Submitted to *INFORMS Journal on Computing*
manuscript JOC-2017-01-OA-007

A Unified Decomposition Matheuristic for Assembly, Production and Inventory Routing

Masoud Chitsaz
HEC Montréal and GERAD, 3000 Chemin de la Côte-Sainte-Catherine, Montréal, H3T 2A7 Canada
masoud.chitsaz@hec.ca

Jean-François Cordeau
HEC Montréal and GERAD, 3000 Chemin de la Côte-Sainte-Catherine, Montréal, H3T 2A7 Canada
jean-francois.cordeau@hec.ca

Raf Jans
HEC Montréal and GERAD, 3000 Chemin de la Côte-Sainte-Catherine, Montréal, H3T 2A7 Canada
raf.jans@hec.ca

While the joint optimization of production and outbound distribution decisions in a manufacturing context has been intensively studied in the past decade, the integration of production, inventory and inbound transportation from suppliers has received much less attention despite its practical relevance. This paper aims to fill the gap by introducing a general model for the assembly routing problem (ARP), which consists of simultaneously planning the assembly of a finished product at a plant and the routing of vehicles collecting materials from suppliers to meet the inventory requirements imposed by the production. We formulate the problem as a mixed-integer linear program and we propose a three-phase decomposition matheuristic that relies on the iterative solution of different subproblems. The first phase determines a setup schedule while the second phase optimizes production quantities, supplier visit schedules and shipment quantities. The third phase solves a vehicle routing problem for each period in the planning horizon. The algorithm is flexible and we show how it can also be used to solve two well-known problems related to the ARP: the production routing problem (PRP) and the inventory routing problem (IRP). Using the same parameter setting for all problems and instances, we obtain 781 new best known solutions out of 2,628 standard IRP and PRP test instances. In particular, on large-scale multi-vehicle instances, the new algorithm outperforms specialized state-of-the-art heuristics for these two problems.

Key words: production, inventory, routing, assembly decomposition matheuristic, three-phase iterative heuristic
History:

1. Introduction

The literature on production planning has paid a lot of attention in the past decade to the integration of lot sizing and outbound transportation decisions. The typical supply chain that is considered consists of a plant that delivers final products to several customers. Considering both the production planning at the plant and the outbound delivery to the customers via routes results in what is called the production routing problem (PRP) (Adulyasak et al. 2015). If the production quantities at the plant are assumed to be given and the decisions only relate to the inventory and route planning, the problem is referred to as the inventory routing problem (IRP) (Andersson et al. 2010, Bertazzi et al. 2008, Coelho et al. 2013).

In contrast, only few studies have focused on the integration of production planning with inbound transportation planning. Yet, in a standard supply chain, a plant often uses several different components to assemble a final product. These components are typically produced in other plants or purchased from suppliers. If the assembly plant is responsible for organizing the inbound transportation of the various components, then gains can be achieved by integrating the production planning with the inbound vehicle routing. We refer to this problem as the assembly routing problem (ARP).

The aim of this paper is to introduce a general model for the ARP. We provide a mathematical formulation of the problem which serves as the basis for a decomposition matheuristic that iteratively solves different subproblems. We also explain how the same methodology can solve the related IRP and PRP. Using the same parameter setting for all three problems, this algorithm outperforms existing heuristics on large-scale multi-vehicle instances of the IRP and PRP, obtaining new best known solutions to many standard test instances.

The ARP has many industrial applications in situations where the production plant and several suppliers are owned by the same company, or when the manufacturer is the biggest player in the supply chain and centrally coordinates the inbound logistics decisions. This is a relevant practical problem in several areas. Fleischmann and Meyr (2003) indicate that in the automotive industry the organization that receives the components is usually responsible for the supply transport. Florian et al. (2011) show that in addition to the direct financial benefits for the supply chain, inbound logistics integration for a German car manufacturer has some further important outcomes such as a reduction in CO_2 emissions.

In an application for the Delco Electronics Division of General Motors (GM), Blumenfeld et al. (1987) find that the overall optimization of the inbound transportation resulted in a 26% (2.9 million dollars per year, in 1987's USD) savings potential. They propose the use of an approximation method for the routing cost estimation in their studies to reduce the complexity of the problem. Implementing their solution package, GM of Canada reports savings of approximately 157 thousand USD in four months. Danese (2006) presents the case of GlaxoSmithKline (GSK), an international pharmaceutical group that extended the vendor managed inventory (VMI) approach to its suppliers as a response to the highly competitive and regulated market to benefit from the integrated and coordinated planning process.

Other cases where the buyer is responsible for the transportation are incorporated in several Incoterms, which are often used to clearly define the contractual responsibilities of the buyer and seller in international commercial transactions. Several of these terms consider the cases where the buyer is responsible for the transportation costs and risks. The Incoterm EXW (Ex Works) indicates a situation in which the seller makes the goods available, typically at the factory or a warehouse, and the buyer is responsible for the further transportation. In maritime transport, the Incoterms like FOB (Free On Board) for sea transport or inland waterway transport and FCA (Free Carrier) for roll-on/roll-off or container traffic, address the situation where the seller is responsible for the costs and risks up to when the goods are delivered to the ship at the named port of shipment. Then, it is the buyer who is responsible for the costs and risks from that point onwards.

In the retail sector, the concept of factory gate pricing (FGP) has emerged (Whiteoak 1994, Le Blanc et al. 2006, Fernie and Sparks 2014). Under FGP, the supplier no longer delivers the products to the customer but makes them available at its own factory gate (Le Blanc et al. 2006). This requires the customer to plan and synchronize the pickups from the suppliers to reduce the transportation costs as reported by a number of FGP studies. Examples are Le Blanc et al. (2006) for a large Dutch retail distribution company and Potter et al. (2007) for UK retailers. Potter et al. (2007), Whiteoak (1994) and Fernie and Sparks (2014) report success in increasing the product flow while at the same time reducing distance for Tesco, ASDA and Sainsbury's retailers.

To the best of our knowledge, the problem of jointly optimizing production planning and inbound vehicle routing with a finite horizon and discrete planning periods has only been considered by Hein and Almeder (2016). They study the case of multiple components and products, and consider two scenarios. In the first scenario, components can be kept at the plant, whereas the second scenario considers a JIT environment assuming that the components that arrive at the plant must be used immediately in production. Furthermore, the holding cost at the suppliers is not considered in their specific study. Consequently, the combined decision making is entirely centered on the plant costs without taking the suppliers' cost into account.

Motivated by the above-mentioned applications and to fill the gap in the literature, we study for the first time the problem of the integrated inbound transportation, production and inventory planning in a finite planning horizon with the standard basic assumptions similar to the IRP and PRP. This is the first contribution of this paper. Second, we present a unified decomposition matheuristic capable of solving not only the ARP, but also the IRP and PRP. Third, we present a procedure to compute an upper bound on the number of routes in an optimal solution to the capacitated vehicle routing problem, and show that applying this bound significantly reduces the solution time in our heuristic. Fourth, we propose several cost update mechanisms to approximate the routing cost, and as our sensitivity analysis indicates, using a mix of two update mechanisms improves the quality of the solutions. Fifth, we report the results of extensive computational experiments on more than four thousand instances for these three problems, including standard data sets for the IRP and PRP. The results indicate that our algorithm outperforms the state-of-the-art heuristics on the large-scale multi-vehicle IRP and PRP instances. Finally, further analyses demonstrate the robust behavior of the algorithm.

The remainder of the paper is organized as follows. We provide a short literature review on the integration of production planning with outbound and inbound transportation in Section 2 in order to better position our problem with respect to the existing literature. Then, we define the ARP and express it mathematically in Section 3. We describe the decomposition matheuristic in Section 4. We present the algorithm implementation, the benchmark algorithms and the results of extensive computational experiments on all data sets in Section 5. Finally, Section 6 concludes the paper.

2. Literature Review

The majority of the research on the integrated production planning and outbound routing problem, which is most commonly referred to as the PRP, considers a finite time horizon with discrete planning periods. The associated models are typically formulated as mixed integer linear programs. Chandra (1993) was the first to address this problem by assuming a fixed cost for the warehouse orders, which in terms of modeling is similar to the production setup cost; Chandra (1993) studies a problem with an uncapacitated order size and an unlimited number of capacitated vehicles. Later, Chandra and Fisher (1994) define the same multi-commodity version of the problem in a more formal way, this time by considering the production setup costs. Several studies on this problem (Boudia et al. 2007, Boudia and Prins 2009, Bard and Nananukul 2009, 2010, Adulyasak et al. 2014a,b, Absi et al. 2014) consider one capacitated production plant producing a single product for multiple customers with inventory costs and inventory capacities both at the plant and customers. The plant is responsible for fulfilling the deterministic demand of the customers during the planning periods. The production setup cost is considered to be constant over the periods. A limited number of homogeneous and capacitated vehicles is also considered to perform the shipments from the plant to the customers. The multi-commodity version of the problem was studied by Fumero and Vercellis (1999) and Armentano et al. (2011). Lei et al. (2006) is the only study that considers multiple production plants producing one single final product and they assume a heterogeneous fleet of vehicles. The studies of Solyalı et al. (2009) and of Archetti et al. (2011) do not assume a capacity for the production. The state-of-the-art heuristic algorithms for the PRP are the adaptive large neighborhood search (ALNS) of Adulyasak et al. (2014b) and the matheuristic of Absi et al. (2014). For the IRP, the heuristic of Archetti et al. (2012) is the best performing algorithm for single-vehicle instances and the matheuristic of Archetti et al. (2017) is the best algorithm for multi-vehicle instances.

There are some studies that consider the optimization of the inbound transportation and inventory decisions without considering the production planning at the central plant. Inspired by the automotive parts supply chain, Lee et al. (2003), Moin et al. (2011) and Mjirda et al. (2014) study a multi-period, multi-supplier problem with a single assembly plant in which each supplier provides a distinct part type. Popken (1994) and Berman and Wang (2006) study a single period (static) multicommodity inbound logistics problem with three sets of nodes: origin nodes or suppliers, a destination node, and transshipment terminal nodes. In their model, the origin-destination commodity flows pass through the paths of the network using at most one terminal node, but the vehicle routes are not considered explicitly.

Some studies address inbound vehicle routing in JIT/lean production systems to coordinate the material inflow with the production rate. Vaidyanathan et al. (1999) and, later, Patel and Patel (2013) and Satoglu and Sahin (2013) investigate the delivery of parts in a central warehouse to the stations of an assembly line on a JIT basis. The quantity delivered per trip should meet the demand for the duration of the trip. As a result, vehicles will have no idle time between trips and inventories at the demand points are minimized. Qu et al. (1999) and Sindhuchao et al. (2005) consider the joint replenishment of multiple items in an inbound material-collection system for a central warehouse under the assumption of an infinite planning horizon. They do not take into account the vehicle capacity and storage space limit. Chuah and Yingling (2005) consider these two assumptions and study a JIT supply pickup problem for an automotive assembly plant with a restricted set of possible discrete frequencies. They also assume time windows at the suppliers. Stacey et al. (2007) and Natarajarathinam et al. (2012) offer new heuristics for the same problem. Ohlmann et al. (2007) expand the work of Chuah and Yingling (2005) by assuming general visit frequencies. They allow suppliers on the same route to have different pickup frequencies so that not every supplier is visited every time. Jiang et al. (2010) study a JIT parts supply problem in the automobile industry to minimize the inventory and transportation costs under storage space limit and common frequency routing assumptions. Yücel et al. (2013) consider a bilevel optimization problem for transporting specimens from a number of geographically dispersed sites to the processing facility of a clinical testing company. At the first level they maximize the daily processed amount while at the second level they minimize the daily transportation cost. Dong and Turnquist (2015) investigate a similar problem to design the inbound material collection routes. They consider pick-up frequency and spatial design as joint decisions to minimize total inventory and transportation costs with a single-level objective function. Lamsal et al. (2016) study a deterministic sugarcane harvest logistics problem in Brazil. The decisions to make are the harvest rate at the geographically dispersed fields and the truck assignment schedule to pick up the loads to minimize the time between the cutting of the sugar cane in the field and the crushing at the mill. They consider the constraint that the mill should never run out of raw material. Francis et al. (2006) study a variation of the periodic vehicle routing problem (PVRP) in which service frequency is a decision of the model. This brings more flexibility for the system's operator.

