CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

Consistency in Multi-Vehicle Inventory-Routing

Leandro Callegari Coelho
Jean-François Cordeau
Gilbert Laporte

October 2011

CIRRELT-2011-66

Bureaux de Montréal :
Université de Montréal
C.P. 6128, succ. Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Consistency in Multi-Vehicle Inventory-Routing

Leandro Callegari Coelho[1,2,*], Jean-François Cordeau[1,2], Gilbert Laporte[1,3]

[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. Inventory-routing problems (IRPs) arise in vendor-managed inventory systems. They require jointly solving a vehicle routing problem and an inventory management problem. Whereas the solutions they yield tend to benefit the vendor and customers, solving IRPs solely based on cost considerations may lead to inconveniences to both parties. These are related to the fleet size and vehicle load, to the frequency of the deliveries, and to the quantities delivered. In order to alleviate these problems, we introduce the concept of consistency in IRP solutions, thus increasing quality of service. We formulate the multi-vehicle IRP, with and without consistency requirements, as mixed integer linear programs, and we propose a matheuristic for their solution. This heuristic applies an adaptive large neighborhood search scheme in which some subproblems are solved exactly. The proposed algorithm generates solutions offering a good compromise between cost and quality. We analyze the effect of different inventory policies, routing decisions and delivery sizes.

Keywords. Vendor-managed inventory systems, inventory-routing, consistency, service quality, adaptive large neighborhood search, matheuristic.

Acknowledgements. This work was partly supported by the Natural Sciences and Engineering Council of Canada (NSERC) under grants 227837-09 and 39682-10. This support is gratefully acknowledged. The authors thank Andrew Goldberg and Boris Cherkassky for making their implementation of the scaling push-relabel minimum-cost cost flow algorithm available.

---

* Corresponding author: Leandro.Coelho@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2011

1 Introduction

In vendor-managed inventory (VMI) systems, the replenishment and distribution making process is centralized at the supplier's level. The application of this policy leads to an overall reduction of logistics costs [20] and is often described as a win-win situation. By deciding when and how much to deliver to their customers, suppliers can reduce their overall distribution costs by smoothing their delivery schedules and by efficiently combining in the same period visits to customers that are geographically close to one another. Customers also benefit by saving on ordering costs.

Optimizing a VMI system requires the solution of a difficult combinatorial optimization problem called the Inventory-Routing Problem (IRP). The IRP combines inventory management and routing decisions over several periods into the same problem. Typically, the supplier is free to decide the size of the delivery to each customer, being constrained only by the inventory holding capacity at each site and by the capacities of its vehicles. This general delivery policy is called maximum level (ML). Several heuristics [4, 10, 12] and an exact algorithm [3] have been proposed for the single vehicle case of this problem. A large number of variants of the IRP have arisen since this problem was first introduced by Bell et al. [9]. Literature reviews can be found in [2] and [13].

Whereas VMI policies are clearly beneficial from a system's perspective, they may sometimes result in inconveniences both to the supplier and to the customers. This is the case, for example, when very small deliveries take place on consecutive days, followed by a very large delivery, after which the customer is not visited for a long period. Another example, this time undesirable for the supplier, is that it could be optimal to dispatch a mix of almost full and almost empty vehicles, which does not yield a proper load balancing and may irritate some drivers.

Companies need not only provide cost effective solutions to their customers, but also high quality service. This can be partly achieved by incorporating quality of service features in IRP solutions, which should yield a competitive advantage. To this end, we introduce the concept of *consistency* in the IRP in order to reflect some common quality of service standards. This can be achieved, for example, through the application of workforce management policies [6, 18, 26]. Thus, one would expect that regularly assigning the same driver to customers will help create a bond that can benefit both parties. Drivers will gain an increased familiarity with the region and the customer sites assigned to them, and will thus develop a more personal rapport with the customers. Another example of consistency is the spacing of deliveries to customers. To ensure smoother operations, visits should ideally be spread out evenly over the planning horizon. This type of requirement is often modeled as constraints in the context of the periodic Vehicle Routing Problem (VRP) [11, 16] but it has not yet been imposed in the IRP. Finally, the quantities delivered to customers can also be controlled in order to avoid large variations over time, which are negatively perceived by customers [8]. In this paper, we consider six different consistency features in IRP solutions:

1. Quantity consistency: any delivery performed to a customer must lie within certain customer-dependent intervals, to avoid large variations. From the customers' point of view, delivery size is important. If deliveries are too small, then customers will have to receive frequent visits, which is inconvenient and time-consuming. Deliveries that are too large may create congestion in the warehouse.

2. Vehicle filling rate: a vehicle can only be used if its filling rate lies within a certain interval.

3. Order-up-to (OU) policy: this is a common IRP constraint (see e.g [3, 4, 5, 10, 12]) which can be viewed as a consistency feature. It states that whenever a visit is performed to a customer, the delivery should fill the customer's inventory capacity.

4. Driver consistency: this requirement means that each customer is assigned to one driver.

5. Driver partial consistency: one shortcoming of the previous feature is that it may cause a vehicle to serve very few customers and thus its effect may be very costly. We relax this rule by allowing some deliveries not to be subject to it.

6. Visit spacing: we impose a minimum and a maximum interval between two consecutive visits to the same customer.

Some of these features (e.g. 1 and 6) should depend on the stability of the demand. If the demand is highly variable, customers would expect deliveries to be variable as well, because consistency would then make little sense. However, it is known [7, 22] that the application of VMI requires some demand stability, which legitimates the consistency features we propose. Note that the concept of driver consistency has already been applied by Groër et al. [18] to a version of the VRP in which customers receive visits on prespecified days. The authors have proposed a model ensuring that the same customer is always served by the same driver as a means of improving quality of service, but the application of this constraint to the IRP is new and more complicated because the visit days are endogenous and because of the inventory management issues involved.

We model and solve the *basic* multi-vehicle version of the problem (MIRP) considered in [3], [4] and [10] to which we incorporate the consistency features just described. Although the MIRP has previously been studied, the variety of assumptions has left a gap in the literature in the sense that one cannot find benchmarks to a common version of the problem. For instance, to cite some recent contributions to the MIRP literature and their different assumptions, Abdelmaguid and Dessouky [1] allow backorders and use a non-linear transportation cost function which depends on the quantity delivered, Dauzère-Pérès et al. [14] have studied the stochastic version of the problem, and Yugang et al.

[27] did not include supplier inventory costs. Here we define and solve benchmark instances of the MIRP derived from those of [3, 4] for the single vehicle case, with and without consistency requirements.

The main scientific contribution of this paper is to add consistency requirements to the basic MIRP and to develop a matheuristic for this version of the MIRP, called the *consistent* MIRP. The remainder of the paper is organized as follows. In Section 2 we formally describe the basic MIRP and we present a mixed-integer linear programming formulation for it and for the consistent MIRP. Section 3 describes our algorithm which combines adaptive large neighborhood search and the exact solution of mixed integer linear programs. This algorithm can solve the basic MIRP and the consistent MIRP defined by any combination of the six features just introduced. This is followed by the results of extensive computational experiments in Section 4, and by conclusions in Section 5.

2 Formal problem description and mathematical models

We now formally introduce the basic MIRP. The problem is defined on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V} = \{0, ..., n\}$ is the vertex set and $\mathcal{A} = \{(i, j) : i, j \in \mathcal{V}, i \neq j\}$ is the arc set. Vertex 0 is a depot at which the supplier is located and the vertices of $\mathcal{V}' = \mathcal{V} \setminus \{0\}$ represent customers. The problem is defined over a planning horizon of length p and, at each time period $t \in \mathcal{T} = \{1, ..., p\}$, the quantity of product made available at the supplier is equal to r^t. A unit inventory holding cost h_i is incurred by customer i and by the supplier at each period, and customer i has an inventory holding capacity C_i. 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_0^t and I_i^t are defined as the inventory levels 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_0^0 and I_i^0 for $i \in \mathcal{V}'$), and has full knowledge of the demand d_i^t of each customer i for each time period t.

