} x_{ij}^{kt} + \sum_{j \in V, j < i} x_{ji}^{kt} = 2 y_{i}^{kt}, \quad i \in V, \quad k \in K, \quad t \in T \\
& \quad \sum_{i \in S} \sum_{j \in S, i < j} x_{ij}^{kt} \leq \sum_{i \in S} y_{i}^{kt} - \sum_{i \in S} y_{g}^{kt}, \quad S \subseteq V', \quad k \in K, \quad t \in T
\end{align*}
\]

for some $g \in S$

\[
\begin{align*}
q_{i}^{mkt} & \geq 0, \quad i \in V', \quad k \in K, \quad t \in T \\
x_{i0}^{kt} & \in \{0, 1, 2\}, \quad i \in V', \quad k \in K, \quad t \in T \\
x_{ij}^{kt} & \in \{0, 1\}, \quad i, j \in V', \quad k \in K, \quad t \in T \\
y_{i}^{kt} & \in \{0, 1\}, \quad i \in V, \quad k \in K, \quad t \in T.
\end{align*}
\]

Constraints (2) and (3) define the inventories at the supplier and at the customers, while constraints (4) prevent stockouts at the supplier. Constraints (5) impose maximal inventory level at the customer.
Constraints (6) and (7) link the quantities delivered to the routing variables. In particular, they only allow a vehicle to deliver any products to a customer if the customer is visited by this vehicle. Constraints (8) ensure the vehicle capacities are respected while constraints (9) and (10) are degree constraints and subtour elimination constraints, respectively. Constraints (11)–(14) enforce integrality and non-negativity conditions on the variables.

### 3.1 Valid Inequalities

The formulation just presented can be further strengthened by adding the following valid inequalities:

\[
\begin{align*}
x_{i0}^{kt} &\leq 2y_i^{kt} & i \in \mathcal{V}, & k \in \mathcal{K}, & t \in \mathcal{T} \quad (15) \\
x_{ij}^{kt} &\leq y_i^{kt} & i, j \in \mathcal{V}, & k \in \mathcal{K}, & t \in \mathcal{T} \quad (16) \\y_i^{kt} &\leq y_0^{kt} & i \in \mathcal{V}', & k \in \mathcal{K}, & t \in \mathcal{T} \quad (17)
\end{align*}
\]

\[
\sum_{k \in \mathcal{K}} \sum_{l=1}^{t} y_i^{kt} \geq \left\lceil \left( \sum_{k \in \mathcal{K}} \sum_{l=1}^{t-1} d_i^{mkt} - I_i^{m0} \right) / C_i \right\rceil & i \in \mathcal{V}, & m \in \mathcal{M}, & t \in \mathcal{T}. \quad (18)
\]

Constraints (15) and (16) are referred to as logical inequalities. They enforce the condition that if the supplier is the successor of a customer in the route of vehicle \( k \) in period \( t \), i.e., \( x_{i0}^{kt} = 1 \) or 2, then \( i \) must be visited by the same vehicle, i.e., \( y_i^{kt} = 1 \). A similar reasoning is applied to customer \( j \) in inequalities (16). Constraints (17) include the supplier in the route of vehicle \( k \) if any customer is visited by that vehicle in that period. Constraints (18) ensure that customer \( i \) is visited at least the number of times corresponding to the right-hand side of the inequality. This inequality is only valid if the fleet is homogeneous. It was originally developed for the single-vehicle case by Archetti et al. [4] and was later extended to the multi-vehicle case by Coelho and Laporte [14].

Finally, we also tighten this formulation by imposing the following symmetry breaking constraints valid for the case where the vehicle fleet is homogeneous:

\[
\begin{align*}
y_0^{kt} &\leq y_0^{k-1,t} & k \in \mathcal{K}\backslash\{1\}, & t \in \mathcal{T} \quad (19) \\y_i^{kt} &\leq \sum_{j<i} y_j^{k-1,t} & i \in \mathcal{V}', & k \in \mathcal{K}\backslash\{1\}, & t \in \mathcal{T}. \quad (20)
\end{align*}
\]

Constraints (19) ensure that vehicle \( k \) cannot leave the depot if vehicle \( k - 1 \) is not used. This symmetry breaking rule is then extended to the customer vertices by constraints (20) which state that if customer \( i \) is assigned to vehicle \( k \) in period \( t \), then vehicle \( k - 1 \) must serve a customer with an index smaller than \( i \) in the same period. These constraints were also used by Coelho and Laporte [14] and are inspired from those proposed by Fischetti et al. [20] for the capacitated vehicle routing problem and by Albareda-Sambola et al. [2] for a plant location problem.