The problem of integrating inbound transportation with the production and inventory decisions is also gaining attention. Almost all of the research on this problem, with the exception of the previously mentioned study by Hein and Almeder (2016), considers an infinite planning horizon in a continuous time framework and uses mixed integer nonlinear programming models. This problem is referred to in the literature as the economic lot and supply scheduling problem (ELSSP) and was introduced by Liske and Kuhn (2009). Extending the economic order quantity (EOQ) assumptions, the ELSSP aims at finding synchronized cyclic production and routing patterns. Other studies on this problem include Kuhn and Liske (2011), Kuhn and Liske (2014), Bae et al. (2014), and Chen and Sarker (2014).

3. Problem Definition and Formulation

We consider a many-to-one assembly system where n suppliers, represented by the set $N_s = \{1, ..., n\}$, each provide a unique component necessary for the production of a final product at the central plant, denoted by node 0. The planning horizon comprises a finite number of discretized time periods, represented by the set $T = \{1, .., l\}$. The component supply, s_{it}, at each supplier $i \in N_s$ in each period $t \in T$ is predetermined over the planning horizon. The production system has to satisfy the external demand, d_t, for the final product at the plant in each period $t \in T$ without stockouts while respecting the plant's production capacity, which is given by C. Both the suppliers and the plant can hold inventory. Each supplier $i \in N_s$ has a storage capacity L_i for its components. The plant provides a shared storage with capacity L for the components and has a separate outbound storage capacity K for the final product. A fleet of m homogeneous vehicles, each with a capacity of Q, is available to perform shipments from the suppliers to the plant using routes that start and end at the plant. We suppose throughout that the components delivered to the plant in period $t \in T$ can be used for production in the same period.

We assume that one unit of each component is needed to make one unit of the final product. Note that in basic assembly structures, it is possible to define the units of measurement of the components so as to satisfy this assumption without loss of generality (see Pochet and Wolsey 2006, chap. 13). Obviously, the unit components may not have identical sizes. Therefore, we consider that each component has a unit size of b_i. This size will be taken into account in the vehicle capacity and plant storage area for components. We consider a unit production cost u and setup cost f at the plant level. The unit holding costs of h_i and r_i are imposed for the inventory of component i at its supplier and at the plant, respectively. The inventory of the final product incurs a unit holding cost of r_0 at the plant. When a vehicle travels from location i to j it entails a period-independent cost of c_{ij}.

In the ARP, the following decisions should be optimized simultaneously for each period:
1. whether or not to produce the final product at the plant and the quantity to be produced;
2. the quantity to be shipped from the suppliers to the plant, and;
3. which suppliers to visit, in what order and by which vehicle.

To model the ARP we define a complete undirected graph $G = (N, E)$, and assume that the triangular inequality holds. Let $N = N_s \cup \{0\}$ be the set of nodes, and $E = \{(i, j) : i, j \in N, i < j\}$ be the set of edges. Since we assume a one-to-one relationship between suppliers and components, N_s also represents the set of components and $i = 0$ the final product. For each period $t \in T$, we let the binary variable y_t take value 1 if and only if production takes place at the plant and we let p_t denote the production quantity. Let I_{it} represent the inventory of component i at supplier $i \in N_s$ at the end of period t. Define F_{it} as the inventory of component $i \in N_s$ or of the final product $i = 0$ at the plant at the end of period t. Let q_{it} indicate the shipment quantity from supplier i to the plant in period t. The variable x_{ijt} represents the number of times a vehicle traverses the edge $(i, j) \in E$ in period $t \in T$. Since we define the model on an undirected network, x_{ijt} is a binary variable for $i > 0$ and may take values in $\{0, 1, 2\}$ for $i = 0$. The binary supplier visit variable z_{it} takes value 1 if and only if a supplier $i \in N_s$ is visited in period t, and the integer variable z_{0t} indicates the number of vehicles dispatched from the plant in period t. Table 1 presents a summary of the notation.

Using this notation, the ARP can be formulated as the following mixed integer program $(\mathcal{M}_{ARP})$.

$$(\mathcal{M}_{ARP}) \quad \min \sum_{t \in T} \Big(up_t + fy_t + \sum_{i \in N_s} h_i I_{it} + \sum_{i \in N} r_i F_{it} + \sum_{(i,j) \in E} c_{ij} x_{ijt} \Big) \tag{1}$$

Table 1 ARP notation

Sets:	
N	Set of nodes, indexed by $i \in \{0, ..., n\}$, where 0 represents the plant and $N_s = N \setminus \{0\}$ is the set of suppliers. Note that since there is a one-to-one relationship between nodes and items, N also represents the set of components and the final product.
E	Set of edges, $E = \{(i,j) : i,j \in N, i < j\}$.
T	Set of time periods, indexed by $t \in \{1, .., l\}$.
$E(S)$	Set of edges $(i,j) \in E$ such that $i,j \in S$, where $S \subseteq N$ is a given set of nodes.
$\delta(S)$	Set of edges incident to a node set S, $\delta(S) = \{(i,j) \in E : i \in S, j \notin S$ or $i \notin S, j \in S\}$.
Parameters:	
f, u	Fixed setup and unit production costs, respectively.
h_i	Unit holding cost at node $i \in N_s$.
r_i	Unit holding cost of component/final product $i \in N$ at the plant.
c_{ij}	Transportation cost between nodes i and j, $(i,j) \in E$.
C, Q	Production and vehicle capacity, respectively.
m	Fleet size.
s_{it}	Component supply at node $i \in N_s$ in period t.
b_i	Unit size of component $i \in N_s$.
d_t	Demand for the final product in period t.
L_i	Inventory capacity for the components at node $i \in N$.
L	Shared inventory capacity for the components at the plant.
K	Inventory capacity for the final product (at the plant).
I_{i0}	Initial inventory available at node $i \in N_s$.
F_{i0}	Initial inventory of component/final product $i \in N$ at the plant.
Decision variables:	
p_t	Production quantity in period t at the plant.
y_t	Equals to 1 if there is a setup at the plant in period t, 0, otherwise.
I_{it}	Inventory of component i at node $i \in N_s$ at the end of period t.
F_{it}	Inventory of component/final product $i \in N$ at the plant at the end of period t.
x_{ijt}	Number of times a vehicle traverses the edge $(i,j) \in E$ in period t.
z_{it}	Equals to 1 if node $i \in N_s$ is visited in period t, 0, otherwise.
z_{0t}	Number of vehicles dispatched from the plant in period t.
q_{it}	Quantity shipped from node $i \in N_s$ to the plant in period t.

s.t.

$$F_{i,t-1} + q_{it} = p_t + F_{it} \quad \forall i \in N_s, \forall t \in T \tag{2}$$

$$F_{0,t-1} + p_t = d_t + F_{0t} \quad \forall t \in T \tag{3}$$

$$I_{i,t-1} + s_{it} = q_{it} + I_{it} \quad \forall i \in N_s, \forall t \in T \tag{4}$$

$$p_t \leq C y_t \quad \forall t \in T \tag{5}$$

$$\sum_{i \in N_s} b_i F_{it} \leq L \quad \forall t \in T \tag{6}$$

$$F_{0t} \leq K \quad \forall t \in T \tag{7}$$

$$I_{it} \leq L_i \quad \forall i \in N_s, \forall t \in T \tag{8}$$

$$z_{0t} \leq m \quad \forall t \in T \tag{9}$$

$$b_i q_{it} \leq Q z_{it} \quad \forall i \in N_s, \forall t \in T \tag{10}$$

$$\sum_{(j,j') \in \delta(i)} x_{jj't} = 2 z_{it} \quad \forall i \in N, \forall t \in T \tag{11}$$

$$Q \sum_{(i,j) \in E(S)} x_{ijt} \leq \sum_{i \in S} (Q z_{it} - b_i q_{it}) \quad \forall S \subseteq N_s, |S| \geq 2, \forall t \in T \tag{12}$$

$$z_{0t} \in \mathbb{Z} \quad \forall t \in T \tag{13}$$

$$F_{0t}, p_t \geq 0, y_t \in \{0,1\} \quad \forall t \in T \tag{14}$$

$$I_{it}, F_{it}, q_{it} \geq 0, z_{it} \in \{0,1\} \quad \forall i \in N_s, \forall t \in T \tag{15}$$

$$x_{ijt} \in \{0,1\} \quad \forall (i,j) \in E : i \neq 0, \forall t \in T \tag{16}$$

$$x_{0it} \in \{0,1,2\} \quad \forall i \in N_s, \forall t \in T. \tag{17}$$

The objective function (1) minimizes the total production, setup and holding costs in addition to the transportation costs. The holding cost includes component inventory at the suppliers and plant as well as the final product inventory at the plant. The inventory flow balance for the components and the final product at the plant is imposed through constraints (2) and (3). Constraints (4) ensure the inventory flow balance at each supplier. Constraints (5) force a setup at the plant for each period in which production takes place. They also impose the production capacity. Constraints (6) and (7) represent the storage capacity for the components and final product at the plant. The storage capacity for the components at each supplier is imposed by constraints (8). Constraints (9) limit the fleet size. Constraints (10) force a vehicle visit whenever components are shipped from a supplier to the plant. The maximum component shipment quantity from each supplier in each period is also limited by the vehicle capacity. Constraints (11) are the degree constraints. Constraints (12) are the subtour elimination constraints (SECs) and they also impose the vehicle capacity. These constraints are the modified version of the VRP capacity-cut constraints (Toth and Vigo 2002, Lysgaard et al. 2004, Iori et al. 2007), and are referred to as generalized fractional subtour elimination constraints (GFSEC) (Adulyasak et al. 2014a) in the context of the PRP.

It is easy to show that the ARP is NP-hard since the VRP is a special case of it. Note that the ARP and PRP are not special cases of each other. Moreover, the ARP and PRP are not mirror problems and one cannot simply exchange customers and suppliers. In the ARP, we consider two separate storage areas at the plant for the components (inbound storage) and the final product (outbound storage), respectively. This results in inventory balance constraints for both the components and the final product at the plant. In the ARP, one unit of each component is required for producing one unit of the final product. In contrast, in the PRP, only the final product is represented. Another difference is that in the ARP it may be necessary to visit a supplier to avoid exceeding the maximum storage capacity (overflow). However, in the PRP, one prevents the stockout at the customers/retailers. For the same reasons, although the IRP is a special case of the PRP (where the production rates are predetermined and given), it is not a special case of the ARP.

4. A Decomposition Matheuristic

In this section we present a unified decomposition matheuristic for the ARP, which can also be applied to the PRP and the IRP. We explain the algorithm in the context of the ARP and its adaptation for the other two problems is explained in Section 5.2 and in Appendix C.

Our algorithm decomposes the $\mathcal{M}_{ARP}$ model into three separate subproblems. The first subproblem, $\mathcal{M}_y$, is a special lot-sizing problem that determines a setup schedule by using the number of dispatched vehicles to calculate an approximation of the routing costs (Section 4.1). Considering a given setup schedule, the second subproblem, $\mathcal{M}_z$, uses a transportation cost approximation (σ_{it}) associated with each visit to supplier i, and chooses the node visits and shipment quantities (Section 4.2). For multi-vehicle instances, a modified model ($\mathcal{M}_z^{\mathcal{R}}$) is employed in this phase to look for possible improvements in node visits and shipments. Finally, the third subproblem solves a series of separate vehicle routing problems (Section 4.3), one for each period t (VRP_t). The solutions of the routing subproblems are then used to update the transportation cost approximation (σ_{it}) in the $\mathcal{M}_z$ model (Section 4.4). This procedure is repeated for a number of iterations to reach a local optimum. Then, a local branching scheme is used to change the setup schedule and explore other parts of the feasible solution space, looking for better solutions (Section 4.5). The entire procedure continues until a stopping condition is met (Section 4.6).