A set $\mathcal{K} = \{1, ..., K\}$ of vehicles are available. We denote by Q_k the capacity of vehicle k. Each vehicle is able to perform one route per time period, from the supplier to a subset of customers. A routing cost c_{ij} is associated with arc $(i, j) \in \mathcal{A}$.

The objective of the problem is to minimize the total routing and inventory holding cost while meeting the demand for each customer. The replenishment plan is subject to the following constraints:

- at the end of period t, the inventory at a customer location cannot exceed its maximum capacity;
- inventories are not allowed to be negative;

- the supplier's vehicles can each perform at most one route per time period;
- each route starts and ends at the depot;
- the vehicle capacities cannot be exceeded.

The solution to the problem specifies which customers to serve at each time period, which vehicle to use on each route, how much to deliver to each visited customer, and how to sequence customers on the vehicle routes. Throughout the paper, we assume that the quantity r^t becoming available at the supplier in period t can be used for deliveries to customers in the same period, and that the quantities q_i^{kt} received by customer i in period t can be used to meet the demand in that period.

The model works with the following binary variables: x_{ij}^{kt} is equal to 1 if and only if vertex j immediately follows vertex i on the route of vehicle k in period t, and y_i^{kt} is equal to 1 if and only if customer i is visited by vehicle k in period t. We denote by q_i^{kt} the quantity of product delivered from the supplier to customer i using vehicle k in time period t. The model also uses continuous variables w_i^{kt} to enforce the VRP subtour elimination constraints [15, 19]. They represent the sum of the deliveries made by vehicle k in period t after visiting customer i.

2.1 Mixed integer linear program for the basic MIRP

The mathematical model for the basic MIRP is as follows:

$$\text{(MIRP)} \quad \text{minimize} \sum_{t\in\mathcal{T}} h_0 I_0^t + \sum_{i\in\mathcal{V}'}\sum_{t\in\mathcal{T}} h_i I_i^t + \sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{V}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} c_{ij} x_{ij}^{kt} \tag{1}$$

subject to

$$I_0^t = I_0^{t-1} + r^t - \sum_{i\in\mathcal{V}'}\sum_{k\in\mathcal{K}} q_i^{kt} \qquad t\in\mathcal{T} \tag{2}$$

$$I_0^t \geq 0 \qquad t\in\mathcal{T} \tag{3}$$

$$I_i^t = I_i^{t-1} + \sum_{k\in\mathcal{K}} q_i^{kt} - d_i^t \qquad i\in\mathcal{V}', t\in\mathcal{T} \tag{4}$$

$$I_i^t \geq 0 \qquad i\in\mathcal{V}', t\in\mathcal{T} \tag{5}$$

$$I_i^t \leq C_i \qquad i\in\mathcal{V}', t\in\mathcal{T} \tag{6}$$

$$\sum_{k\in\mathcal{K}} q_i^{kt} \leq C_i - I_i^{t-1} \qquad i\in\mathcal{V}', t\in\mathcal{T} \tag{7}$$

$$\sum_{k\in\mathcal{K}} q_i^{kt} \leq C_i \sum_{j\in\mathcal{V}}\sum_{k\in\mathcal{K}} x_{ij}^{kt} \qquad i\in\mathcal{V}', t\in\mathcal{T} \tag{8}$$

$$\sum_{i \in \mathcal{V}'} q_i^{kt} \leq Q_k \qquad t \in \mathcal{T}, k \in \mathcal{K} \tag{9}$$

$$q_i^{kt} \leq y_i^{kt} C_i \qquad i \in \mathcal{V}', t \in \mathcal{T}, k \in \mathcal{K} \tag{10}$$

$$\sum_{j \in \mathcal{V}} x_{ij}^{kt} = \sum_{j \in \mathcal{V}} x_{ji}^{kt} = y_i^{kt} \qquad i \in \mathcal{V}', t \in \mathcal{T}, k \in \mathcal{K} \tag{11}$$

$$\sum_{j \in \mathcal{V}'} x_{0j}^{kt} \leq 1 \qquad k \in \mathcal{K} \quad t \in \mathcal{T} \tag{12}$$

$$\sum_{k \in \mathcal{K}} y_i^{kt} \leq 1 \qquad i \in \mathcal{V}', t \in \mathcal{T} \tag{13}$$

$$w_i^{kt} - w_j^{kt} + Q_k x_{ij}^{kt} \leq Q_k - q_j^{kt} \qquad i \in \mathcal{V}', j \in \mathcal{V}', t \in \mathcal{T}, k \in \mathcal{K} \tag{14}$$

$$q_i^{kt} \leq w_i^{kt} \leq Q_k \qquad i \in \mathcal{V}', t \in \mathcal{T}, k \in \mathcal{K} \tag{15}$$

$$q_i^{kt} \geq 0 \qquad i \in \mathcal{V}', j \in \mathcal{V}, t \in \mathcal{T}, k \in \mathcal{K} \tag{16}$$

$$x_{ij}^{kt}, y_i^{kt} \in \{0,1\} \qquad i, j \in \mathcal{V}, i \neq j, t \in \mathcal{T}, k \in \mathcal{K}. \tag{17}$$

In this model, the objective function is the sum of inventory costs at the supplier and customer locations, and of routing costs. Constraints (2) define the inventory at the supplier carried at the end of period t. Constraints (3) forbid stockouts at the supplier. Constraints (4) and (5) are similar to (2) and (3) but apply to the customers. Constraints (6) define the maximum inventory level at customer locations, while constraints (7) and (8) ensure that the quantity delivered to customer i at period t will not exceed the customer's inventory capacity if the customer is served, and will be zero otherwise. Constraints (9) mean that vehicle capacities are never exceeded. Constraints (10)−(15) impose linking and routing conditions. In particular, constraints (14) ensure the consistency of the load of each vehicle along its route and prevent subtours. Finally, constraints (16) and (17) enforce the non-negativity and integrality requirements.

2.2 Modeling the features of the consistent MIRP

We now formally describe the features of six versions of the consistent MIRP and we show how they can be modeled separately or jointly.

2.2.1 Quantity consistency

A way to ensure that all deliveries to a given customer will be consistent over time is to force the delivery amounts to lie within an interval $[g_l,\ g_u]$ around a target value equal to the average demand of the customer over the planning horizon:

$$y_i^{kt} g_l \sum_{t \in \mathcal{T}} d_i^t / p \leq q_i^{kt} \leq y_i^{kt} g_u \sum_{t \in \mathcal{T}} d_i^t / p \qquad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T}. \tag{18}$$

2.2.2 Vehicle filling rate

To balance the load between vehicles and to avoid dispatching vehicles with very low loads, we impose a vehicle filling rate constraint which specifies that a vehicle can only be used if the total quantity it delivers fills at least a fraction γ of its capacity. This is achieved by adding the following constraint to the basic model:

$$\sum_{i \in \mathcal{V}'} q_i^{kt} \geq \gamma \sum_{i \in \mathcal{V}'} x_{0i}^{kt} Q_k \qquad k \in \mathcal{K}, t \in \mathcal{T}. \tag{19}$$

2.2.3 Order-up-to policy

Under an OU inventory policy, the decisions of when and how much to deliver to a customer are linked: whenever a customer is visited, the quantity delivered must fill the customer's inventory capacity. The OU policy is imposed through the constraints

$$q_i^{kt} \geq C_i \sum_{j \in \mathcal{V}} x_{ij}^{kt} - I_i^{t-1} \qquad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T}. \tag{20}$$

2.2.4 Driver consistency

Driver consistency is modeled with an extra binary variable z_i^k equal to 1 if and only if vehicle k visits customer i. Then, two sets of constraints are added to the basic model:

$$\sum_{k \in \mathcal{K}} z_i^k = 1 \qquad i \in \mathcal{V}' \tag{21}$$

$$y_i^{kt} \leq z_i^k \qquad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T} \tag{22}$$

$$z_i^k \in \{0, 1\} \qquad i \in \mathcal{V}', k \in \mathcal{K}. \tag{23}$$

Constraints (21) ensure that exactly one vehicle is assigned to each customer over the planning horizon. Constraints (22) allow deliveries only from the vehicle assigned to the customer.