### 3.2 The MMIRP with consistency features

We now describe minor modifications to the model necessary to account for each of the consistency features described in Section 2.
3.2.1 Driver partial consistency

This feature can be handled in a number of ways. We have modeled it with an extra binary variable $z^k_i$ equal to 1 if and only if vehicle $k$ visits customer $i$. As in Coelho et al. [16] we add to the objective function a penalty term proportional to the number of extra vehicles assigned to each customer, and we introduce a binary variable $s^k_i$ indicating whether an extra vehicle $k$ is assigned to customer $i$. We then impose the following sets of constraints to the model:

$$\sum_{k \in K} z^k_i = 1 \quad i \in V'$$ (21)

$$y^k_{it} \leq z^k_i + s^k_i \quad i \in V', k \in K, t \in T$$ (22)

$$s^k_i, z^k_i \in \{0, 1\} \quad i \in V', k \in K.$$ (23)

Constraints (21) assign a first vehicle to each customer, while constraints (22) allow additional vehicles to be assigned to the same customer. We then add a penalty term

$$\alpha \sum_{i \in V'} \sum_{k \in K} s^k_i$$ (24)

to the objective function (1). By adjusting the parameter $\alpha$, one can control how restrictive the driver partial consistency policy will be. If $\alpha$ is sufficiently high, the model considers only one driver per customer, or a strict driver consistency rule. If it is equal to zero, the model reduces to the standard MMIRP without consistency.

3.2.2 Visit spacing

Adding the following constraints to the basic model will ensure that at least one visit will take place every $(V_i + 1)$ periods, and no more than one visit will take place in any $(v_i + 1)$ successive periods:

$$\sum_{k \in K} \sum_{l=t}^{t+v_i} y^k_{il} \leq 1 \quad i \in V', t \in \{1, ..., p - v_i\}$$ (25)

$$\sum_{k \in K} \sum_{l=t}^{t+V_i} y^k_{il} \geq 1 \quad i \in V', t \in \{1, ..., p - V_i\}.$$ (26)

4 Branch-and-Cut Algorithm

The MMIRP is $\mathcal{NP}$-hard since it contains the Vehicle Routing Problem (VRP) as a special case. If the instance size is not excessive, the proposed undirected formulation can be solved exactly by branch-and-cut as follows. We provide a sketch of our branch-and-cut implementation in Algorithm 1.

4.1 Branch-and-cut Scheme

At a generic node of the search tree, a linear program (LP) defined by (1)−(9) is solved, a search for violated subtour elimination constraints (10) is performed, and some of these constraints are added to the current program which is then reoptimized. This process is repeated until a feasible or dominated solution is reached, or until there are no more cuts to be added. At this point branching on a fractional variable occurs. We accelerate the process of finding good feasible upper bounds by solving an exact mixed-integer program (MIP) as follows.
In order to solve the MMIRP with driver partial consistency, the objective function is the sum of (1) and (24), subject to (2)–(14). For the MMIRP with visit spacing consistency, the LP is defined by (1)–(9), (25) and (26). All valid inequalities remains the same.

4.2 Solution Improvement Algorithm

The purpose of the Solution Improvement (SI) algorithm is to approximate the cost of a new solution resulting from vertex removals and reinsertions. It is solved exactly whenever the branch-and-cut search identifies a new best solution. Using an idea proposed by Archetti et al. [5], we simplify and approximate the routing costs resulting from vertex removals and reinsertions as follows. Let \( a_{kti} \) represent the routing cost reduction if customer \( i \) is removed from the route of vehicle \( k \) at period \( t \); let \( b_{kti} \) represent the routing cost if customer \( i \) is inserted in the route of vehicle \( k \) at period \( t \); finally, let \( r_{kti} \) be a binary parameter equal to 1 if and only if customer \( i \) is visited in the current route of vehicle \( k \) at period \( t \). Also define the following binary variables:

- Let \( u_{kti} \) be equal to 1 if and only if customer \( i \) is removed from the existing route of vehicle \( k \) at period \( t \), and
- Let \( v_{kti} \) be equal to 1 if and only if customer \( i \) is inserted in the route of vehicle \( k \) at period \( t \). This subproblem is then to

\[
\text{(SI)} \quad \text{minimize} \quad \sum_{i \in V} \sum_{m \in M} \sum_{t \in T} h_{m}^i I_{m}^i t + \sum_{i \in V'} \sum_{k \in K} \sum_{t \in T} (b_{kti} v_{kti} - a_{kti} u_{kti})
\]

subject to (2)–(6) and to

\[
q_{mkt} \leq (r_{kti} - u_{kti} + v_{kti})C_i \quad i \in V' \quad m \in M \quad k \in K \quad t \in T
\]

\[
v_{kti} \leq 1 - r_{kti} \quad i \in V' \quad k \in K \quad t \in T
\]

\[
u_{kti} \leq r_{kti} \quad i \in V' \quad k \in K \quad t \in T
\]

\[
\sum_{i \in V'} \sum_{m \in M} q_{mkt} \leq Q_k \quad k \in K \quad t \in T
\]

\[
q_{mkt} \geq 0 \quad i \in V' \quad m \in M \quad k \in K \quad t \in T
\]

\[
u_{kti}, v_{kti} \in \{0, 1\} \quad i \in V' \quad k \in K \quad t \in T.
\]

The objective function (27) minimizes the total inventory, removal and insertion cost. Constraints (28) enforce the ML policy. Constraints (29) ensure that if a customer is already present in a route, it cannot be reinserted in the same route. Likewise, constraints (30) guarantee that only those customers belonging to a route can be removed from it. Constraints (32) ensure that the vehicle capacity is respected. If the incumbent solution is changed by more than one customer, then this model only provides an approximation of the actual routing costs. For this reason, we have decided to limit the number of insertions and removals that could take place in the solution of SI, and we have added constraints (31) to limit the number of insertions and removals per route to a small value \( \varepsilon \).

4.2.1 SI for the driver partial consistency

The driver partial consistency is also modeled in SI with a binary variable \( s_{kt} \) and a penalty in the objective function, as above. The variable \( s_{kt} \) will be equal to one if and only if an extra vehicle \( k \) is assigned to customer \( i \). The required constraints are

\[
\sum_{k \in K} z_{kt} = 1 \quad i \in V', k \in K, t \in T
\]
The penalty to the objective function is added in the same fashion as in Section 3.2.1.

4.2.2 SI for the visit spacing consistency

The imposition of minimum and maximum intervals between visits is modeled by adding the following sets of constraints to the SI model:

\[
\sum_{k \in K} \sum_{l = t}^{t+m} (r_{ik}^t - u_{ik}^t + v_{ik}^t) \leq 1 \quad i \in V', t \in \{1, ..., p - v_i\} \tag{38}
\]

\[
\sum_{k \in K} \sum_{l = t}^{t+M} (r_{ik}^t - u_{ik}^t + v_{ik}^t) \geq 1 \quad i \in V', t \in \{1, ..., p - V_i\} \tag{39}
\]

Algorithm 1 Proposed branch-and-cut algorithm

1: Randomly generate a starting solution and apply SI to it.
2: At the root node of the search tree, generate and insert all valid inequalities (15) – (20) into the program.
3: Subproblem solution. Solve the LP relaxation of the node.
4: Termination check:
5: if there are no more nodes to evaluate then
6: Stop.
7: else
8: if The current solution is a new best solution then
9: Apply the SI algorithm to the incumbent solution.
10: if the SI algorithm yields an improved solution then
11: Update the solution vector at the branch-and-cut level
12: end if
13: end if
14: Select one node from the branch-and-bound tree.
15: end if
16: while the solution of the current LP relaxation contains subtours do
17: Identify the connected components using the separation procedure of Padberg and Rinaldi [26].
18: Add all violated subtour elimination constraints (10).
19: Subproblem solution. Solve the LP relaxation of the node.
20: end while
21: if the solution of the current LP relaxation is integer then
22: Go to the termination check.
23: else
24: Branching: branch on one of the fractional variables.
25: Go to the termination check.
26: end if

5 Computational Experiments

We now describe some details related to the computational experiments used to evaluate our algorithm. All computations were carried out on a grid of Intel Xeon processors running at 2.66 GHz with up to 48 GB
of RAM installed per node, with the Scientific Linux 6.1 operating system. A single thread was used. The algorithms just described were coded in C++ and we use IBM Concert Technology and CPLEX 12.4 as the MIP solver. The generation of the instances is described in Section 5.1 and detailed computational results are provided in Section 5.2.