Our algorithm shares similarities with the decomposition-based heuristic developed by Absi et al. (2014) for the PRP. However, there are also important differences between the two algorithms. The method of Absi et al. (2014) uses a two-phase approach where in the first phase it fixes y_t, p_t and q_{it} decisions. In our algorithm this is done in two separate phases: it fixes the y_t decisions at the end of the first phase, then finds p_t and q_{it} in the second phase. Our method also prevents the same solution to appear twice by adding diversification constraints (Section 4.5) to cut the current node visit pattern in the next iteration and to cut the current setup schedule in order to diversify the search. We also implement two transportation cost approximation mechanisms. Finally, for the diversification, Absi et al. (2014) employ a random transportation cost perturbation mechanism while we change the setup schedule. An overview of our three-phase decomposition heuristic is presented in Algorithm 1.

Algorithm 1: CCJ-DH

Initialize σ_{it}
repeat
    if first iteration **or** diversification step **then**
        if diversification step **then**
            Cut the current setup schedule from $\mathcal{M}_y$ model
            Reset aggregate fleet capacity for all periods
        end if
        Solve $\mathcal{M}_y \rightarrow y_t$ (and p_t, z_{it}, q_{it})
        Fix y_t decisions
    else
        Solve $\mathcal{M}_z$ with fixed $y_t \rightarrow p_t, z_{it}, q_{it}$
    end if
    Solve VRP_t subproblems with fixed $z_{it}, q_{it} \rightarrow x_{ijt}$
    Select transportation cost update mechanism $\rightarrow \sigma_{it}$
    if all VRP_t solutions are feasible **then**
        Update incumbent solution
        if (effective aggregate fleet capacity is reduced in some periods **and**
            after a minimum number of iterations **and**
            for a minimum quality of the current solution) **then**
            repeat
                Solve $\mathcal{M}_z^{\mathcal{R}}$ with fixed $z_{it} \rightarrow p_t, z_{it}, q_{it}$
                Solve VRP_t subproblems with fixed $z_{it}, q_{it} \rightarrow x_{ijt} \rightarrow \sigma_{it}$
                Update incumbent solution
            until the stopping condition is met
        end if
    else
        Decrease effective aggregate fleet capacity for the periods with infeasible VRP_t
    end if
    Cut the current node visit pattern from $\mathcal{M}_z$ model
until the stopping condition is met
return incumbent solution

4.1. Phase 1: The $\mathcal{M}_y$ Subproblem

The $\mathcal{M}_y$ subproblem aims to generate a good setup schedule by solving a simplified problem in which we use an approximate transportation cost based on the number of vehicles dispatched from the plant. To this end, we update the original objective function (1) with the following:

$$\min \sum_{t\in T}\Big(up_t + fy_t + \sum_{i\in N_s} h_i I_{it} + \sum_{i\in N} r_i F_{it} + \sigma_{0t} z_{0t}\Big). \tag{18}$$

We consider a cost (σ_{0t}) for each dispatched vehicle in each period (Section 4.4). With this modification, constraints (11)-(12) become redundant and they are replaced with the following constraints which impose an aggregate fleet capacity:

$$\sum_{i\in N_s} b_i q_{it} \le Q z_{0t} \quad \forall t \in T. \tag{19}$$

We define the $\mathcal{M}_y$ model with the objective function (18) subject to constraints (2)-(10), (13)-(15) and (19). This model yields a setup schedule in the first iteration and whenever a diversification step is performed. Adding a diversification constraint LBI_y (Section 4.5) prevents the same setup schedule to appear when we solve the model again. As a by-product, the solution to this model specifies the shipment quantity variables, q_{it}. Based on these shipment quantities, we can deduce the corresponding node visit variables. Therefore, whenever solving the $\mathcal{M}_y$ model, we can skip phase 2 and immediately go to phase 3 in which the VRP_t subproblems (line 13 of Algorithm 1) are solved.

4.2. Phase 2: $\mathcal{M}_z$ and $\mathcal{M}_z^{\mathcal{R}}$ Subproblems

In the second phase, the focus is on obtaining proper node visit decisions and shipment quantities. Using the solution found in the first phase, the binary decisions y_t are fixed in constraints (5) of the $\mathcal{M}_{ARP}$ model. We approximate the transportation cost in the objective function using the node visit variables z_{it}, which results in the following objective function:

$$\min \sum_{t\in T}\Big(up_t + \sum_{i\in N_s} h_i I_{it} + \sum_{i\in N} r_i F_{it} + \sum_{i\in N_s} \sigma_{it} z_{it}\Big). \tag{20}$$

We assume a cost (σ_{it}) for each node visit in each period (Section 4.4). With the removal of variables x_{ijt} and z_{0t} as well as constraints (9) and (11)-(13), it is no longer possible to enforce the vehicle capacity. However, by adding constraint $\sum_{i\in N_s} b_i q_{it} \le mQ$ for every period t we can preserve the aggregate fleet capacity. Since split pickups are not allowed, we may not be able to find a feasible VRP solution for a certain period in phase 3 because the different quantities (q_{it}) to be shipped cannot be packed in the available vehicles.

Therefore, as in Absi et al. (2014), we use the following constraints to impose a smaller aggregate fleet capacity ($0 \leq \lambda_t \leq 1$):

$$\sum_{i \in N_s} b_i q_{it} \leq \lambda_t m Q \quad \forall t \in T. \tag{21}$$

The $\mathcal{M}_z$ model minimizes the objective function (20) subject to constraints (2)-(8), (10), (14)-(15) and (21). In the single-vehicle case ($m = 1$) and unlimited vehicle case ($m = n$), a modification of the aggregate fleet capacity is not necessary since a feasible VRP solution can always be found in phase 3 for each period; in these cases, lines 19-21 of Algorithm 1 are not executed. When the routing subproblem cannot find a feasible solution for a certain period, we reduce the λ_t for that period (line 25). Next, the $\mathcal{M}_z$ model is solved with the reduced capacity (line 11), and based on this solution the VRP_t subproblems are solved (line 13). If all VRP_t solutions are feasible, we update the incumbent solution. Since we have reduced some λ_t, this yields some unused aggregate fleet capacity. To explore the possible benefits from the unutilized capacity, we solve a modified $\mathcal{M}_z$ model. Let $\mathcal{R}_{kt}$ be the set of suppliers visited by vehicle k in period t. We replace constraints (21) with the following constraints for the periods where $\lambda_t < 1$ in the $\mathcal{M}_z$ model:

$$\sum_{i \in \mathcal{R}_{kt}} b_i q_{it} \leq Q \qquad \forall k \in \{1, ..., m\}, \forall t \in T | \lambda_t < 1. \tag{22}$$

Each constraint (22) relates to a vehicle that is used in a period with $\lambda_t < 1$. Then, we fix the node visit decisions z_{it} for these periods and obtain the $\mathcal{M}_z^{\mathcal{R}}$ model (line 19 of Algorithm 1). Using the $\mathcal{M}_z^{\mathcal{R}}$ model the algorithm can directly impose the vehicle capacity for each route, while we avoid the vehicle-indexed formulation which requires many more binary node visit variables for every vehicle as well as continuous quantity variables. We repeatedly solve $\mathcal{M}_z^{\mathcal{R}}$ with an updated approximation of the transportation cost (lines 18-22 of Algorithm 1) until the stopping criterion specified in Section 4.6 is met.

4.3. Phase 3: VRP_t Subproblems

Following each solution of the $\mathcal{M}_y$, $\mathcal{M}_z$ and $\mathcal{M}_z^{\mathcal{R}}$ models, we fix for each time period the current node visit decisions $\bar{z}_{it}$ and the shipment amounts $\bar{q}_{it}$. Therefore, we have to solve one VRP for each period. As discussed in the previous section, this routing problem can be infeasible for one or several periods. To solve this subproblem we use the tabu search heuristic of Cordeau et al. (1997), which allows violations of the vehicle capacity constraints through a penalty cost in the objective function. Based on this (possibly infeasible) solution, we update the transportation costs for the next iteration. To reduce the computing time in the tabu search heuristic, we limit the number of available vehicles in each period by a valid upper bound which significantly reduces the solution times. We first explain a naive procedure to merge two routes with a total load less than or equal to the vehicle capacity in the capacitated vehicle routing problem. Next, we will use this in our proof of the upper bound.

LEMMA 1. *Under the assumption that the triangle inequality holds, and given a solution to the capacitated vehicle routing problem, two routes with a load less than or equal to half of the vehicle capacity can be merged to obtain a new solution with a smaller or equal cost.*

Proof. It is sufficient to remove exactly one of the edges incident to the plant (depot) from each route and connect the resulting partial routes with exactly one new edge. This will result in a solution with a smaller or equal cost, while still satisfying the vehicle capacity. □

Note that although this procedure gives a shorter route compared to the original two, it does not necessarily produce the optimal route.

PROPOSITION 1. *Under the assumption that the triangle inequality holds, in a feasible instance of the capacitated vehicle routing problem with the set of nodes N_s to be visited, shipments q_i, $i \in N_s$, and route capacity Q, there exists an optimal solution with a number of routes smaller than or equal to* $\max\{1, \lceil \frac{2}{Q} \sum_{i \in N_s} q_i \rceil - 1\}$.

Proof. Consider a feasible solution with $m \geq 2$ routes, and let $k = \frac{\sum_{i \in N_s} q_i}{Q}$ and $\mathcal{Q}^j$ denote the load in route $j \in \{1, ..., m\}$. If $m \geq \lceil 2k \rceil$, we show that there exists a better or equivalent solution with $m-1$ routes. Let j_1 and j_2 be the two routes with the smallest loads among all routes. We will prove, by contradiction, that $\mathcal{Q}^{j_1} + \mathcal{Q}^{j_2} \leq Q$, in which case we can merge routes j_1 and j_2 according to Lemma 1 and arrive at a total number of routes equal to $m-1$. If $m-1 \geq \lceil 2k \rceil$, we repeat this route reduction procedure until we have a solution with $\lceil 2k \rceil - 1$ routes. Because this holds for any feasible solution, the number of routes in an optimal solution cannot exceed $\max\{1, \lceil 2k \rceil - 1\}$.

The proof, by contradiction, that $\mathcal{Q}^{j_1} + \mathcal{Q}^{j_2} \leq Q$, is as follows. Suppose that $\mathcal{Q}^{j_1} + \mathcal{Q}^{j_2} > Q$, then the larger load among $\mathcal{Q}^{j_1}$ and $\mathcal{Q}^{j_2}$ must be strictly larger than $Q/2$. We then have:

$$\mathcal{Q}^{j_1} + \mathcal{Q}^{j_2} + \sum_{j \in \{1, ..., m\} | j \neq j_1, j \neq j_2} \mathcal{Q}^j > Q + (m-2)\frac{Q}{2} = m\frac{Q}{2} \geq \lceil 2k \rceil \frac{Q}{2} \geq kQ = \sum_{i \in N_s} q_i,$$

which is a contradiction. The first inequality is valid because for j_1 and j_2, $\mathcal{Q}^{j_1} + \mathcal{Q}^{j_2} > Q$, and each of the remaining $m-2$ routes $(j \in \{1,...,m\} | j \neq j_1, j \neq j_2)$ has a load greater than or equal to j_1 and j_2 by assumption. The next expression is obtained by algebraic manipulation. The second inequality is valid based on the assumption that $m \geq \lceil 2k \rceil$. The next expression is trivial because $\lceil 2k \rceil \frac{Q}{2} \geq 2k\frac{Q}{2} = kQ$. The last expression is valid based on the definition of k. This leads to the contradiction that the sum of the route loads (first term) is strictly greater than the total shipments (last term). Therefore, the sum of the two smallest loaded routes (j_1 and j_2) cannot be strictly greater than the route capacity. □

Hence, we implement the following formula to specify the number of available vehicles for each VRP_t subproblem:

$$\bar{m}_t = \min\left\{m, \max\left\{1, \left\lceil \frac{2}{Q} \sum_{i \in N_s | z_{it}=1} b_i \bar{q}_{it} \right\rceil - 1\right\}\right\} \quad \forall t \in T. \tag{23}$$

Our sensitivity analysis on the effectiveness of this upper bound (Table 16, Appendix D) shows a solution time reduction potential of up to 4 times. Moreover, to control the running time of the heuristic, we set the number of tabu search iterations $\iota^{VRP} = \iota^V \sqrt{\bar{m}_t \sum_{i \in N_s | z_{it}=1} \bar{z}_{it}}$, for every period t, where ι^V is a parameter in our algorithm. To spend more time on promising solutions, we use a linearly varying value for the tabu search coefficient ι^V in the $[\iota^V_{\min}, \iota^V_{\max}]$ interval. When the previous solution is more than g (%) away from the incumbent, we let $\iota^V = \iota^V_{\min}$, and when it is better than or equal to the incumbent solution, we set $\iota^V = \iota^V_{\max}$.