2.2.5 Driver partial consistency

It may sometimes be preferable to apply a partially consistent policy by which a large number of deliveries follow the driver consistency rule, but in some cases the rule may be relaxed. We have modeled this policy by adding to the objective function a penalty term proportional to the number of extra vehicles assigned to each customer, beyond their regular vehicle. We have introduced a binary variable s_i^k indicating whether an extra vehicle k is assigned to customer i, and we impose the following sets of constraints to the basic model:

$$\sum_{k \in \mathcal{K}} z_i^k = 1 \qquad i \in \mathcal{V}' \tag{24}$$

$$y_i^{kt} \leq z_i^k + s_i^k \qquad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T} \tag{25}$$

$$s_i^k, z_i^k \in \{0,1\} \qquad i \in \mathcal{V}', k \in \mathcal{K}. \tag{26}$$

Constraints (24) assign a first vehicle to each customer, while constraints (25) allow additional vehicles to be assigned to the same customer. We then add a penalty term

$$\alpha \sum_{i \in \mathcal{V}'} \sum_{k \in \mathcal{K}} s_i^k \tag{27}$$

to the objective function (1). By adjusting the parameter α, one can control how restrictive the driver partial consistency policy will be.

2.2.6 Visit spacing

One may also want to enforce a minimum and maximum time interval between two consecutive visits to the same customer, since it may be undesirable to visit the same customer on several successive days or to leave a customer unvisited for a long period. Adding the following constraints to the basic model will ensure that at least one visit will take place every $(M_i + 1)$ periods, and no more than one visit will take place in any $(m_i + 1)$ successive periods. In practice, both M_i and m_i should depend on the capacity and on the demand of customer i:

$$\sum_{k \in \mathcal{K}} \sum_{l=t}^{t+m_i} y_i^{kl} \leq 1 \qquad i \in \mathcal{V}', t \in \{1, ..., p - m_i\} \tag{28}$$

$$\sum_{k \in \mathcal{K}} \sum_{l=t}^{t+M_i} y_i^{kl} \geq 1 \qquad i \in \mathcal{V}', t \in \{1, ..., p - M_i\}. \tag{29}$$

3 A matheuristic for the consistent MIRP

The MIRP is $\mathcal{NP}$-hard since it generalizes the capacitated VRP. As a result, the models described in Section 2 can only be used for the exact solution of relatively small instances. For this reason, we have opted to solve the problem heuristically. The heuristic we have developed can solve the basic MIRP and any combination of the six versions of the consistent MIRP just defined. It applies an Adaptive Large Neighborhood Search (ALNS) scheme in which some subproblems are solved exactly as MILPs. It can therefore be described as a *matheuristic* [21], i.e. as a hybridization of a heuristic and of a mathematical programming algorithm.

3.1 Adaptive Large Neighborhood Search

Our ALNS heuristic follows the general framework proposed by [24] and works as follows. At each iteration, a number of customers are removed from their current route by a destroy operator and are eventually reinserted back elsewhere

by a repair operator. The choice of an operator is governed by a roulette-wheel mechanism. Each operator i is assigned a weight ω_i whose value depends on its past performance, as well as a score. Given h operators with weights ω_i, operator j will be selected with probability $\omega_j / \sum_{i=1}^{h} \omega_i$. Initially, all weights are equal to one and all scores are equal to zero. At each iteration, the score of the selected operator is increased by σ_1 if it finds a new best solution, by σ_2 if it finds a solution better than the incumbent, and by σ_3 if the solution is not better but is still accepted. Obviously $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq 0$. The search is divided into segments of φ iterations each, after which the weights and scores are updated as follows. Let π_i and o_{ij} be, respectively, the score of operator i and the number of times it has been used in the last segment j, normalized by a factor $\nu_i \geq 1$ reflecting the computational effort it requires (see [12, 24]). The *normalization factor* ν_i multiplies o_{ij}, and therefore decreases the weight of operator i, so that the more time consuming operators are applied less frequently. The values used for the normalization factors are all equal to one in our implementation, except for two cases where different values are used. These are provided in Sections 3.1.1 and 3.1.2. The updated weights are then

$$\omega_i := \begin{cases} \omega_i & \text{if } o_{ij} = 0 \\ (1-\eta)\omega_i + \eta \pi_i / \nu_i o_{ij} & \text{if } o_{ij} \neq 0, \end{cases} \tag{30}$$

where $\eta \in [0, 1]$ is called the reaction factor, controling how quickly the weight adjustment reacts to changes in the movement performance (see Section 3.3). All scores are reset to zero.

As in [24] we use the same acceptance criterion as in simulated annealing: given a solution s, a neighbor solution s' is accepted if $z(s') < z(s)$, and with probability $e^{-(z(s')-z(s))/\tau}$ otherwise, where $z(s)$ is the solution cost and $\tau > 0$ is the current temperature. The temperature is initialized at τ_{start} and is decreased by a cooling rate factor ϕ at each iteration, where $0 < \phi < 1$.

Our computational tests have shown that the initial solution does not have a significant impact on the overall solution cost or on the running time. We therefore initialize the search with a randomly generated solution.

3.1.1 Destroy operators

1. **Randomly remove ρ**: This operator randomly selects one period and one vehicle and removes one randomly selected customer from it. It is repeated ρ times. The operator is useful for refining the solution, since it does not change it much when ρ is small (which happens frequently), but still yields a major transformation when ρ is large.

2. **Remove worst ρ**: This operator removes the customer that will save the most when removed, considering the total routing and inventory cost. It is applied ρ times. Its normalization factor is 20.

3. **Shaw removal**: Following the ideas developed in [23] and [25], this operator removes customers that are relatively close to each other. Specifically, it randomly selects one vehicle, one period and one customer served in this period, it computes the distance $dist_{min}$ to the closest customer also being served by the same route, and it removes all customers within $2dist_{min}$ units from the selected route.

4. **Avoid consecutive visits**: This operator is based on our observation that good solutions often do not contain visits to the same customer on two consecutive periods. Then, the operator verifies whether any customer is visited on two consecutive periods and removes the latest visit.

5. **Empty one period**: This operator selects one random period and empties all routes performed during that period.

6. **Empty one vehicle**: This operator selects one random vehicle and empties all routes performed by this vehicle.

3.1.2 Repair operators

1. **Randomly insert ρ**: This operator randomly inserts ρ customers into the current solution. Specifically, it selects one random customer, one random vehicle and one random period, and inserts the customer into the route of that vehicle in that period if it is not already routed in the same period. This operator is applied ρ times.

2. **Insert best ρ**: This operator is analogous to the previous one. It is applied ρ times by computing the cheapest insertion with respect to the total cost. The normalization factor used for this operator is 20.

3. **Shaw insertions**: This operator is similar to the Shaw removal operator in the sense that it selects similar customers to be inserted together. It selects one vehicle, one period and one customer not served in that period by any vehicle. The operator then computes $dist_{min}$ and all customers within a $2dist_{min}$ distance are inserted in the same route, always following the cheapest insertion rule.

4. **Swap ρ customers**: This operator selects two customers from two different routes and swaps their assignments, following the cheapest insertion rule. It is also applied ρ times.