5.1 Generation of the Instances

We have generated a large data set varying the number of customers, products, vehicles and planning horizon. Our test bed is generated as follows:

- number of customers $n$: $10c$ where $c = 1, 2, 3, 4, 5$;
- number of products $M$: equal to 1, 3, or 5 products;
- number of vehicles $K$: equal to 1, 3, or 5 vehicles;
- horizon $p$: equal to 3, 5, or 7 periods;
- customer demand $d_{mt}^i$: drawn randomly from a discrete uniform distribution in the interval $[10, 100]$;
- product availability at the supplier $r_{mt}$: $f_{mt}n$, where $f_{mt}$ is drawn randomly from a discrete uniform distribution in the interval $[50, 140]$;
- maximum inventory level $C_i$: $mg_if_i$, where $g_i$ is randomly selected from the set $\{2, 3, 4\}$ and $f_i$ is drawn randomly from a discrete uniform distribution in the interval $[150, 200]$;
- starting inventory level $I_{m0}^i$: drawn randomly from a discrete uniform distribution in the interval $[100, 150]$;
- maximum inventory level $I_{m0}^i$: $\sum_i I_{m0}^i$;
- starting inventory level $I_{m0}^0$: $\sum_i I_{m0}^i$;
- inventory holding cost $h_{m0}^i$: $0.01$;
- inventory holding cost $h_{m0}^i$ ($i > 0$): drawn randomly from a continuous uniform distribution in the interval $[0.02, 0.20]$;
- vehicle capacity $Q$: $2 \lceil \sum \sum d_{mt} / pk \rceil$
- distance/cost $c_{ij}$: $\lceil \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} + 0.5 \rceil$, where the points $(X_i, Y_i)$ are the coordinates of vertex $i$ and are obtained randomly from a discrete uniform distribution in the interval $[0, 1000]$.

For each combination of $n, M, K, p$, we have generated five instances, yielding a total of 675 instances. This set of instances as well as the solutions presented next are available at the URL http://www.leandro-coelho.com/instances/.

5.2 Computational Experiments

We now report solutions for the MMIRP without consistency features in Section 5.2.1 and for the MMIRP with consistency features in Section 5.2.2.
5.2.1 Computational Results for the MMIRP

We organize the presentation of the computational results based on the number of products, which is the main feature of this paper. We compare the solutions obtained for different combinations of the parameters. Specifically, we present in Tables 1–3 average solution values for all combination of the remaining parameters, namely the time horizon \( p = 1, 3, \) and 5, and the number of vehicle \( K = 1, 3, \) and 5. In each table we present averages comprising the five instances for each number of customers. We show the average percentage gap, the number of instances solved to optimality and the average running time.

Table 1 shows that the average size of instances solved to optimality are similar to those of previous studies on the single product IRP [14, 1]. Comparing these figures with those of Tables 2 and 3 indicates that the addition of more products to the problem does not increase its difficulty. This is because the number of binary variables remain unchanged if more products are added to the problem. For instance, for a problem with 50 customers, one product, five vehicles and seven time periods, the number of binary variables is 41,740 and the number of general variables is slightly over 15,000. For the same instance with five products, approximately 10,000 continuous variables are added to the problem, but no extra binary variable are needed. These explains the relative ease of solving problems with multiple products. For the sake of comparison, for this instance size, each time period requires approximately 8,000 binary variables and each vehicle requires approximately 10,000 binary variables.

5.2.2 Computational Results for the MMIRP with Consistency

We have also solved a subset of instances with the two consistency features presented in Section 3.2.1 and 3.2.2. We have applied these consistency features to the instances with 30 customers, and all combinations of three and five periods, vehicles, and products. These instances are significantly difficult since many of them could not be solved to optimality in the case without consistency. We compare the solution cost of each set of instances with respect to the base case without consistency requirements. Specifically, in Tables 4 and 5 we provide the average gap after solving these instances with a 12-hour time limit, the average running time and the average cost increase of each set of instances with respect to the solutions obtained without consistency features presented in the previous section.