4.4. Node Visit and Vehicle Dispatch Costs (σ_{it} and σ_{0t})

We tested three mechanisms to update the node visit costs for the next iteration. Having a complete solution at hand, the first mechanism (Marginal) approximates the node visit costs as follows: If node i is visited in the current solution, then we set $\sigma_{it} = (c_{i_p i} + c_{i i_s}) - c_{i_p i_s}$, where i_p and i_s are the predecessor and successor of node i in its current route. If node i is currently not served in period t, then we set σ_{it} equal to the cost of the cheapest insertion into an existing route. This is based on the assumption that when a node i is eliminated from its route, an acceptable route can be obtained by connecting the predecessor and successor nodes. Similarly, when inserting node i in a certain period t, an acceptable route can be obtained by the best insertion among all the routes in that period. Hence, σ_{it} can be seen as the (estimated) marginal transportation cost for visiting node i in period t. This marginal cost updating procedure is also used by Absi et al. (2014).

The second mechanism (TSP-share) splits the TSP cost of each route in each period over its nodes proportional to their direct shipment cost. Let c_{kt}^{TSP} be the route cost of vehicle k in period t, and $\mathcal{R}_{kt}$ be the set of suppliers visited by vehicle k in period t. We define

$$\sigma_{it} = c_{kt}^{TSP} \frac{c_{0i}}{\sum_{j \in \mathcal{R}_{kt}} c_{0j}} \quad \forall k \in \{1, ..., m\}, \forall t \in T, \forall i \in \mathcal{R}_{kt},$$

$$\sigma_{it} = \min_k \left\{ (c_{kt}^{TSP} + c_{kt}^i) \frac{c_{0i}}{\sum_{j \in \mathcal{R}_{kt} \cup \{i\}} c_{0j}} \right\} \quad \forall t \in T, \forall i \in N_s | z_{it} = 0,$$

where c_{kt}^i is equal to the cheapest insertion cost for non-visited node i into vehicle route k in period t. The last mechanism (VRP-share) divides the entire transportation cost of a certain period among the visited nodes proportional to their direct shipment cost. Let $c_t^{VRP} = \sum_{k=1}^m c_{kt}^{TSP}$ be the total transportation cost in period t, and $\mathcal{R}_t$ be the set of suppliers visited in period t. We define

$$\sigma_{it} = c_t^{VRP} \frac{c_{0i}}{\sum_{j \in \mathcal{R}_t} c_{0j}} \quad \forall t \in T, \forall i \in \mathcal{R}_t,$$

$$\sigma_{it} = (c_t^{VRP} + c_t^i) \frac{c_{0i}}{\sum_{j \in \mathcal{R}_t \cup \{i\}} c_{0j}} \quad \forall t \in T, \forall i \in N_s | z_{it} = 0,$$

where c_t^i is equal to the cheapest insertion cost for a non-visited node i into the available vehicle routes in period t.

The first and second mechanisms generally return better results than the last one. Our initial experiments revealed that by switching between the first two mechanisms, after using each for ι^U iterations, we generally get better results compared to using any one of them alone (line 14 of Algorithm 1). The maximum improvement by this hybrid update mechanism in the average solution cost is 1.9% compared to the marginal cost mechanism, 1.7% compared to the TSP-share mechanism, and 4.5% compared to the VRP-share mechanism. We report results with this mixed mechanism in Section 5.4. The effects on the algorithm's performance of using these three updating mechanisms as well as different starting node visit costs are presented in Appendix D.

Throughout the algorithm, we fix the vehicle dispatch cost $\sigma_{0t} = \sum_{i \in N_s} \bar{\sigma}_{it}/m$ where $\bar{\sigma}_{it}$ represents the initial node visit transportation cost. The analysis of the effect of different initial node visit costs on the algorithm performance is presented in Appendix D.

4.5. Inequalities LBI_z and LBI_y

To diversify the search, we rely on two types of inequalities inspired by the local branching approach of Fischetti and Lodi (2003). The first application of these inequalities as diversification constraints is presented in Fischetti et al. (2004) for a telecommunication network design problem. The first type of inequality, LBI_z, is specific to the $\mathcal{M}_z$ model and ensures that we do not return to a node visit pattern (and hence solution) we obtained before. The inequality

$$\sum_{i,t|\bar{z}_{it}=1}(1-z_{it})+\sum_{i,t|\bar{z}_{it}=0}z_{it}\geq r \tag{24}$$

forces at least r node visit variables to change value compared to the current solution. By varying r we can force different numbers of node visit changes in the next iteration of our algorithm. Our experiments show that if we let $r>1$ the algorithm reaches a better solution in a shorter time compared to the case of $r=1$. However, large values of r may remove some good quality solutions. We choose two different values for r. When the algorithm returns a better solution value compared to the previous iteration, we let $r=1$ to allow the algorithm to search the entire neighborhood of the current solution. In case a worse solution value (compared to the previous iteration) is obtained, we let $r=l$, where l is the number of periods in the planning horizon. We add one inequality to the $\mathcal{M}_z$ model at each iteration. Because these inequalities slow down the solution of the $\mathcal{M}_z$ model, we remove all the previous LBI_z inequalities when the setup schedule is changed (by means of the diversification mechanism), and we continue adding new ones in future iterations.

The second type of inequality, LBI_y, is specific to the $\mathcal{M}_y$ model and forces the model to obtain a new setup schedule. Therefore, it is used as a means of diversification. The inequality

$$\sum_{t|\bar{y}_t=1}(1-y_t)+\sum_{t|\bar{y}_t=0}y_t\geq 1 \tag{25}$$

forces at least one of the binary setup schedule variables to change value. We add one inequality to the $\mathcal{M}_y$ model each time we execute the diversification procedure and we keep these inequalities until the end of the algorithm. Adulyasak et al. (2014b) use this latter type of inequality to generate new setup schedules in their ALNS.

4.6. Stopping Conditions

The stopping condition for the overall algorithm (line 28 of Algorithm 1) is a maximum number of iterations, ι^A. To terminate the search for a local optimum within a specific setup schedule and introduce a diversification step (line 3 of Algorithm 1), we consider two stopping conditions. The search procedure stops after a maximum number of local search iterations, ι^L, or after a number of iterations without incumbent solution improvement, ι^N. Whenever one of these stopping conditions is met, the algorithm stops the local search, adds the associated LBI_y and solves the $\mathcal{M}_y$ model to find another setup schedule (lines 4-9 of Algorithm 1). We allow the algorithm to use the $\mathcal{M}_z^{\mathcal{R}}$ model only when it has performed at least ι^s iterations. This is to avoid wasting time with the very first solutions. The algorithm also runs the $\mathcal{M}_z^{\mathcal{R}}$ model only for the cases where the current solution obtained from the $\mathcal{M}_z$ model (and subsequent VRP_t subproblems) is close enough to the incumbent solution. More specifically, if the gap is less than g (%) the algorithm starts using the $\mathcal{M}_z^{\mathcal{R}}$ model to fix some vehicle routes as explained in Section 4.2. The $\mathcal{M}_z^{\mathcal{R}}$ model is allowed to be run until a maximum of $\iota^{\mathcal{R}}$ iterations is reached or until at any iteration it fails to return a solution with a gap less than $g^{\mathcal{R}}$ (%) from the incumbent solution. This condition corresponds to line 22 of Algorithm 1. The specific setting for the algorithm parameters and stopping conditions will be presented in the next section.

5. Computational Experiments

We test our algorithm on three different problems, the IRP, the PRP and the ARP, with a total of 4,068 instances. The IRP data sets were generated by Archetti et al. (2007) for the single-vehicle case and were later adapted to the multi-vehicle case by Coelho and Laporte (2013a) and by Desaulniers et al. (2015). The PRP data sets were introduced by Archetti et al. (2011) and by Boudia et al. (2005). We introduce the ARP instances in Section 5.1. Appendix A provides an overview of all the problem data sets.

We consider the same parameter setting when applying our algorithm to all data sets. The maximum number of algorithm iterations ι^A is set to 200 and the number of local search iterations ι^L is set to 80. The maximum number of non-improving iterations ι^N is set to 60. A maximum number of $\mathcal{M}_z^{\mathcal{R}}$ model iterations ($\iota^{\mathcal{R}}$) equal to 10 is considered. The values of the minimum and maximum tabu search iteration coefficients ($\iota^V_{\min}$ and $\iota^V_{\max}$) are 100 and 500, respectively. We let the algorithm switch between the marginal and TSP-share mechanisms every 7 iterations (ι^U). The values of g and $g^{\mathcal{R}}$ are set to 3% and 0.3%, respectively. We set the initial node visit cost equal to $c_{0i}/2$, where c_{0i} is the cost of the edge between the plant and node i. We explain the details of the parameter setting procedure in Appendix B.

5.1. ARP Test Instances

We use the PRP data sets of Archetti et al. (2011) as a base for developing our ARP data sets. For each test instance, the number of nodes, their position and the distance function as well as the number of time periods and vehicles have been kept the same as in the corresponding Archetti et al. (2011) instance. Note that the nodes are suppliers in the ARP, but represent customers in the IRP and PRP.

As in Archetti et al. (2011), we consider an unlimited production capacity. The component supply at each node, s_{it}, is constant and equal to the amount used for the demand in Archetti et al. (2011). The demand at the central plant, d_t, is set equal to the average amount of all the suppliers' production rates. We randomly generate an integer number according to a uniform distribution in the interval [1,10] for each component's size, b_i. Then, to adjust the vehicle capacity (Q) we multiply the values given in Archetti et al. (2011) by a factor of 10. We set the component inventory capacities at the suppliers (L_i) the same as the retailers' capacities presented in Archetti et al. (2011). We assume an uncapacitated storage for the components at the plant. We consider a uniformly distributed random integer between 2 to 4 times the product demand of a period as the storage limit, K. The unit component holding cost at the suppliers, h_i, is set the same as in Archetti et al. (2011). The unit component holding cost at the plant, r_i, is set equal to a uniform random integer over the $[h_i, 2h_i]$ interval. To generate the unit product holding cost, r_0, we select a uniformly distributed random integer over the interval $[\sum_{i\in N_s} r_i, 2\sum_{i\in N_s} r_i]$. The initial inventory of the components at the suppliers, I_{i0}, is set equal to the amount that Archetti et al. (2011) established for the customers. The initial inventory of the final product at the plant, F_{00}, is set randomly in the interval from 0 to the demand of two periods ($[0, 2d_t]$). To avoid infeasibility and meet the final product demand, we need to have enough initial component inventory at the plant. Therefore, we set for each component i the initial inventory F_{i0} equal to $\max\{0, \sum_{t\in T}(d_t - s_{it}) - I_{i0} - F_{00}\}$. Table 2 presents an overview of the ARP instance parameters.