3.2 Exact subproblem solutions

Our matheuristic embeds the exact solution of two subproblems. The first one, called Delivery Quantities (DQ) optimizes the delivery quantities associated with a given set of vehicle routes. It is solved every time a new routing solution is computed by the ALNS mechanism. It uses a binary parameter $\bar{x}_{ij}^{kt}$ equal to one if and only if customer j follows customer i in the route of vehicle k in

period t. As shown in [12], DQ can be formulated as the following network flow problem.

$$\text{(DQ)} \quad \text{minimize} \sum_{t\in\mathcal{T}} h_0 I_0^t + \sum_{i\in\mathcal{V}'}\sum_{t\in\mathcal{T}} h_i I_i^t \tag{31}$$

subject to

$$I_0^t = I_0^{t-1} + r^t - \sum_{i\in\mathcal{V}'}\sum_{k\in\mathcal{K}} q_i^{k,t} \qquad t\in\mathcal{T} \tag{32}$$

$$I_i^t = I_i^{t-1} + \sum_{k\in\mathcal{K}} q_i^{k,t} - d_i^t \qquad i\in\mathcal{V}', t\in\mathcal{T} \tag{33}$$

$$I_0^t \geq 0 \qquad t\in\mathcal{T} \tag{34}$$

$$I_i^t \geq 0 \qquad i\in\mathcal{V}', t\in\mathcal{T} \tag{35}$$

$$I_i^t \leq C_i \qquad i\in\mathcal{V}', t\in\mathcal{T} \tag{36}$$

$$\sum_{k\in\mathcal{K}} q_i^{kt} \leq C_i - I_i^{t-1} \qquad i\in\mathcal{V}', t\in\mathcal{T} \tag{37}$$

$$\sum_{k\in\mathcal{K}} q_i^{kt} \leq C_i \sum_{j\in\mathcal{V}}\sum_{k\in\mathcal{K}} \bar{x}_{ij}^{kt} \qquad i\in\mathcal{V}', t\in\mathcal{T} \tag{38}$$

$$\sum_{i\in\mathcal{V}'} q_i^{kt} \leq Q_k \qquad t\in\mathcal{T}, k\in\mathcal{K}. \tag{39}$$

Constraints (32) and (33) define the flow conservation. Lower and upper bounds on the flows are defined by (34)−(38). Vehicle capacity constraints (39) still define an upper bound on the quantity delivered by the vehicle, even though the customers to be visited are now fixed.

The purpose of the second subproblem, called Solution Improvement (SI), is to approximate the cost of a new solution resulting from vertex removals and reinsertions. This problem is no longer a network flow problem. It is solved every θ iterations or whenever a new best solution has been identified. Using an idea proposed by [4], we simplify and approximate the routing costs resulting from vertex removals and reinsertions as follows. Let a_i^{kt} represent the routing cost reduction if customer i is removed from the route of vehicle k at period t, which obviously visits customer i; let b_i^{kt} represent the routing cost if customer i is inserted in the route of vehicle k at period t, which obviously does not already visit customer i; finally, let r_i^{kt} 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_i^{kt} 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_i^{kt} 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

$$\text{(SI)} \quad \text{minimize} \sum_{t\in\mathcal{T}} h_0 I_0^t + \sum_{i\in\mathcal{V}'}\sum_{t\in\mathcal{T}} h_i I_i^t - \sum_{i\in\mathcal{V}'}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} a_i^{kt} u_i^{kt} + \sum_{i\in\mathcal{V}'}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} b_i^{kt} v_i^{kt} \tag{40}$$

subject to (2)−(6) and

$$q_i^{kt} \leq C_i - I_i^{t-1} \qquad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T} \tag{41}$$

$$q_i^{kt} \leq (r_i^{kt} - u_i^{kt} + v_i^{kt})C_i \qquad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T} \tag{42}$$

$$v_i^{kt} \leq 1 - r_i^{kt} \quad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T} \tag{43}$$

$$u_i^{kt} \leq r_i^{kt} \qquad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T} \tag{44}$$

$$\sum_{i \in \mathcal{V}'} u_i^{kt} + \sum_{i \in \mathcal{V}'} v_i^{kt} \leq \beta \qquad k \in \mathcal{K}, t \in \mathcal{T} \tag{45}$$

$$\sum_{i \in \mathcal{V}'} q_i^{kt} \leq Q_k \qquad k \in \mathcal{K}, t \in \mathcal{T} \tag{46}$$

$$q_i^{kt} \geq 0 \qquad i \in \mathcal{V}', t \in \mathcal{T}, k \in \mathcal{K} \tag{47}$$

$$u_i^{kt}, v_i^{kt} \in \{0, 1\} \qquad i \in \mathcal{V}', t \in \mathcal{T}, k \in \mathcal{K}. \tag{48}$$

The objective function (40) minimizes the total inventory, removal and insertion cost. Constraints (41)−(42) are similar to (7)−(8) and enforce the ML policy. Constraints (43) ensure that if a customer is already present in a route, it cannot be reinserted in the same route. Likewise, constraints (44) guarantee that only those customers belonging to a route can be removed from it. Constraints (46) ensure that vehicle capacities are 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 (45) to limit the number of insertions and removals per route to a small value β.

3.2.1 Quantity consistency

Guaranteeing a minimum and a maximum delivery quantity to each customer is controlled by adding the following constraints to SI, which ensures that the quantities delivered lie within their specified intervals:

$$q_i^{kt} \geq (r_i^{kt} - u_i^{kt} + v_i^{kt}) g_l \sum_{t \in \mathcal{T}} d_i^t / p \qquad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T} \tag{49}$$

$$q_i^{kt} \leq (r_i^{kt} - u_i^{kt} + v_i^{kt}) g_u \sum_{t \in \mathcal{T}} d_i^t / p \qquad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T}. \tag{50}$$

3.2.2 Vehicle filling rate

To ensure a minimum vehicle filling rate in SI, the following constraints are added. They use new binary variables y^{kt} equal to 1 if and only if vehicle k is used in period t:

$$y^{kt} \geq z_i^{kt} \qquad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T} \tag{51}$$

$$\sum_{i \in \mathcal{V}'} q_i^{kt} \geq \gamma y^{kt} Q_k \qquad k \in \mathcal{K}, t \in \mathcal{T} \tag{52}$$

$$y^{kt} \in \{0,1\} \qquad k \in \mathcal{K}, t \in \mathcal{T}. \tag{53}$$

3.2.3 Order-up-to policy

The OU policy is handled through the following constraints:

$$q_i^{kt} \geq (r_i^{kt} - u_i^{kt} + v_i^{kt})C_i - I_i^{t-1} \qquad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T}. \tag{54}$$

These constraints ensure that if a delivery to a customer is performed, the quantity delivered should be at least equal to the difference between its current inventory and its inventory holding capacity. Together with constraints (41) and (42) they ensure that the quantity delivered will exactly fill the customer's inventory capacity.

3.2.4 Driver consistency

The driver consistency requirement is modeled in SI by means of an extra binary variable z_i^k equal to 1 if and only if vehicle k visits customer i, as it was defined in Section 2.2.4. Then, three sets of constraints are added to the SI model:

$$\sum_{k \in \mathcal{K}} z_i^k = 1 \qquad i \in \mathcal{V}', k \in \mathcal{K} \tag{55}$$

$$r_i^{kt} - u_i^{kt} + v_i^{kt} \leq z_i^k \qquad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T} \tag{56}$$

$$z_i^k \in \{0,1\} \qquad i \in \mathcal{V}', k \in \mathcal{K}. \tag{57}$$

Constraints (55) ensure that exactly one vehicle is assigned to each customer, while constraints (56) only allow deliveries from the vehicle assigned to that customer.

3.2.5 Driver partial consistency

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

$$\sum_{k \in \mathcal{K}} z_i^k = 1 \qquad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T} \tag{58}$$

$$r_i^{kt} - u_i^{kt} + v_i^{kt} \leq z_i^k + s_i^k \qquad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T} \tag{59}$$

$$s_i^k, z_i^k \in \{0,1\} \qquad i \in \mathcal{V}', k \in \mathcal{K}. \tag{60}$$

The penalty to the objective function is added in the same fashion as in Section 2.2.5.