In line with the observations of Coelho and Laporte [14], ensuring the driver partial consistency feature does not increase the solution cost by much. It remains a meaningful way of increasing the quality of the solutions provided by the IRP both to customers and suppliers in a multi-product environment. The visit spacing consistency feature has shown, once again and for the first time in a multi-product framework, that is helps reduce the search space while providing meaningful solutions. Indeed, depending on the values chosen for its intervals the solutions it provides are valid for the most general case without consistency, but the introduction of this feature significantly reduces the time needed to obtain such good solutions. This is the case, for example, when solving instances with five products and five periods, which are very difficult for the general case, but for which the MMIRP with visit spacing consistency was able to find better solutions. This shows that some of these consistency features are beneficial not only from a business perspective but also from a computational point of view.

6 Conclusions

We have developed for the first time a simple yet powerful branch-and-cut algorithm to solve the IRP with multi-product and multi-vehicle. Our formulation is flexible and our algorithm is able to solve the IRP with one or several vehicles, and with many products, each with a specific demand, but sharing inventory and vehicle capacities. We have also solved instances consistency features by imposing some regularity in the the distribution process. We have proposed a large set of benchmark instances ranging from 10 to 50 customers, with up to seven time periods, five vehicles and five products on which we confirm the success of our exact algorithm.
Table 1: Summary of computational results for the MMIRP with one product ($M = 1$)

<table>
<thead>
<tr>
<th>Horizon</th>
<th># customers</th>
<th>$K = 1$</th>
<th></th>
<th>$K = 3$</th>
<th></th>
<th>$K = 5$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Gap (%)</td>
<td># solved</td>
<td>Time (s)</td>
<td>Gap (%)</td>
<td># solved</td>
<td>Time (s)</td>
</tr>
<tr>
<td>$p = 3$</td>
<td>10</td>
<td>0.00</td>
<td>5</td>
<td>0.00</td>
<td>0.00</td>
<td>5</td>
<td>6.60</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.00</td>
<td>5</td>
<td>1.60</td>
<td>0.00</td>
<td>5</td>
<td>184.00</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>0.00</td>
<td>5</td>
<td>13.20</td>
<td>0.00</td>
<td>5</td>
<td>5113.40</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>0.00</td>
<td>5</td>
<td>101.80</td>
<td>2.92</td>
<td>3</td>
<td>28221.60</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.00</td>
<td>5</td>
<td>299.60</td>
<td>19.44</td>
<td>0</td>
<td>43200.00</td>
</tr>
<tr>
<td>$p = 5$</td>
<td>10</td>
<td>0.00</td>
<td>5</td>
<td>1.00</td>
<td>0.22</td>
<td>4</td>
<td>12555.40</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.00</td>
<td>5</td>
<td>20.40</td>
<td>17.04</td>
<td>0</td>
<td>41683.20</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>0.00</td>
<td>5</td>
<td>182.20</td>
<td>31.16</td>
<td>0</td>
<td>43181.00</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>0.00</td>
<td>4</td>
<td>9464.60</td>
<td>42.11</td>
<td>0</td>
<td>43188.80</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.56</td>
<td>4</td>
<td>20712.60</td>
<td>48.12</td>
<td>0</td>
<td>43197.20</td>
</tr>
<tr>
<td>$p = 7$</td>
<td>10</td>
<td>0.00</td>
<td>5</td>
<td>27.80</td>
<td>14.40</td>
<td>0</td>
<td>28173.20</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.00</td>
<td>5</td>
<td>3335.60</td>
<td>27.67</td>
<td>0</td>
<td>41297.40</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>3.06</td>
<td>2</td>
<td>28529.00</td>
<td>33.60</td>
<td>0</td>
<td>43171.40</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>6.03</td>
<td>0</td>
<td>43200.00</td>
<td>61.12</td>
<td>0</td>
<td>43200.00</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>9.97</td>
<td>0</td>
<td>43187.80</td>
<td>62.17</td>
<td>0</td>
<td>43200.00</td>
</tr>
</tbody>
</table>
Table 2: Summary of computational results for the MMIRP with three products ($M = 3$)