Table 2 ARP test instances

	Set 1	Set 2	Set 3
# of instances (SA‡)	480	480	480
# of components (SA‡): n	14	50	100
# of periods (SA‡): l	6	6	6
# of suppliers (SA‡): n	14	50	100
# of trucks (SA‡): m	1	UL†	UL†
Component supply: s_{it}		SA‡	
Production capacity: C		SA‡	
Demand (final product): d_t		$(\sum_{i=1}^{n} s_{it})/n$	
Item size: b_i		UDRI†† $[1, 10]$	
Vehicle capacity: Q		SA‡ by a factor of 10	
Supplier inventory capacity: L_i		SA‡	
Plant inventory capacity for components: L		UL†	
Plant inventory capacity for final product: K		UDRI†† $[2d_t, 4d_t]$	
Supplier initial inventory: I_{i0}		SA‡	
Plant initial inventory of components: F_{i0}		$\max\{0, \sum_{t\in T}(d_t - s_{it}) - I_{i0} - F_{00}\}$	
Plant initial inventory of final product: F_{00}		UDRI†† $[0, 2d_t]$	
Supplier and plant x,y coordinates		SA‡	
Unit production cost: u		SA‡	
Production setup cost: f		SA‡	
Unit transportation cost		SA‡	
Travel distance		SA‡	
Supplier unit holding cost: h_i		SA‡	
Plant unit component holding cost: r_i		UDRI†† $[h_i, 2h_i]$	
Plant unit final product holding cost: r_0		UDRI†† $[\sum_{i\in N_s} r_i, 2\sum_{i\in N_s} r_i]$	

† UL: Unlimited
‡ SA: The same as Archetti et al. (2011)
†† UDRI: Uniformly Distributed Random Integer

5.2. Algorithm Implementation

Some modules of the algorithm become redundant for some problems or data sets. The main modules of the algorithm are the $\mathcal{M}_y, \mathcal{M}_z$ and $\mathcal{M}_z^{\mathcal{R}}$ models and VRP_t subproblems. The aim of the $\mathcal{M}_y$ model is to find proper setup schedules. Therefore, this module is not applicable in the case of the IRP. The module with which we find node visit schedules, the $\mathcal{M}_z$ model, is relevant and necessary for all data sets and problems. The $\mathcal{M}_z^{\mathcal{R}}$ model is only required for the data sets with a limited number of vehicles ($1 < m < n$). We present the $\mathcal{M}_y$ model for the PRP and $\mathcal{M}_z$ models for the IRP and PRP in Appendix C. We solve the $\mathcal{M}_y$, $\mathcal{M}_z$ and $\mathcal{M}_z^{\mathcal{R}}$ models with CPLEX 12.6. Because all problems take the routing decisions into account, we must solve the VRP_t subproblems in every case.

5.3. Benchmark Algorithms

Since the ARP is a new problem, there is no algorithm to use as a benchmark. Consequently, we developed two lower bounding procedures as a basis for comparison. Furthermore, we validate the quality of our algorithm by applying it to the IRP and PRP standard test instances. For the IRP and PRP, we select the state-of-the-art algorithms as basis for comparison. Some of these are exact algorithms which we include for two reasons: to show the difference in running times and to consider their best found solutions in our comparison. We set the acronyms for each algorithm (including ours) by the authors' family name initials, followed by the implemented method identifier. For example, BC stands for branch-and-cut algorithm. Note that SV and MV in the data set names refer to single-vehicle and multi-vehicle instances, respectively.

It is difficult to make comparisons between different platforms and algorithms. It becomes more complicated when different numbers of threads are used. Therefore, we report the running times for each benchmark algorithm as it was presented in the original paper. To have an approximation of the speed of each employed platform, we additionally report a time adjustment factor for each benchmark algorithm using the CPU marks presented in CPU-benchmark (2017). Table 3 provides the list of benchmark algorithms, their running platform, number of threads, time adjustment factor and solver version. Since some of the algorithms for the IRP are applied to only a subset of the instances, we provide more details in Table 4 on the number of instances each algorithm was applied to.

Table 3 Benchmark algorithms, the running platforms and standard MILP solver for the IRP and PRP data sets

Prob	Reference	Name	Sol	CPU	#Thread	TAF	Solver
IRP	Archetti et al. (2007)	ABLS-BC	E	Pentium IV 2.8GHz	Def	323	CPLEX 9.0
	Coelho and Laporte (2013b)	CL-BC	E	Xeon 2.67GHz	6	7,518	CPLEX 12.3
	Archetti et al. (2017)	ABS-H	HM	Xeon W3680, 3.33GHz	8	9,211	CPLEX 12.5
	Avella et al. (2017)	ABW-BC	E	Core i7-2620, 2.70GHz	1	3,825	Xpress 7.6
	Desaulniers et al. (2015)	DRC-BPC	E	Core i7-2600 3.4GHz	1	8,220	CPLEX 12.2
	Archetti et al. (2012)	ABHS-H	H	Intel Dual Core 1.86GHz	Def	2,288	CPLEX 10.1
	Coelho et al. (2012)	CCL-ALNS	M	Intel T7700, 2.4GHz	Def	1,419	-
	Adulyasak et al. (2014a)	ACJ-ALNS-1000	M	2.10GHz Duo CPU PC	Def	6,340	CPLEX 12.3
PRP	Archetti et al. (2011)	ABPS-BC ABPS-H	E H	AMD Athlon 64, 2.89GHz Intel Core 2, 2.40GHz	Def 1	437 1,440	CPLEX 10.1 -
	Boudia and Prins (2009)	BP-MA	M	2.30GHz PC	1	3,298	-
	Bard and Nananukul (2009)	BN-TS	M	2.53GHz PC	1	3,538	-
	Armentano et al. (2011)	ASL-TS	M	Pentium IV 2.8GHz	1	323	-
	Adulyasak et al. (2014b)	ACJ-ALNS-500 ACJ-ALNS-1000	M M	2.10GHz Duo CPU PC	Def	6,340	CPLEX 12.2
	Absi et al. (2014)	AADF-MS AADF-DMS AADF-VRP AADF-MTSP	H H H H	Xeon 2.67GHz PC	Def	7,518	CPLEX 12.1
	Solyalı and Süral (2017)	SS-H	H	2.40GHz PC	12	3,538	CPLEX 12.5
Both	This paper	CCJ-DH	H	Xeon X5650 2.67GHz	1	7,518	CPLEX 12.6

Note. Prob: Problem, Sol: Solution approach, E: Exact, M: Metaheuristic, H: Heuristic, Def: Default
TAF: Time adjustment factor according to CPU-benchmark (2017)

Table 4 Number of instances each benchmark algorithm is applied to for the IRP and PRP data sets

Name of the Algorithm					ABLS-BC	CL-BC	ABS-H	ABW-BC	DRC-BPC	ABHS-H	CCL-ALNS	ABPS-BC	ABPS-H	BP-MA	BN-TS	ASL-TS	ACJ-ALNS-500	ACJ-ALNS-1000	AADF-MS	AADF-DMS	AADF-VRP	AADF-MTSP	SS-H	CCJ-DH
Prob	Set	m	Class	Size	E	E	HM	E	E	H	M	E	H	M	M	M	M	M	H	H	H	H	H	H
IRP	SV-I1	1	-	160	160	160	-	-	-	160	160	-	-	-	-	-	-	-	-	-	-	-	-	160
	MV-I1	2	-	160	-	160	160	50	158	-	-	-	-	-	-	-	-	150	-	-	-	-	-	160
		3	-	160	-	160	160	50	160	-	-	-	-	-	-	-	-	150	-	-	-	-	-	160
		4	-	160	-	160	160	50	160	-	-	-	-	-	-	-	-	-	-	-	-	-	-	160
		5	-	158	-	158	158	49	158	-	-	-	-	-	-	-	-	-	-	-	-	-	-	158
	SV-I2	1	-	60	-	60	-	-	-	60	-	-	-	-	-	-	-	-	-	-	-	-	-	60
	MV-I2	2	-	60	-	40	60	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	60
		3	-	60	-	40	60	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	60
		4	-	60	-	-	60	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	60
		5	-	60	-	-	60	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	60
PRP	SV-A1	1	1	120	-	-	-	-	-	-	-	120	120	-	-	-	120	120	120	120	-	-	120	120
		1	2	120	-	-	-	-	-	-	-	120	120	-	-	-	120	120	120	120	-	-	120	120
		1	3	120	-	-	-	-	-	-	-	120	120	-	-	-	120	120	120	120	-	-	120	120
		1	4	120	-	-	-	-	-	-	-	116	120	-	-	-	120	120	120	120	-	-	120	120
	MV-A2	UL	1	120	-	-	-	-	-	-	-	-	120	-	-	-	120	120	-	-	120	120	120	120
		UL	2	120	-	-	-	-	-	-	-	-	120	-	-	-	120	120	-	-	120	120	120	120
		UL	3	120	-	-	-	-	-	-	-	-	120	-	-	-	120	120	-	-	120	120	120	120
		UL	4	120	-	-	-	-	-	-	-	-	120	-	-	-	120	120	-	-	120	120	120	120
	MV-A3	UL	1	120	-	-	-	-	-	-	-	-	120	-	-	-	120	120	-	-	120	120	120	120
		UL	2	120	-	-	-	-	-	-	-	-	120	-	-	-	120	120	-	-	120	120	120	120
		UL	3	120	-	-	-	-	-	-	-	-	120	-	-	-	120	120	-	-	120	120	120	120
		UL	4	120	-	-	-	-	-	-	-	-	120	-	-	-	120	120	-	-	120	120	120	120
	MV-B1	5	-	30	-	-	-	-	-	-	-	-	-	30	30	30	30	30	-	-	30	30	30	30
	MV-B2	9	-	30	-	-	-	-	-	-	-	-	-	30	30	30	30	30	-	-	30	30	30	30
	MV-B3	13	-	30	-	-	-	-	-	-	-	-	-	30	30	30	30	-	-	-	30	30	30	30
Total				2628	160	938	878	199	636	220	160	476	1440	90	90	90	1530	1800	480	480	1050	1050	1530	2628

Note. E: Exact, H: Heuristic, M: Metaheuristic, MV: Multi-Vehicle, Prob: Problem, SV: Single-Vehicle, UL: Unlimited

5.4. Computational Results for the IRP and PRP Data Sets

Tables 5, 6 and 7 present the computational results and comparison between our algorithm, CCJ-DH, and the benchmark algorithms. Table 5 presents the average gap of the different algorithms applied to the IRP and PRP data sets. We calculate the percentage gap for each solution to each instance with respect to the previous best known solution so far (not including CCJ-DH). Then, for each class and number of vehicles (m) of a data set, we calculate the average gaps of the different algorithms.

Table 6 presents the number of best solutions found by different algorithms. Because for some small instances it is possible that CCJ-DH finds the same previous best found solution, we also present the number of new best solutions (NBS) in the last column of this table. Table 7 shows the average running times (in seconds) of the different algorithms.