3.2.6 Visit spacing

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

$$\sum_{k \in \mathcal{K}} \sum_{l=t}^{t+m_i} (r_i^{kr} - u_i^{kl} + v_i^{kr}) \leq 1 \qquad i \in \mathcal{V}', t \in \{1, ..., p - m_i\} \tag{61}$$

$$\sum_{k \in \mathcal{K}} \sum_{l=t}^{t+M_i} (r_i^{kr} - u_i^{kl} + v_i^{kr}) \geq 1 \qquad i \in \mathcal{V}', t \in \{1, ..., p - M_i\}. \tag{62}$$

3.3 Parameter settings

We now describe the parameters that govern our algorithm. We have tested different combinations for the parameters during a tuning phase. We have evaluated how the algorithm performed with different numbers of iterations. To this end, we have run it it 5,000, 10,000, 15,000, 20,000, 25,000, 30,000, 40,000 and 50,000 iterations on a small subset of instances. We then computed the average solution gap that each number of iterations provided with respect to the best solution found. Since the drop of the average gap is steep when the algorithm reaches 50,000 iterations and only equal to 0.12% we have decided to run the algorithm for 50,000 iterations without a time limit. Figure 1 depicts the performance just described.

The starting temperature τ_{start} is set to 30,000 and the cooling rate ϕ is 0.999701, which yields roughly 50,000 iterations. The stopping criterion is satisfied when the temperature reaches 0.01 or when 50,000 iterations have been performed. We have decided not to stop the algorithm after a pre-determined running time because we wanted to evaluate the impact of the different policies themselves, not an algorithmic performance. The segment length φ was set to 200 iterations and the reaction factor η was set to 0.8, that is, new weights will reflect 80% of the performance of the last segment and 20% of the last weight value. Scores are updated with $\sigma_1 = 10$, $\sigma_2 = 5$ and $\sigma_3 = 2$. A trade-off must be made between the CPU consumption and the quality of each operator of the ALNS, as well as how often SI is solved. We have evaluated this trade-off and decided to solve this subproblem with $\beta = 10$ every $\theta = 40$ ALNS iterations, which proved to be a good compromise between computing time and solution quality.

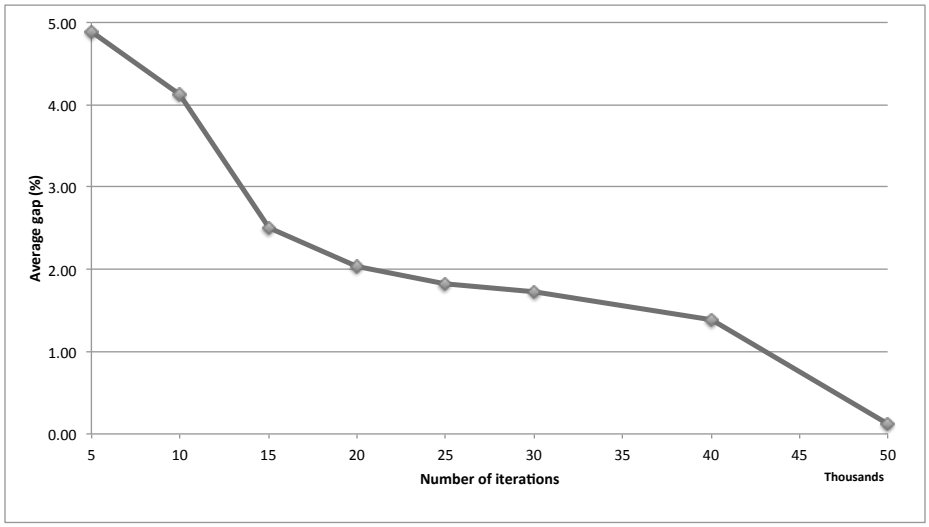

Figure 1: Average solution gap over different number of iterations.

3.4 Special rules

The algorithm can handle all six consistency features without modifications. However, its performance can be improved if some adjustments are made to better handle some features.

The first adjustment consists in applying the *avoid consecutive visits* operator only to the basic MIRP, since it could conflict with some of the consistency features proposed, thus decreasing the effectiveness of the algorithm. For example, it may pay to visit some customers on two consecutive periods if this helps achieve a better vehicle filling rate. Similarly, a later visit to a customer can be anticipated if this reduces routing costs (due to geographical proximity) or if this improves driver consistency. After some tests and considerations, we realized that whenever this operator is applied, it directs the search towards good neighborhoods, leading to better solutions. The idea is that a good solution should not visit the same customer on consecutive days, considering that it usually has sufficient inventory to meet its demand and that the number of vehicles and their capacity are limited, and their use is expensive. We have evaluated the impact of the *avoid consecutive visits* operator during the search, by running the algorithm on a subset of instances, both with and without this operator. The results of this experiment are depicted in Figure 2. It is clear from Figure 2 that the operator has a positive impact on the search process. The average percentage gap with respect to the best solution value found in this experiment is always smaller when the operator is applied. This operator is a direct result of the *visit spacing* consistency feature. We have tried different ideas from other consistency features, but none proved to be as effective for the general case.

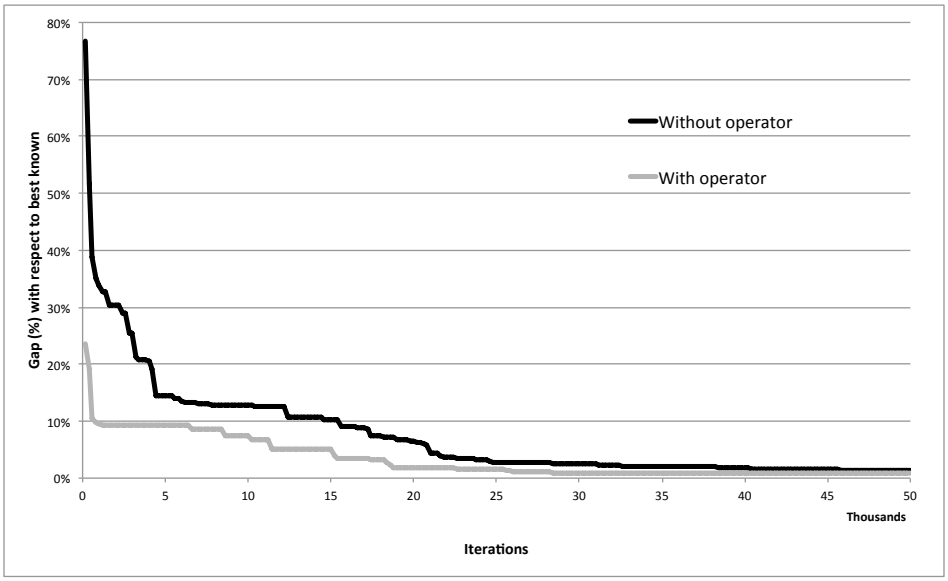

Figure 2: Impact of the *avoid consecutive visits* operator.

The second modification relates to implementation details of the different consistency features proposed. For some variants of the main problem, we have made slight modifications to the ALNS operators and to the associated network flow model in order to take into account the specifics of the variant under consideration. In order to enforce the driver consistency rule, we have modified the ALNS operators to allow insertions of customers only in vehicles that had already visited them earlier in the current solution. For the driver partial consistency rule, the only modification needed was related to the computation of the solution cost, in order to take into account the number of vehicles assigned to each customer. For the visit spacing case, the only modifications were made to the insertion operators of the ALNS, as was the case for the driver consistency feature. The OU policy was modeled directly into the remaining network flow problem as in [12], as were the minimum and maximum delivered quantity in the quantity consistency requirements. For the vehicle filling rate case, we have opted to solve SI after each ALNS iteration to help regain feasibility since in this case many ALNS operations yield infeasible solutions.

The third adjustment concerns the SI subproblem. Since it provides an approximation of the true routing costs, it is possible that after applying it to a solution, the output has a higher solution cost than the input. For this reason, we only accept the SI solution if it is better than the solution to which it was applied. In our experiments we have observed that on average 69% of the calls to SI led to improvements.