<table>
<thead>
<tr>
<th>Horizon</th>
<th># customers</th>
<th>$K = 1$</th>
<th></th>
<th>$K = 3$</th>
<th></th>
<th>$K = 5$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Gap (%)</td>
<td># solved</td>
<td>Time (s)</td>
<td>Gap (%)</td>
<td># solved</td>
<td>Time (s)</td>
</tr>
<tr>
<td>$p = 3$</td>
<td>10</td>
<td>0.00</td>
<td>5</td>
<td>1.60</td>
<td>0.00</td>
<td>5</td>
<td>5.80</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.00</td>
<td>5</td>
<td>0.80</td>
<td>0.00</td>
<td>5</td>
<td>322.60</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>0.00</td>
<td>5</td>
<td>1.00</td>
<td>0.00</td>
<td>5</td>
<td>3574.60</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>0.00</td>
<td>5</td>
<td>5.20</td>
<td>0.26</td>
<td>4</td>
<td>12952.60</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>0.00</td>
<td>5</td>
<td>11.40</td>
<td>2.27</td>
<td>2</td>
<td>35501.60</td>
</tr>
<tr>
<td>$p = 5$</td>
<td>10</td>
<td>0.00</td>
<td>5</td>
<td>3.40</td>
<td>0.00</td>
<td>5</td>
<td>8008.20</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.00</td>
<td>5</td>
<td>45.80</td>
<td>17.05</td>
<td>0</td>
<td>42202.40</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>0.00</td>
<td>5</td>
<td>506.20</td>
<td>20.67</td>
<td>0</td>
<td>43196.00</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>0.00</td>
<td>5</td>
<td>6967.80</td>
<td>23.57</td>
<td>0</td>
<td>43188.60</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>2.72</td>
<td>1</td>
<td>40242.60</td>
<td>45.54</td>
<td>0</td>
<td>18555.20</td>
</tr>
<tr>
<td>$p = 7$</td>
<td>10</td>
<td>0.00</td>
<td>5</td>
<td>23.80</td>
<td>21.94</td>
<td>0</td>
<td>27184.00</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.00</td>
<td>5</td>
<td>6173.40</td>
<td>29.36</td>
<td>0</td>
<td>43197.60</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>5.41</td>
<td>0</td>
<td>43200.00</td>
<td>30.89</td>
<td>0</td>
<td>43197.80</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>11.04</td>
<td>0</td>
<td>43200.00</td>
<td>38.35</td>
<td>0</td>
<td>43200.00</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>15.15</td>
<td>0</td>
<td>43199.80</td>
<td>54.69</td>
<td>0</td>
<td>43200.00</td>
</tr>
</tbody>
</table>
Table 3: Summary of computational results for the MMIRP with five products ($M = 5$)