For the SV-I1 data set, the exact BC algorithms (ABLS-BC and CL-BC) solved all the instances to optimality. ABHS-H and CCL-ALNS are the state-of-the-art heuristics on this

Table 5 Average gaps by different algorithms applied to IRP and PRP data sets (%)

Name of the Algorithm					ABLS-BC	CL-BC	ABS-H	ABW-BC	DRC-BPC	ABHS-H	CCL-ALNS	ABPS-BC	ABPS-H	BP-MA	BN-TS	ASL-TS	ACJ-ALNS-500	ACJ-ALNS-1000	AADF-MS	AADF-DMS	AADF-VRP	AADF-MTSP	SS-H	CCJ-DH
Prob	Set	m	Class	Size	E	E	HM	E	E	H	M	E	H	M	M	M	M	M	H	H	H	H	H	H
IRP	SV-I1	1	-	160	**0**	**0**	-	-	-	0.05	0.46	-	-	-	-	-	-	-	-	-	-	-	-	1.62
	MV-I1	2	-	160	-	**0**	0.21	1.3	20.2	-	-	-	-	-	-	-	-	7.01	-	-	-	-	-	2.6
		3	-	160	-	**0.27**	0.67	2.63	18.41	-	-	-	-	-	-	-	-	7.2	-	-	-	-	-	2.4
		4	-	160	-	5.32	**1.34**	3.75	17.05	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2.55
		5	-	158	-	10.53	**1.5**	4.08	14.85	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2.75
	SV-I2	1	-	60	-	10.91	-	-	-	**0.27**	-	-	-	-	-	-	-	-	-	-	-	-	-	3.51
	MV-I2	2	-	60	-	61.32	0.12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**-1.82**
		3	-	60	-	106.28	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**-2.86**
		4	-	60	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**-4.73**
		5	-	60	-	-	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**-4.9**
PRP	SV-A1	1	1	120	-	-	-	-	-	-	-	**0**	2.21	-	-	-	1.73	1.7	0.09	0.13	-	-	0.03	0.24
		1	2	120	-	-	-	-	-	-	-	**0**	0.3	-	-	-	0.36	0.36	0.01	0.02	-	-	**0**	0.03
		1	3	120	-	-	-	-	-	-	-	**0**	3.65	-	-	-	9.1	8.43	0.57	0.72	-	-	0.18	1.49
		1	4	120	-	-	-	-	-	-	-	**0**	0.9	-	-	-	0.96	0.96	0.04	0.07	-	-	0.05	0.13
	MV-A2	UL	1	120	-	-	-	-	-	-	-	-	2.08	-	-	-	1.14	1.13	-	-	0.17	1.09	0.05	**-0.06**
		UL	2	120	-	-	-	-	-	-	-	-	0.38	-	-	-	0.17	0.17	-	-	0.05	0.09	**0.01**	**0.01**
		UL	3	120	-	-	-	-	-	-	-	-	3.5	-	-	-	3.76	3.52	-	-	0.99	2.73	0.09	**-0.2**
		UL	4	120	-	-	-	-	-	-	-	-	1.25	-	-	-	0.2	0.19	-	-	0.1	0.46	0.06	**-0.03**
	MV-A3	UL	1	120	-	-	-	-	-	-	-	-	2.28	-	-	-	1.06	1	-	-	0.22	1.87	**0.07**	0.18
		UL	2	120	-	-	-	-	-	-	-	-	0.35	-	-	-	0.31	0.3	-	-	0.2	0.3	**0.18**	**0.18**
		UL	3	120	-	-	-	-	-	-	-	-	3.66	-	-	-	3.83	3.65	-	-	1.22	3.8	**0.15**	1.07
		UL	4	120	-	-	-	-	-	-	-	-	1.34	-	-	-	0.4	0.38	-	-	0.3	0.75	**0.11**	**0.11**
	MV-B1	5	-	30	-	-	-	-	-	-	-	-	-	14.55	7.7	5.37	1.15	1.04	-	-	1.65	1.41	**0.03**	0.78
	MV-B2	9	-	30	-	-	-	-	-	-	-	-	-	13.26	12.91	8.71	0.97	0.96	-	-	0.85	1.32	**0.07**	1.21
	MV-B3	13	-	30	-	-	-	-	-	-	-	-	-	16.09	19.95	10.29	1.61	-	-	-	0.36	1.93	2.24	**0.18**

Note. The best average gap at each row is presented with the bold font.

data set. They were able to find 125 and 72 optimal solutions, respectively, and finished with small gaps. Our algorithm was able to find 31 of the optimal solutions. The average gap of our algorithm on this data set is 1.62%, which is higher than the gap of the other algorithms. For the MV-I1 data set, the state-of-the-art heuristic algorithm is ABS-H. It was applied to all the instances in this set and obtained 261 best solutions with average gaps ranging from 0.21% to 1.5%. ACJ-ALNS-1000 found 26 best upper bounds (BUBs) in total with gaps of more than 7%. CCJ-DH obtained solutions with an average gap of 2.4% to 2.75% and found 126 best solutions among which it was successful to obtain 66 new best solutions.

For the SV-I2 data set, results are available for the CL-BC and ABHS-H algorithms. The first algorithm (which is a BC) spent on average more than 64,000 seconds to solve the instances in the set and obtained 30 BUBs. This algorithm has an average gap of more than 10.9%. The ABHS-H heuristic spent an average computing time of 3,630 seconds and obtained 31 BUBs. The UBs obtained by this algorithm are generally of high quality,

Table 6 Number of BUBs found by different algorithms applied to IRP and PRP data sets

Name of the Algorithm					ABLS-BC	CL-BC	ABS-H	ABW-BC	DRC-BPC	ABHS-H	CCL-ALNS	ABPS-BC	ABPS-H	BP-MA	BN-TS	ASL-TS	ACJ-ALNS-500	ACJ-ALNS-1000	AADF-MS	AADF-DMS	AADF-VRP	AADF-MTSP	SS-H	CCJ-DH	
Prob	Set	m	Class	Size	E	E	HM	E	E	H	M	E	H	M	M	M	M	M	H	H	H	H	H	H	NBS
IRP	SV-I1	1	-	160	**160**	**160**	-	-	-	125	72	-	-	-	-	-	-	-	-	-	-	-	-	31	0
	MV-I1	2	-	160	-	**158**	98	22	84	-	-	-	-	-	-	-	-	14	-	-	-	-	-	28	0
		3	-	160	-	**142**	74	2	93	-	-	-	-	-	-	-	-	12	-	-	-	-	-	31	13
		4	-	160	-	**108**	50	5	95	-	-	-	-	-	-	-	-	-	-	-	-	-	-	29	22
		5	-	158	-	77	39	6	**102**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	38	31
	SV-I2	1	-	60	-	30	-	-	-	**31**	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0
	MV-I2	2	-	60	-	8	12	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**40**	40
		3	-	60	-	0	15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**45**	45
		4	-	60	-	-	9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**51**	51
		5	-	60	-	-	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**58**	58
PRP	SV-A1	1	1	120	-	-	-	-	-	-	-	**119**	6	-	-	-	1	1	81	70	-	-	71	19	0
		1	2	120	-	-	-	-	-	-	-	**120**	5	-	-	-	0	0	81	68	-	-	71	20	0
		1	3	120	-	-	-	-	-	-	-	**120**	2	-	-	-	0	0	52	44	-	-	67	5	0
		1	4	120	-	-	-	-	-	-	-	**116**	14	-	-	-	1	1	85	76	-	-	70	32	0
	MV-A2	UL	1	120	-	-	-	-	-	-	-	-	0	-	-	-	0	0	-	-	9	2	10	**101**	99
		UL	2	120	-	-	-	-	-	-	-	-	4	-	-	-	2	2	-	-	12	8	**49**	47	46
		UL	3	120	-	-	-	-	-	-	-	-	0	-	-	-	0	0	-	-	2	2	27	**92**	89
		UL	4	120	-	-	-	-	-	-	-	-	0	-	-	-	0	1	-	-	10	6	29	**75**	74
	MV-A3	UL	1	120	-	-	-	-	-	-	-	-	9	-	-	-	0	0	-	-	5	0	28	**79**	79
		UL	2	120	-	-	-	-	-	-	-	-	24	-	-	-	0	0	-	-	17	4	29	**46**	46
		UL	3	120	-	-	-	-	-	-	-	-	7	-	-	-	0	0	-	-	8	0	**77**	28	28
		UL	4	120	-	-	-	-	-	-	-	-	16	-	-	-	0	0	-	-	17	3	38	**46**	46
	MV-B1	5	-	30	-	-	-	-	-	-	-	-	-	0	0	1	0	1	-	-	0	0	**26**	2	2
	MV-B2	9	-	30	-	-	-	-	-	-	-	-	-	0	0	0	0	1	-	-	2	1	**26**	0	0
	MV-B3	13	-	30	-	-	-	-	-	-	-	-	-	0	0	0	0	-	-	-	9	3	6	**12**	12
Total (All Instances)				2628	160	683	299	35	374	156	72	475	87	0	0	1	4	33	299	258	91	29	624	955	781
Total (LSMV† Instances)				1290	0	8	38	0	0	0	0	0	60	0	0	1	2	5	0	0	91	29	345	722	715

Note. The largest number of obtained BUBs at each row is presented with the bold font. NBS: New best solutions.
† Large-scale multi-vehicle.

resulting in an average gap of 0.27%. CCJ-DH spent about 6,700 seconds on average for the instances in this set and ended up with an average gap of around 3.5%.

For the MV-I2 data set there are two algorithms to compare with: CL-BC and ABS-H. Because the size of the instances and the number of available vehicles are larger compared to the MV-I1 data set, the CL-BC algorithm was not able to solve the instances with $m = 4$ and 5, and $n = 200$. This algorithm left average gaps of more than 61% and 106% for the instances with $m = 2$ and 3, respectively and found only 8 BUBs (among the instances with $m = 2$) while spending 86,400 seconds on every instance in the set. ABS-H was also successful on this data set by finding 38 BUBs. CCJ-DH outperformed the two existing approaches on this data set, finding 194 new best solutions which counts for more than 80% of the instances in this data set. Our algorithm obtained average gaps between -1.82% and

Table 7 Average running time of different algorithms applied to IRP and PRP data sets (seconds)

Name of the Algorithm					ABLS-BC	CL-BC	ABS-H	ABW-BC	DRC-BPC	ABHS-H	CCL-ALNS	ABPS-BC	ABPS-H	BP-MA	BN-TS	ASL-TS	ACJ-ALNS-500	ACJ-ALNS-1000	AADF-MS	AADF-DMS	AADF-VRP	AADF-MTSP	SS-H	CCJ-DH
Prob	Set	m	Class	Size	E	E	HM	E	E	H	M	E	H	M	M	M	M	M	H	H	H	H	H	H
IRP	SV-I1	1	-	160	620	19	-	-	-	459	498	-	-	-	-	-	-	-	-	-	-	-	-	48
	MV-I1	2	-	160	-	4099	1259	2729	4121	-	-	-	-	-	-	-	-	34	-	-	-	-	-	68
		3	-	160	-	15319	1585	3467	4124	-	-	-	-	-	-	-	-	39	-	-	-	-	-	69
		4	-	160	-	23884	800	3600	3862	-	-	-	-	-	-	-	-	-	-	-	-	-	-	74
		5	-	158	-	28244	914	3600	3680	-	-	-	-	-	-	-	-	-	-	-	-	-	-	65
	SV-I2	1	-	60	-	64509	-	-	-	3630	-	-	-	-	-	-	-	-	-	-	-	-	-	6668
	MV-I2	2	-	60	-	86400	4066	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	5657
		3	-	60	-	86400	4540	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4209
		4	-	60	-	-	4257	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	5132
		5	-	60	-	-	4418	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	5527
PRP	SV-A1	1	1	120	-	-	-	-	-	-	-	445	†	-	-	-	5	9	251	243	-	-	5	18
		1	2	120	-	-	-	-	-	-	-	11	†	-	-	-	5	9	214	210	-	-	5	18
		1	3	120	-	-	-	-	-	-	-	81	†	-	-	-	5	9	237	233	-	-	5	17
		1	4	120	-	-	-	-	-	-	-	527	†	-	-	-	5	9	217	218	-	-	5	18
	MV-A2	UL	1	120	-	-	-	-	-	-	-	-	11	-	-	-	29	50	-	-	23	315	5	400
		UL	2	120	-	-	-	-	-	-	-	-	12	-	-	-	28	50	-	-	23	288	14	344
		UL	3	120	-	-	-	-	-	-	-	-	9	-	-	-	24	43	-	-	26	335	16	309
		UL	4	120	-	-	-	-	-	-	-	-	11	-	-	-	26	44	-	-	26	330	25	434
	MV-A3	UL	1	120	-	-	-	-	-	-	-	-	188	-	-	-	136	249	-	-	86	514	324	2125
		UL	2	120	-	-	-	-	-	-	-	-	217	-	-	-	125	221	-	-	76	497	51	1947
		UL	3	120	-	-	-	-	-	-	-	-	168	-	-	-	107	191	-	-	75	509	350	1461
		UL	4	120	-	-	-	-	-	-	-	-	181	-	-	-	108	189	-	-	87	507	125	2213
	MV-B1	5	-	30	-	-	-	-	-	-	-	-	-	173	331	317	298	481	-	-	551	1653	2464	3559
	MV-B2	9	-	30	-	-	-	-	-	-	-	-	-	1108	976	1148	1405	1570	-	-	2054	9483	7487	9811
	MV-B3	13	-	30	-	-	-	-	-	-	-	-	-	4098	2492	3926	5794	-	-	-	4197	19270	16365	15891

† The computing times are negligible.

-4.9%. The larger the number of nodes and the number of vehicles, the better the results obtained by CCJ-DH compared to ABS-H. This is an interesting result since ABS-H is a specialized algorithm for the multi-vehicle IRP.