3.5 Summary of the algorithm

Algorithm 1 provides the pseudocode of our matheuristic.

Algorithm 1 Matheuristic pseudocode

Initialize weights of removal and insertion operators to 1 and scores to 0.
$s_{best} \leftarrow s \leftarrow$ *initial solution*.
$\tau \leftarrow \tau_{start}$.
while $\tau > 0.01$ and *iterations* $< 50{,}000$ **do**
    $s' \leftarrow s$.
    Select a destroy and a repair operator using the roulette-wheel and apply it to s'.
    Fix routing decisions, solve DQ to determine the delivery quantities.
    if $f(s') < f(s)$ **then**
        $s \leftarrow s'$;
        if $f(s) < f(s_{best})$ **then**
            Solve the SI model associated with s;
            $s_{best} \leftarrow s$;
            increase the score of the operators by σ_1;
        else
            increase the score of the operators by σ_2;
        end if
    else
        if s' is accepted by the simulated annealing criterion **then**
            $s \leftarrow s'$;
            increase the score of the operators by σ_3.
        end if
    end if
    if the iteration count is a multiple of φ **then**
        update the weights of all operators and reset their scores.
    end if
    if the iteration count is a multiple of θ **then**
        solve the SI model associated with s.
    end if
end while
return s_{best};

4 Computational experiments

The algorithm just described was coded in C++. We have used the scaling push-relabel algorithm for the minimum-cost flow problem developed by Goldberg [17] to solve DQ and IBM Concert Technology and CPLEX 12.2 as the solver for SI. All computations were executed on a grid of Dual Core AMD Opteron(tm) Processor 275 machines running at 2.20 GHz, each with 12 GB of RAM installed,

running a Linux operating system.

To evaluate the performance of the algorithm, we have adapted to the multi-vehicle case the 160 small single vehicle IRP instances of Archetti et al. [3, 4]. These were used in [4, 10, 12] to evaluate single vehicle algorithms for the IRP and are made up of instances with up to three time periods and 50 customers, and six time periods and 30 customers. These instances are described as small-n-low or small-n-high, where the last field refers to a low or high inventory holding cost. There are five instances for each combination and we report average statistics over these. The second set is more recent and contains 60 larger instances proposed in [4], with up to six time periods and 200 customers. They are described as large-n-low or large-n-high. There are 10 instances for each combination and we again report average values. We have adapted these instances to account for multiple vehicles by dividing the original vehicle capacity by the number of vehicles considered. We have tested our algorithm on the smaller set with two and three vehicles, and on the larger set with two to five vehicles. In total, we have solved $160 \times 2 + 60 \times 4 = 560$ instances for the basic MIRP. In the case of the consistent MIRP, we have solved instances with three vehicles. Since we have defined six versions of this problem, this means that an additional $6 \times (160 + 60) = 1{,}320$ instances were solved.

Given that there are no reported solutions for the basic MIRP, we have compared our heuristic against a truncated execution of CPLEX, both with a time limit of 3,600s. Our solutions are consistently and significantly better than those generated by CPLEX. On average the application of our heuristic reduces the gap with respect to the best known lower bound by 30%.

We provide in Tables 1 and 2 the average solution values yielded by our heuristic over the five small basic MIRP instances with two and three vehicles for $p = 3$ and $p = 6$, respectively. Table 3 contains average solution values over the 10 large basic MIRP instances with two to five vehicles, and $p = 6$.

We also report in Tables 4 to 9 the solution values of the consistent MIRP for each of the six features described in Section 2.2. The last line provides the average percentage increase of each consistent MIRP solution value with respect to the basic MIRP solution values (column $K = 3$ in Tables 1−3). Specifically, Tables 4 to 6 report statistics for each set of the low inventory cost instances, starting with three periods and five customers, and going up to six periods and 200 customers, when compared to the solution obtained by our heuristics for the general problem. Tables 7 to 9 provide statistics for the high inventory cost instances. The parameters we have used to run the tests for each type of consistency are the following:

- Quantity consistency: each delivery performed to any customer must lie within one and three times the average demand of the customer, that is $g_l = 1.0$ and $g_u = 3.0$.
- Vehicle filling rate: each dispatched vehicle must be at least 50% filled, i.e. $\gamma = 0.5$.
- Driver partial consistency: we have tested several different values for the

Table 1: Average solution values for the small basic MIRP instances, $p = 3$

Instance	Number of vehicles	
	$K = 2$	$K = 3$
small-5-low	1572.27	1963.11
small-10-low	2349.42	2850.57
small-15-low	2536.31	2911.20
small-20-low	3084.59	3563.21
small-25-low	3373.87	3865.72
small-30-low	3603.26	3985.43
small-35-low	3811.30	4292.73
small-40-low	4104.21	4451.24
small-45-low	4324.40	4681.85
small-50-low	4841.90	5391.42
small-5-high	2494.64	2879.29
small-10-high	4774.99	5276.68
small-15-high	5768.59	6143.17
small-20-high	7644.25	8130.39
small-25-high	9395.88	9849.23
small-30-high	11230.10	11590.26
small-35-high	11765.62	12251.50
small-40-high	12938.10	13374.34
small-45-high	14325.60	14692.62
small-50-high	15895.42	16488.44

Table 2: Average solution values for the small basic MIRP instances, $p = 6$

Instance	Number of vehicles	
	$K = 2$	$K = 3$
small-5-low	3926.47	4990.03
small-10-low	5793.91	7177.62
small-15-low	6433.08	7607.57
small-20-low	7875.37	9320.24
small-25-low	8605.21	10234.46
small-30-low	9054.79	10290.92
small-5-high	6147.72	7206.68
small-10-high	9803.98	11053.62
small-15-high	12601.52	13814.68
small-20-high	15934.08	17285.32
small-25-high	18194.68	19573.78
small-30-high	21706.46	22916.90

Table 3: Average solution values for the large basic MIRP instances, $p = 6$

Instance	Number of vehicles			
	$K = 2$	$K = 3$	$K = 4$	$K = 5$
large-50-low	13049.91	14249.57	18450.18	21260.23
large-100-low	25546.13	23591.50	34722.01	37561.98
large-200-low	46524.72	48225.70	63351.94	73145.96
large-50-high	32585.83	33926.45	37972.05	39836.93
large-100-high	60773.11	64562.34	72772.20	75192.23
large-200-high	121982.72	132976.90	141319.30	144866.10

penalty parameter, as reported later; for these tables, we provide results with $\alpha = 10$.

- Visit spacing: a customer may not be visited more than once in every two periods and should be visited at least once in every three periods, i.e. $m_i = 1$ and $M_i = 2$. We did not need to consider customer-dependent values since the instances were generated taking the capacity/demand ratio into account.

The *driver partial consistency* feature deserves further comments. Obviously, the choice of the value of the parameter α is highly related to the performance of the consistency feature itself and to the cost of the solutions it yields. Thus, we have also evaluated how the *driver partial consistency* case responded to different values of the penalty parameter α. Specifically we have used $\alpha = 0.1$, 1, 10 and 100. We then observed how many vehicle assignments were made in the final solution, as well as the cost of the solution. As expected, the number of extra vehicles increased in the instances with six time periods, compared with the solutions obtained for the three-period intances. This is due to the fact that many customers were served only once in the shorter horizon instances and automatically respected the driver consistency rule. Also, the number of vehicle assignments decreased to close to one per customer as the value of α increased. Figure 3 depicts the average number of vehicle assignments and solution cost per customer.