<table>
<thead>
<tr>
<th>Horizon # customers</th>
<th>$K = 1$</th>
<th></th>
<th></th>
<th>$K = 3$</th>
<th></th>
<th></th>
<th>$K = 5$</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gap (%)</td>
<td># solved</td>
<td>Time (s)</td>
<td>Gap (%)</td>
<td># solved</td>
<td>Time (s)</td>
<td>Gap (%)</td>
<td># solved</td>
<td>Time (s)</td>
</tr>
<tr>
<td>$p = 3$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.00</td>
<td>5</td>
<td>4.60</td>
<td>0.00</td>
<td>5</td>
<td>5.00</td>
<td>0.00</td>
<td>5</td>
<td>64.00</td>
</tr>
<tr>
<td>20</td>
<td>0.00</td>
<td>5</td>
<td>1.00</td>
<td>0.00</td>
<td>5</td>
<td>520.00</td>
<td>0.60</td>
<td>4</td>
<td>9813.20</td>
</tr>
<tr>
<td>30</td>
<td>0.00</td>
<td>5</td>
<td>2.00</td>
<td>0.81</td>
<td>3</td>
<td>17738.80</td>
<td>3.13</td>
<td>2</td>
<td>37412.20</td>
</tr>
<tr>
<td>40</td>
<td>0.00</td>
<td>5</td>
<td>6.40</td>
<td>0.71</td>
<td>4</td>
<td>14205.40</td>
<td>5.55</td>
<td>0</td>
<td>43199.40</td>
</tr>
<tr>
<td>50</td>
<td>0.00</td>
<td>5</td>
<td>19.00</td>
<td>1.14</td>
<td>2</td>
<td>33966.20</td>
<td>12.14</td>
<td>0</td>
<td>41009.60</td>
</tr>
<tr>
<td>$p = 5$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.00</td>
<td>5</td>
<td>7.40</td>
<td>5.10</td>
<td>3</td>
<td>16428.00</td>
<td>17.73</td>
<td>0</td>
<td>43122.00</td>
</tr>
<tr>
<td>20</td>
<td>0.00</td>
<td>5</td>
<td>95.20</td>
<td>15.28</td>
<td>0</td>
<td>43200.00</td>
<td>30.16</td>
<td>0</td>
<td>43200.00</td>
</tr>
<tr>
<td>30</td>
<td>0.00</td>
<td>5</td>
<td>5670.40</td>
<td>22.49</td>
<td>0</td>
<td>43188.60</td>
<td>39.07</td>
<td>0</td>
<td>43200.00</td>
</tr>
<tr>
<td>40</td>
<td>2.11</td>
<td>2</td>
<td>32088.60</td>
<td>30.40</td>
<td>0</td>
<td>43194.60</td>
<td>54.65</td>
<td>0</td>
<td>43200.00</td>
</tr>
<tr>
<td>50</td>
<td>7.49</td>
<td>0</td>
<td>43194.00</td>
<td>45.56</td>
<td>0</td>
<td>42173.00</td>
<td>54.34</td>
<td>0</td>
<td>41344.60</td>
</tr>
<tr>
<td>$p = 7$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.00</td>
<td>5</td>
<td>53.60</td>
<td>22.77</td>
<td>0</td>
<td>26783.20</td>
<td>29.57</td>
<td>0</td>
<td>42114.00</td>
</tr>
<tr>
<td>20</td>
<td>0.33</td>
<td>4</td>
<td>18773.60</td>
<td>30.34</td>
<td>0</td>
<td>43200.00</td>
<td>40.03</td>
<td>0</td>
<td>43200.00</td>
</tr>
<tr>
<td>30</td>
<td>9.39</td>
<td>4</td>
<td>43195.00</td>
<td>30.84</td>
<td>0</td>
<td>43200.00</td>
<td>44.05</td>
<td>0</td>
<td>43185.40</td>
</tr>
<tr>
<td>40</td>
<td>12.30</td>
<td>0</td>
<td>43200.00</td>
<td>62.96</td>
<td>0</td>
<td>43192.00</td>
<td>54.70</td>
<td>0</td>
<td>43200.00</td>
</tr>
<tr>
<td>50</td>
<td>17.64</td>
<td>0</td>
<td>43194.80</td>
<td>55.65</td>
<td>0</td>
<td>43200.00</td>
<td>55.06</td>
<td>0</td>
<td>43200.00</td>
</tr>
</tbody>
</table>
Table 4: Summary of computational results for the MMIRP with driver partial consistency ($n = 30$)

<table>
<thead>
<tr>
<th>Products</th>
<th>Horizon</th>
<th>$K = 3$</th>
<th>Gap (%)</th>
<th>Time (s)</th>
<th>% increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M = 3$</td>
<td>$p = 3$</td>
<td></td>
<td>3.91</td>
<td>43200.0</td>
<td>0.07</td>
</tr>
<tr>
<td></td>
<td>$p = 5$</td>
<td></td>
<td>28.12</td>
<td>43185.2</td>
<td>6.17</td>
</tr>
<tr>
<td>$M = 5$</td>
<td>$p = 3$</td>
<td></td>
<td>2.63</td>
<td>30200.2</td>
<td>6.71</td>
</tr>
<tr>
<td></td>
<td>$p = 5$</td>
<td></td>
<td>25.24</td>
<td>43181.0</td>
<td>1.75</td>
</tr>
</tbody>
</table>

Table 5: Summary of computational results for the MMIRP with visit spacing consistency ($n = 30$)

<table>
<thead>
<tr>
<th>Products</th>
<th>Horizon</th>
<th>$K = 3$</th>
<th>Gap (%)</th>
<th>Time (s)</th>
<th>% increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M = 3$</td>
<td>$p = 3$</td>
<td></td>
<td>0.00</td>
<td>662.8</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>$p = 5$</td>
<td></td>
<td>21.33</td>
<td>43200.0</td>
<td>3.85</td>
</tr>
<tr>
<td>$M = 5$</td>
<td>$p = 3$</td>
<td></td>
<td>0.00</td>
<td>891.6</td>
<td>−0.10</td>
</tr>
<tr>
<td></td>
<td>$p = 5$</td>
<td></td>
<td>17.67</td>
<td>43200.0</td>
<td>−2.84</td>
</tr>
</tbody>
</table>

References