For the SV-A1 data set, there are five algorithms available in the benchmark set that were applied to all the instances: ABPS-BC, ABPS-H, ACJ-ALNS with 500 and 1000 iterations, AADF-MS, AADF-DMS and SS-H. Among the heuristic and metaheuristic algorithms, the specialized algorithms of AADF-MS, AADF-DMS and SS-H are the best performing ones. ABPS-H, ACJ-ALNS (with 500 and 1000 iterations) and SS-H are the only benchmark algorithms that were applied to all three data sets of Archetti et al. (2011). While ABPS-H generally obtained better results than ACJ-ALNS for SV-A1 with almost negligible computing times, both are outperformed by CCJ-DH in terms of the number of BUBs and average gaps.

There are five sophisticated heuristic or metaheuristic algorithms available for the MV-A2 and MV-A3 data sets. Due to the size of the instances ($n = 50$ and 100, $l = 6$), no exact algorithm has yet been applied to these sets. The results presented in Tables 5 and 6 show that our algorithm and SS-H outperform all other algorithms on these two data sets both in total number of BUBs and average gaps. Over all the eight subclasses of MV-A2 and MV-A3, our algorithm provides an equal or better performance with respect to the gap for six subclasses compared to SS-H. Furthermore, our algorithm found 514 BUBs, while SS-H found 287 BUBs. Our algorithm was able to improve the overall previous best known solutions obtained by other benchmark algorithms on MV-A1, MV-A3 and MV-A4.

Seven different algorithms were tested on the MV-B1, MV-B2, and MV-B3 data sets: BP-MA, BN-TS, ASL-TS, ACJ-ALNS with 500 and 1000 iterations, AADF-VRP, AADF-MTSP and SS-H. On MV-B1 and MV-B2, SS-H is the best performing algorithm. However, on the MV-B3 which includes the largest PRP instances, CCJ-DH is the best algorithm with an average gap of 0.18%. SS-H returned a large gap of 2.24% on this data set. Overall, SS-H and CCJ-DH are the best performing algorithms (non-dominated ones) on these three data sets. The average gap of CCJ-DH on all the 90 instances in these data sets is 0.72%, performing better than SS-H with an overall average gap of 0.78%.

On all IRP and PRP data sets with 2,628 instances, CCJ-DH was able to find 955 BUBs out of which 781 are new best solutions. Our algorithm shows consistent performance especially on the large-scale multi-vehicle instances of both IRP and PRP. For this family of instances, CCJ-DH successfully obtains improved solutions compared to the previous BUBs found in the literature by the specialized algorithms. Among the 1,290 large-scale multi-vehicle instances of IRP and PRP data sets (240 instances of MV-I2, 960 instances of MV-A2 and MV-A3 and 90 instances of MV-B1, MV-B2 and MV-B3), CCJ-DH found 715 new best solutions. The algorithm also finished with the best or one of the best average gaps among the other benchmark algorithms. Moreover, CCJ-DH is the only algorithm that has been applied to all the IRP and PRP data sets. ACJ-ALNS-1000 is the only other algorithm that has been applied to both the IRP and PRP problems. This metaheuristic was developed specifically for the PRP (Adulyasak et al. 2014b) and was next applied to a limited set of multi-vehicle IRP instances (Adulyasak et al. 2014a). The results in Table 5 indicate that CCJ-DH obtains improved gaps compared to ACJ-ALSN-1000 in all the tested classes, except for MV-B2.

In the existing algorithms for the IRP and PRP, we observe imbalances between the CPU times. Because we worked with the same parameters for all problems and data sets, it was impossible to find one setting that led to similar CPU times for all classes compared to the state-of-the-art algorithms.

5.5. Computational Results for the ARP Data Sets

On the ARP data sets, we compare our algorithm against a truncated BC method implemented in C++ with the CPLEX callable library and a time limit of 12 hours. In the $\mathcal{M}_{ARP}$ model, we include another type of SEC (Archetti et al. 2011) in addition to constraints (12), to strengthen the LP relaxation of $\mathcal{M}_{ARP}$:

$$\sum_{(i,j)\in E(S)} x_{ijt} \leq \sum_{i\in S} z_{it} - z_{et} \quad \forall S \subseteq N_s, |S| \geq 2, \forall e \in S, \forall t \in T. \tag{26}$$

We add SECs dynamically through the search whenever they are violated. To this end, we use the CVRP package of Lysgaard et al. (2004) for separation. Moreover, we add the following valid inequalities together with constraints (19) a priori to the model:

$$z_{it} \leq z_{0t} \quad \forall i \in N_s, \forall t \in T, \tag{27}$$

$$x_{ijt} \leq z_{it} \quad and \quad x_{ijt} \leq z_{jt} \quad \forall (i,j) \in E(N_s), \forall t \in T. \tag{28}$$

We also implemented another lower bounding method for the ARP instances. Our initial experiments showed that when relaxing GFSEC, i.e. constraints (12), from the $\mathcal{M}_{ARP}$ model, CPLEX is able to solve the resulting MIP for large ARP instances in an average of 60 seconds. However, the integral solution may have subtours in each period. Similar to the cutting plane method, we iteratively add GFSEC for the violated subtours and re-solve the new MIP (MIP-CP). Note that at each iteration the solution time grows significantly due to the newly added SECs and the marginal benefit of adding them becomes smaller. We observed that after five hours this method is no longer able to effectively improve the solutions (lower bound) for the MV-C2 and MV-C3 data sets. Because the BC method is able to solve the MV-C1 instances to optimality in a very short time, we did not apply the MIP-CP method to these instances.

Table 8 presents the performance of CCJ-DH on the ARP data sets. Columns five and six in this table show the number of upper bounds (UBs) and best upper bounds (BUBs) obtained by the BC method, respectively. The next column presents the average gap (%) of the BC UBs with respect to (w.r.t.) BUBs (found either by BC or CCJ-DH). The rest of the columns for the BC method show the number of optimal solutions, the number of best lower bounds (BLBs) found either by BC or MIP-CP, the average gap (%) of the BC method (compared to its own LB), and the average gap (%) of its lower bounds (LBs) w.r.t.

BLBs (found either by BC or MIP-CP). The two columns for MIP-CP present the number of BLBs and the average gap (%) of its LBs w.r.t. BLBs. Note that MIP-CP does not produce a feasible solution. The four columns for CCJ-DH show the solution time, number of BUBs, and the average gap (%) of its solutions w.r.t. BUBs and BLBs, respectively.

The BC method is able to solve every instance in the SV-C1 data set in less than 44 seconds, but for the other two data sets it reaches the time limit of 12 hours. It finds 304 feasible solutions for MV-C2 within the time limit among which only 22 are better UBs compared to CCJ-DH. On the MV-C3 data set, the BC method is unable to find any feasible solution in the time limit. MIP-CP shows little LB improvement potential for the MV-C2 data set compared to BC, except for class four. However, the MIP-CP method proved to be efficient in obtaining 167 BLBs for the MV-C3 data set. On the small test instances, our heuristic provides good quality solutions, with an average gap between 0.3% and 1.1%, compared to the optimal solutions of a specialized BC approach. On the medium and large size instances, our algorithm generally provides high quality solutions compared to the best lower bounds found either by the BC approach or a specialized lower bound algorithm. These average gaps vary between 0.9% and 2.4%, except for the third class of MV-C2 and MV-C3, for which the gaps are close to 6% and 10%. The reason is the very large transportation cost for the instances in this class.

Table 8 CCJ-DH performance on ARP data sets.

				BC†							MIP-CP‡		CCJ-DH			
Set	m	Class	Size	# UB	# BUB	Gap-UB BUB	# Opt	# BLB	Gap CPLEX	Gap-LB BLB	# BLB	Gap-LB BLB	CPU (sec)	# BUB	Gap-UB BUB	Gap-UB BLB
SV-C1	1	1	120	120	120	0	120	120	0	0	-	-	43	23	0.3	0.3
	1	2	120	120	120	0	120	120	0	0	-	-	41	19	0.26	0.26
	1	3	120	120	120	0	120	120	0	0	-	-	42	19	1.07	1.07
	1	4	120	120	120	0	120	120	0	0	-	-	34	16	0.48	0.48
MV-C2	UL	1	120	73	3	47.32	0	114	48.01	0.21	6	0.56	603	117	0	1.54
	UL	2	120	76	7	47.35	0	117	47.91	0.01	3	0.61	592	113	0.01	1.54
	UL	3	120	60	1	62.98	0	114	64.61	0.19	6	2.56	468	119	0	5.98
	UL	4	120	95	11	26.85	0	67	27.39	0.14	53	0.18	914	109	0.02	0.88
MV-C3	UL	1	120	0	0	100	0	95	100	0.56	25	0.32	2967	120	0	2.4
	UL	2	120	0	0	100	0	93	100	0.47	27	0.3	2932	120	0	2.39
	UL	3	120	0	0	100	0	92	100	0.56	28	1.14	1971	120	0	9.81
	UL	4	120	0	0	100	0	33	100	1	87	0.11	4213	120	0	1.58

Note. BC: Branch-and-cut algorithm, MIP-CP: Cutting plane method with sequential MIPs.
† With a time limit of 12 hours and maximum 30 GB memory. The algorithm finds optimal solution for SV-C1 in less than 44 seconds for any instance in the set, and it reaches the time limit for both MV-C2 and MV-C3.
‡ With a time limit of 5 hours.

We further discuss the behavior of our algorithm in Appendix D. All instances, detailed solutions and results can be found at `http://chairelogistique.hec.ca/en/scientific-data/`.

6. Summary and Conclusion

This study fills a gap in the literature by introducing a MILP model for the integrated production, inventory and inbound routing problem. Although some similarities between the PRP and ARP exist, fundamental differences arise in the nature of the problem and in the modeling such as the presence of inventory of both the final product and the components at the plant. We present a compact formulation for the ARP ($\mathcal{M}_{ARP}$) and developed many test instances for this problem as well as an efficient heuristic algorithm. On the small test instances, our heuristic provides good quality solutions, compared to the optimal solutions of a specialized BC approach. On the medium and large size instances, our algorithm generally provides high quality solutions compared to the best obtained lower bounds either by the BC approach or a specialized lower bound algorithm, with the exception of the data sets with the high transportation cost.

We further test this algorithm on other problems of the same nature where the routing decisions are integrated with inventory management (and production planning): the IRP and the PRP. We consider standard data sets from the literature. These data sets include 2,628 instances ranging from small to very large-scale ones. We compare our results to those from the current state-of-the-art algorithms. Our algorithm presents acceptable results on the small data sets and outperforms specialized state-of-the-art algorithms for the large-scale multi-vehicle instances. We also outperform the only other algorithm that has been applied to both the IRP and PRP problems. Moreover, we show that the algorithm finds good quality solutions with different transportation cost update mechanisms as well as different initial node visit costs. We believe this shows the robustness of our decomposition approach. One of the most important contributions of this paper is the design of a unified algorithm that can be applied to different data sets of different problems (ARP, PRP and IRP) with the same parameter setting.

Acknowledgments

We are grateful to two anonymous referees for their constructive comments and suggestions which helped improve the quality of the paper. The authors thank Professor Nabil Absi for making the solutions of Absi et al. (2014) available to us and Professor Claudia Archetti for providing the results of Archetti et al. (2017). Computations were performed on the Calcul Québec infrastructure. This work was partly supported by the Canadian Natural Sciences and Engineering Research Council under grants 2014-03849 and 2014-04959, and by the professorship in operations planning at HEC Montréal. This support is gratefully acknowledged.

References

Absi, N., Archetti, C., Dauzère-Pérès, S. and Feillet, D. (2014). A two-phase iterative heuristic approach for the production routing problem, *Transportation Science* **49**(4): 784–795.

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

Adulyasak, Y., Cordeau, J.-F. and Jans, R. (2014b). Optimization-based adaptive large neighborhood search for the production routing problem, *Transportation Science* **48**(1): 20–45.

Adulyasak, Y., Cordeau, J.-F. and Jans, R. (2015). The production routing problem: A review of formulations and solution algorithms, *Computers & Operations Research* **55**: 141–152.