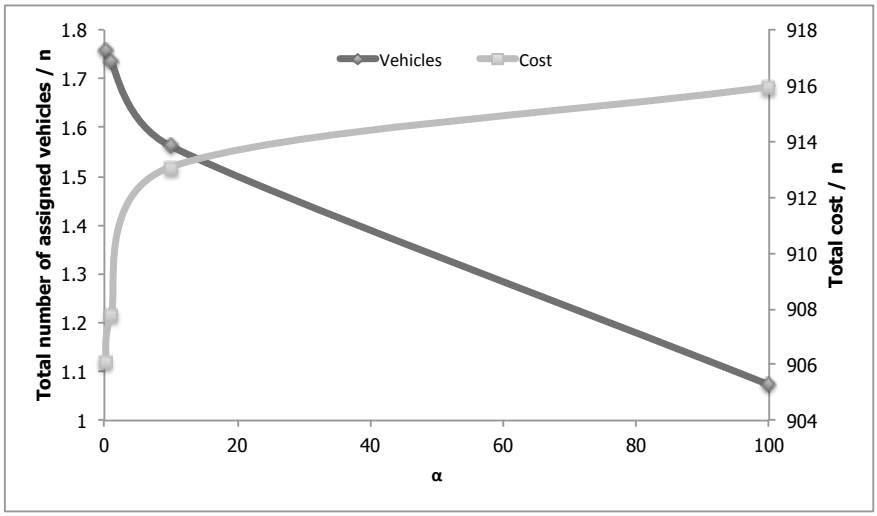

Figure 3: Average number of vehicles and cost of the solution per customer for the consistent MIRP with partial driver consistency.

We have shown that ensuring minimum and maximum intervals between successive visits to the same customer usually does not change the solution cost by more than 1.5%, but can be as high as 17% in some cases. Imposing

Table 4: Solution values for the consistent MIRP and average percentage increase with respect to the basic MIRP: $K = 3$, low inventory cost, small instances, $p = 3$

	Basic MIRP	Consistent MIRP					
		Quantity consistency	Vehicle filling rate	OU	Driver consistency	Driver partial consistency	Visit spacing
small-5-low	1963.11	2071.22	1990.31	2043.97	2027.53	1967.20	2116.30
small-10-low	2850.57	2872.31	3101.05	3062.11	2913.29	2859.45	3001.69
small-15-low	2911.81	3042.45	3073.49	3184.83	2952.26	2921.35	2986.88
small-20-low	3563.21	3625.43	4131.66	3983.33	3566.44	3552.85	3644.36
small-25-low	3865.72	3940.99	4830.65	4335.33	3877.88	3862.27	3877.81
small-30-low	3985.43	4070.08	4804.55	4339.65	3976.88	3976.12	3995.84
small-35-low	4292.73	4311.82	5377.08	4761.42	4280.70	4321.97	4279.98
small-40-low	4451.24	4556.18	5960.80	4964.44	4454.59	4506.95	4463.59
small-45-low	4681.85	4924.37	7135.81	5217.91	4652.14	4699.30	4641.69
small-50-low	5391.42	5660.46	8238.81	6094.35	5374.27	5429.03	5404.85
Average % increase		2.87	24.23	10.12	0.50	0.31	1.53

Table 5: Solution values for the consistent MIRP and average percentage increase with respect to the basic MIRP: $K = 3$, low inventory cost, small instances, $p = 6$

	Basic MIRP	Consistent MIRP					
		Quantity consistency	Vehicle filling rate	OU	Driver consistency	Driver partial consistency	Visit spacing
small-5-low	4990.03	5166.03	5048.28	5849.75	5349.59	5023.87	5094.61
small-10-low	7177.62	7294.13	7400.04	7640.90	7263.38	7139.06	7401.88
small-15-low	7607.57	8024.60	8037.87	8120.28	7774.26	7668.43	7721.73
small-20-low	9320.24	9532.51	9740.61	9692.86	9247.78	9301.70	9416.01
small-25-low	10234.46	11141.80	11171.93	10707.82	10142.26	10285.73	10477.21
small-30-low	10290.92	10830.36	11478.12	10986.84	9993.56	10316.08	10649.92
Average % increase		4.54	5.88	6.67	0.93	0.23	0.97

Table 6: Solution values for the consistent MIRP and average percentage increase with respect to the basic MIRP: $K = 3$, low inventory cost, large instances, $p = 6$

	Basic MIRP	Consistent MIRP					
		Quantity consistency	Vehicle filling rate	OU	Driver consistency	Driver partial consistency	Visit spacing
large-50-low	14249.57	18801.51	21807.19	16884.45	15382.67	14540.93	20178.25
large-100-low	23591.50	32284.46	31505.37	34519.24	29871.05	23048.24	37742.01
large-200-low	48225.70	51122.93	60967.73	57525.15	58337.89	51364.47	47838.81
Average % increase		27.38	39.44	20.70	9.85	3.54	17.48

Table 7: Solution values for the consistent MIRP and average percentage increase with respect to the basic MIRP: $K = 3$, high inventory cost, small instances, $p = 3$

	Basic MIRP	Consistent MIRP					
		Quantity consistency	Vehicle filling rate	OU	Driver consistency	Driver partial consistency	Visit spacing
small-5-high	2879.29	2989.20	2956.83	2980.34	2946.88	2884.22	3038.88
small-10-high	5276.68	5297.42	5651.45	5510.05	5339.68	5285.29	5429.97
small-15-high	6143.17	6159.05	6619.92	6400.43	6182.55	6152.00	6219.67
small-20-high	8130.39	8170.94	8554.95	8546.77	8121.97	8113.82	8200.63
small-25-high	9849.23	9917.80	10587.54	10319.73	9881.52	9847.79	9878.85
small-30-high	11590.26	11715.32	12536.84	12037.78	11600.30	11621.26	11631.32
small-35-high	12251.50	12336.04	15122.32	12747.70	12211.72	12241.54	12244.00
small-40-high	13374.34	13476.70	16953.84	13800.16	13333.00	13369.60	13335.10
small-45-high	14692.62	15060.74	16784.04	15357.78	14644.72	14731.62	14636.70
small-50-high	16488.44	16872.28	20204.08	17301.32	16482.02	16496.72	16528.04
Average % increase		1.27	12.61	4.22	0.31	0.07	0.96

Table 8: Solution values for the consistent MIRP and average percentage increase with respect to the basic MIRP: $K = 3$, high inventory cost, small instances, $p = 6$

	Basic MIRP	Consistent MIRP					
		Quantity consistency	Vehicle filling rate	OU	Driver consistency	Driver partial consistency	Visit spacing
small-5-high	7206.68	7383.89	7276.32	8059.51	7551.10	7250.36	7368.82
small-10-high	11053.62	11430.48	11336.22	11546.98	11234.52	11212.76	11399.50
small-15-high	13814.68	13977.72	13951.22	14131.80	13924.24	13761.24	13844.00
small-20-high	17285.32	17536.64	17774.02	17741.98	17274.16	17377.56	17357.76
small-25-high	19573.78	20841.74	20566.58	20312.10	19716.12	19808.78	19720.80
small-30-high	22916.90	23685.94	24109.86	23399.40	22818.68	22972.18	22946.42
Average % increase		3.00	2.89	4.34	1.18	0.57	1.06

Table 9: Solution values for the consistent MIRP and average percentage increase with respect to the basic MIRP: $K = 3$, high inventory cost, large instances, $p = 6$

	Basic MIRP	Consistent MIRP					
		Quantity consistency	Vehicle filling rate	OU	Driver consistency	Driver partial consistency	Visit spacing
large-50-high	33926.45	37486.37	41449.94	38637.65	35140.25	34470.81	38356.78
large-100-high	64562.34	67306.25	69335.96	73089.95	68054.97	63552.22	67308.71
large-200-high	132076.90	144335.10	134749.30	139010.10	134915.80	129062.10	126762.68
Average % increase		8.07	10.62	10.77	3.75	−0.67	4.51

restrictions on the quantities delivered increases the solution cost by at least 1% and by up to 27% in some sets of instances when one forces the delivered quantity to meet customer-dependent intervals, or by as much as 20% when the OU policy is enforced. Simplifying the decision process by applying the OU inventory policy increases the solution cost by more than 9% on average. This finding is consistent with the observation made in [3] for the IRP, in [12] for the IRP with transshipment, and in [5] for the integrated production-distribution problem. Imposing a high vehicle capacity utilization rate seems to be the most expensive consistency feature we have tested, especially on instances with many customers. Imposing consistency in the assignment of drivers to customers does not change the solution cost if the planning horizon is short, since many customers are served only once. Finally, allowing some of the deliveries to deviate from the driver consistency rule appears to be a very good feature, since most of the deliveries will still benefit from the driver consistency policy. Adjusting the cost parameter associated with the penalty for assigning more than one vehicle to the same customer can have a major impact both on the consistency of the assignments and on the overall cost. In our tests, the driver consistency and partial consistency policies do not increase solution cost by much.

It is also noteworthy that inventory holding costs play a major role not only in the values of the solutions obtained, but also on the performance of the algorithm. From our experiments, the gaps of the different consistency features were larger on the low inventory cost set for all but three cases. This is due to the fact that when inventory costs are low, routing decisions are relatively more important. Generating a good route is significantly harder than obtaining a good inventory replenishment policy, thus the larger gaps when inventory costs are less important.

As mentioned in Section 3.3 we have opted not to stop the algorithm after some predetermined running time because we wanted to evaluate the relative impact of each policy, and not show how the algorithm performed on any particular one. Thus, even though some computational times are large, our experiments enable us to derive insights on how much each policy would cost to the decision maker, and once he makes his decision, a specific algorithm can be applied to obtain a solution for that particular policy in less time. Specifically, the driver consistency rule yields a high average running time, due to the constraint added to the SI subproblem, with 140,000 seconds on average for the large instances. One particular instance of the driver partial consistency rule ran for almost 30,000 seconds. Simpler models, such as the basic MIRP or the OU policy had an average running time of 2,000 seconds for the small instances with three periods and of 8,000 seconds for the small instances with six periods. On the larger instances, both policies yielded an average of 14,000 seconds.

5 Conclusions

We have incorporated six consistency features in the MIRP. One of these is the well-known OU replenishment policy, and another is the concept of driver consistency already introduced in the context of the multi-period VRP. We have developed a matheuristic composed of an ALNS enhanced by the exact solution of two types of MILPs. The first one is a network flow model used to compute delivery quantities associated with a given set of routes. The second one provides an approximation of the cost of a new solution obtained by applying vertex removals and reinsertions to a given solution. The algorithm is sufficiently flexible to handle the basic MIRP as well as any combination of the six consistency features we have considered. However, the performance improves when some adjustments are made for certain features. Extensive computational tests on benchmark instances have shown that introducing some of these features can increase the average solution cost significantly, by up to 40% when imposing a high vehicle capacity utilization, or can cost as little as less than 1% when controlling the interval between successive visits to the same customer. Our study clearly illustrates the costs and benefits of incorporating consistency features in the basic MIRP.

References

[1] T. F. Abdelmaguid and M. M. Dessouky. A genetic algorithm approach to the integrated inventory-distribution problem. *International Journal of Production Research*, 44:4445–4464, 2006.

[2] H. Andersson, A. Hoff, M. Christiansen, G. Hasle, and A. Løkketangen. Industrial aspects and literature survey: Combined inventory management and routing. *Computers & Operations Research*, 37(9):1515–1536, 2010.

[3] C. Archetti, L. Bertazzi, G. Laporte, and M. G. Speranza. A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation Science*, 41(3):382–391, 2007.

[4] C. Archetti, L. Bertazzi, A. Hertz, and M. G. Speranza. A hybrid heuristic for an inventory routing problem. *INFORMS Journal on Computing*, Forthcoming, 2011.

[5] C. Archetti, L. Bertazzi, G Paletta, and M. G. Speranza. Analysis of the maximum level policy in a production-distribution system. *Computers & Operations Research*, 12(38):1731–1746, 2011.

[6] C. A. Barlett and S. Ghoshal. Building competitive advantage through people. *MIT Sloan Management Review*, 43(2):34–41, 2002.

[7] M. Barrat. Positioning the role of collaborative planning in grocery supply chains. *International Journal of Logistics Management*, 14(2):53–66, 2003.

[8] B. M. Beamon. Measuring supply chain performance. *International Journal of Operations & Production Management*, 19(3):275–292, 1999.

[9] W. J. Bell, L. M. Dalberto, M. L. Fisher, A. J. Greenfield, R. Jaikumar, P. Kedia, R. G. Mack, and P. J. Prutzman. Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. *Interfaces*, 13(6):4–23, 1983.

[10] L. Bertazzi, G. Paletta, and M. G. Speranza. Deterministic order-up-to level policies in an inventory routing problem. *Transportation Science*, 36 (1):119–132, 2002.

[11] N. Christofides and J. E. Beasley. The periodic routing problem. *Networks*, 14(2):237–256, 1984.

[12] L. C. Coelho, J.-F. Cordeau, and G. Laporte. The inventory-routing problem with transshipment. Technical Report 2011-21, CIRRELT, Montréal, Canada, 2011.

[13] J.-F. Cordeau, G. Laporte, M. W. P. Savelsbergh, and D. Vigo. Vehicle routing. In C. Barnhart and G. Laporte, editors, *Transportation*, pages 367–428. North-Holland, Amsterdan, 2007.

[14] S. Dauzère-Pérès, A. Nordli, A. Olstad, K. Haugen, P. O. Koester, U. Myrstad, T. Geir, and R. Alf. Omya Hustadmarmor optimizes its supply chain for delivering calcium carbonate slurry to European paper manufacturers. *Interfaces*, 37(1):39–51, 2007.

[15] M. Desrochers and G. Laporte. Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints. *Operations Research Letters*, 10(1):27–36, 1991.

[16] P. Francis, K. Smilowitz, and M. Tzur. The periodic vehicle routing problem and its extensions. In B. L. Golden, S. Raghavan, and E. A. Wasil, editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, pages 239–261. Springer, New York, 2008.

[17] A. V. Goldberg. An efficient implementation of a scaling minimum-cost flow algorithm. *Journal of Algorithms*, 22(1):1 – 29, 1997.

[18] C. Groër, B. L. Golden, and E. A. Wasil. The consistent vehicle routing problem. *Manufacturing & Service Operations Management*, 11(4):630–643, 2009.

[19] I. Kara, G. Laporte, and T. Bektas. A note on the lifted Miller-Tucker-Zemlin subtour elimination constraints for the capacitated vehicle routing problem. *European Journal of Operational Research*, 158(3):793 – 795, 2004.

[20] H. L. Lee and W. Seungjin. The whose, where and how of inventory control design. *Supply Chain Management Review*, 12(8):22 – 29, 2008.

[21] V. Maniezzo, T. Stützle, and S. Voß. *Matheuristics: Hybridizing Metaheuristics and Mathematical Programming*. Springer, New York, 2009.

[22] D. L. Olson and M. Xie. A comparison of coordinated supply chain inventory management systems. *International Journal of Services and Operations Management*, 6(1):73–88, 2010.

[23] S. Ropke and D. Pisinger. An adaptive large neighborghood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4):455–472, 2006.

[24] S. Ropke and D. Pisinger. A unified heuristic for a large class of vehicle routing problems with backhauls. *European Journal of Operational Research*, 171(3):750–755, 2006.

[25] P. Shaw. A new local search algorithm providing high quality solutions to vehicle routing problems. Technical report, University of Strathclyde, Glasgow, 1997.

[26] K. Smilowitz, M. Nowak, and T. Jiang. Workforce management in periodic delivery operations. *Transportation Science*, Forthcoming.

[27] Y. Yugang, C. Haoxun, and C. Feng. A new model and hybrid approach for large scale inventory routing problems. *European Journal of Operational Research*, 189(3):1022–1040, 2008.