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

Archetti, C., Bertazzi, L., Hertz, A. and Speranza, M. G. (2012). A hybrid heuristic for an inventory routing problem, *INFORMS Journal on Computing* **24**(1): 101–116.

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

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

Archetti, C., Boland, N. and Speranza, M. (2017). A matheuristic for the multi-vehicle inventory routing problem, *INFORMS Journal on Computing* **29**(3): 377–387.

Armentano, V. A., Shiguemoto, A. and Løkketangen, A. (2011). Tabu search with path relinking for an integrated production–distribution problem, *Computers & Operations Research* **38**(8): 1199–1209.

Avella, P., Boccia, M. and Wolsey, L. A. (2017). Single-period cutting planes for inventory routing problems, *Transportation Science (Articles in Advance)* .

Bae, H., Moon, I. and Yun, W. (2014). Economic lot and supply scheduling problem: a time-varying lot sizes approach, *International Journal of Production Research* **52**(8): 2422–2435.

Bard, J. F. and Nananukul, N. (2009). The integrated production–inventory–distribution–routing problem, *Journal of Scheduling* **12**(3): 257–280.

Bard, J. F. and Nananukul, N. (2010). A branch-and-price algorithm for an integrated production and inventory routing problem, *Computers & Operations Research* **37**(12): 2202–2217.

Berman, O. and Wang, Q. (2006). Inbound logistic planning: minimizing transportation and inventory cost, *Transportation Science* **40**(3): 287–299.

Bertazzi, L., Savelsbergh, M. and Speranza, M. G. (2008). Inventory routing, *in* B. L. Golden, S. Raghavan and E. A. Wasil (eds), *The Vehicle Routing Problem: Latest Advances and New Challenges*, Springer Science & Business Media, pp. 49–72.

Blumenfeld, D. E., Burns, L. D., Daganzo, C. F., Frick, M. C. and Hall, R. W. (1987). Reducing logistics costs at General Motors, *Interfaces* **17**(1): 26–47.

Boudia, M., Louly, M. A. O. and Prins, C. (2005). Combined optimization of production and distribution, *Proceedings of the International Conference on Industrial Engineering and Systems Management, IESM*, Vol. 5.

Boudia, M., Louly, M. A. O. and Prins, C. (2007). A reactive grasp and path relinking for a combined production–distribution problem, *Computers & Operations Research* **34**(11): 3402–3419.

Boudia, M. and Prins, C. (2009). A memetic algorithm with dynamic population management for an integrated production–distribution problem, *European Journal of Operational Research* **195**(3): 703–715.

Chandra, P. (1993). A dynamic distribution model with warehouse and customer replenishment requirements, *Journal of the Operational Research Society* **44**(7): 681–692.

Chandra, P. and Fisher, M. L. (1994). Coordination of production and distribution planning, *European Journal of Operational Research* **72**(3): 503–517.

Chen, Z. and Sarker, B. R. (2014). An integrated optimal inventory lot-sizing and vehicle-routing model for a multisupplier single-assembler system with JIT delivery, *International Journal of Production Research* **52**(17): 5086–5114.

Chuah, K. H. and Yingling, J. C. (2005). Routing for a just-in-time supply pickup and delivery system, *Transportation Science* **39**(3): 328–339.

Coelho, L. C., Cordeau, J.-F. and Laporte, G. (2012). The inventory-routing problem with transshipment, *Computers & Operations Research* **39**(11): 2537–2548.

Coelho, L. C., Cordeau, J.-F. and Laporte, G. (2013). Thirty years of inventory routing, *Transportation Science* **48**(1): 1–19.

Coelho, L. C. and Laporte, G. (2013a). A branch-and-cut algorithm for the multi-product multi-vehicle inventory-routing problem, *International Journal of Production Research* **51**(23-24): 7156–7169.

Coelho, L. C. and Laporte, G. (2013b). The exact solution of several classes of inventory-routing problems, *Computers & Operations Research* **40**(2): 558–565.

Cordeau, J.-F., Gendreau, M. and Laporte, G. (1997). A tabu search heuristic for periodic and multi-depot vehicle routing problems, *Networks* **30**(2): 105–119.

CPU-benchmark (2017). https://www.cpubenchmark.net/. Accessed: 14 July 2017.

Danese, P. (2006). The extended VMI for coordinating the whole supply network, *Journal of Manufacturing Technology Management* **17**(7): 888–907.

Desaulniers, G., Rakke, J. G. and Coelho, L. C. (2015). A branch-price-and-cut algorithm for the inventory-routing problem, *Transportation Science* **50**(3): 1060–1076.

Dong, Z. and Turnquist, M. (2015). Combining service frequency and vehicle routing for managing supplier shipments, *Transportation Research Part E: Logistics and Transportation Review* **79**: 231–243.

Fernie, J. and Sparks, L. (2014). *Logistics and Retail Management: Emerging Issues and New Challenges in the Retail Supply Chain*, Kogan Page Publishers.

Fischetti, M. and Lodi, A. (2003). Local branching, *Mathematical Programming* **98**(1-3): 23–47.

Fischetti, M., Polo, C. and Scantamburlo, M. (2004). A local branching heuristic for mixed-integer programs with 2-level variables, with an application to a telecommunication network design problem, *Networks* **44**(2): 61–72.

Fleischmann, B. and Meyr, H. (2003). Planning hierarchy, modeling and advanced planning systems, *Handbooks in Operations Research and Management Science* **11**: 455–523.

Florian, M., Kemper, J., Sihn, W. and Hellingrath, B. (2011). Concept of transport-oriented scheduling for reduction of inbound logistics traffic in the automotive industries, *CIRP Journal of Manufacturing Science and Technology* **4**(3): 252–257.

Francis, P., Smilowitz, K. and Tzur, M. (2006). The period vehicle routing problem with service choice, *Transportation Science* **40**(4): 439–454.

Fumero, F. and Vercellis, C. (1999). Synchronized development of production, inventory, and distribution schedules, *Transportation Science* **33**(3): 330–340.

Hein, F. and Almeder, C. (2016). Quantitative insights into the integrated supply vehicle routing and production planning problem, *International Journal of Production Economics* **177**: 66–76.

Iori, M., Salazar-González, J.-J. and Vigo, D. (2007). An exact approach for the vehicle routing problem with two-dimensional loading constraints, *Transportation Science* **41**(2): 253–264.

Jiang, Z., Huang, Y. and Wang, J. (2010). Routing for the milk-run pickup system in automobile parts supply, *Proceedings of the 6th CIRP-Sponsored International Conference on Digital Enterprise Technology*, Springer, pp. 1267–1275.

Kuhn, H. and Liske, T. (2011). Simultaneous supply and production planning, *International Journal of Production Research* **49**(13): 3795–3813.

Kuhn, H. and Liske, T. (2014). An exact algorithm for solving the economic lot and supply scheduling problem using a power-of-two policy, *Computers & Operations Research* **51**: 30–40.

Lamsal, K., Jones, P. C. and Thomas, B. W. (2016). Sugarcane harvest logistics in brazil, *Transportation Science* **51**(2): 771 – 789.

Le Blanc, H. M., Cruijssen, F., Fleuren, H. A. and De Koster, M. (2006). Factory gate pricing: An analysis of the Dutch retail distribution, *European Journal of Operational Research* **174**(3): 1950–1967.

Lee, C.-G., Bozer, Y. A. and White III, C. (2003). A heuristic approach and properties of optimal solutions to the dynamic inventory routing problem, *Technical report*, University of Toronto, Ontario, Canada.

Lei, L., Liu, S., Ruszczynski, A. and Park, S. (2006). On the integrated production, inventory, and distribution routing problem, *IIE Transactions* **38**(11): 955–970.

Liske, T. and Kuhn, H. (2009). The economic lot and supply scheduling problem under a power-of-two policy, *Operations Research Proceedings 2008*, Springer Science & Business Media, pp. 215–220.

Lysgaard, J., Letchford, A. N. and Eglese, R. W. (2004). A new branch-and-cut algorithm for the capacitated vehicle routing problem, *Mathematical Programming* **100**(2): 423–445.

Mjirda, A., Jarboui, B., Macedo, R., Hanafi, S. and Mladenović, N. (2014). A two phase variable neighborhood search for the multi-product inventory routing problem, *Computers & Operations Research* **52**: 291–299.

Moin, N. H., Salhi, S. and Aziz, N. (2011). An efficient hybrid genetic algorithm for the multi-product multi-period inventory routing problem, *International Journal of Production Economics* **133**(1): 334–343.

Natarajarathinam, M., Stacey, J. and Sox, C. (2012). Near-optimal heuristics and managerial insights for the storage constrained, inbound inventory routing problem, *International Journal of Physical Distribution & Logistics Management* **42**(2): 152–173.

Ohlmann, J., Fry, M. and Thomas, B. (2007). Route design for lean production systems, *Transportation Science* **42**(2): 352–370.

Patel, D. and Patel, M. (2013). Design and development of an internal milk-run material supply system in automotive industry., *International Journal of Application or Innovation in Engineering & Management (IJAIEM)* **2**(8): 233–235.

Pochet, Y. and Wolsey, L. A. (2006). *Production Planning by Mixed Integer Programming*, Springer Science & Business Media.

Popken, D. A. (1994). An algorithm for the multiattribute, multicommodity flow problem with freight consolidation and inventory costs, *Operations Research* **42**(2): 274–286.

Potter, A., Mason, R. and Lalwani, C. (2007). Analysis of factory gate pricing in the U.K. grocery supply chain, *International Journal of Retail & Distribution Management* **35**(10): 821–834.

Qu, W. W., Bookbinder, J. H. and Iyogun, P. (1999). An integrated inventory–transportation system with modified periodic policy for multiple products, *European Journal of Operational Research* **115**(2): 254–269.

Satoglu, S. and Sahin, I. (2013). Design of a just-in-time periodic material supply system for the assembly lines and an application in electronics industry, *The International Journal of Advanced Manufacturing Technology* **65**(1-4): 319–332.

Sindhuchao, S., Romeijn, H. E., Akçali, E. and Boondiskulchok, R. (2005). An integrated inventory-routing system for multi-item joint replenishment with limited vehicle capacity, *Journal of Global Optimization* **32**(1): 93–118.

Solyalı, O. and Süral, H. (2017). A multi-phase heuristic for the production routing problem, *Computers & Operations Research* **87**: 114–124.

Solyalı, O., Süral, H., Neogy, S., Das, A. and Bapat, R. (2009). A relaxation based solution approach for the inventory control and vehicle routing problem in vendor managed systems, *Modeling, Computation and Optimization, World Scientific, Singapore* pp. 171–189.

Stacey, J., Natarajarathinam, M. and Sox, C. (2007). The storage constrained, inbound inventory routing problem, *International Journal of Physical Distribution & Logistics Management* **37**(6): 484–500.

Toth, P. and Vigo, D. (2002). An overview of vehicle routing problems, *in* P. Toth and D. Vigo (eds), *The Vehicle Routing Problem*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, USA, pp. 1–26.

Vaidyanathan, B., Matson, J., Miller, D. and Matson, J. (1999). A capacitated vehicle routing problem for just-in-time delivery, *IIE Transactions* **31**(11): 1083–1092.

Whiteoak, P. (1994). The realities of quick response in the grocery sector: a supplier viewpoint, *International Journal of Physical Distribution & Logistics Management* **24**(10): 33–39.

Yücel, E., Salman, F. S., Gel, E. S., Örmeci, E. L. and Gel, A. (2013). Optimizing specimen collection for processing in clinical testing laboratories, *European Journal of Operational Research* **227**(3): 503–514.

Appendix A: Overview of Problem Data Sets

We test our algorithm on three different problems: the IRP, the PRP and the ARP. Each problem has its own set of different instances. IRP instances include four data sets. The first set (SV-I1) is provided by Archetti et al. (2007) and contains a total of 160 single-vehicle IRP instances. This data set includes instances with 5 to 50 nodes and 3 periods (100 instances) and instances with 5 to 30 nodes and 6 periods (60 instances). Coelho and Laporte (2013a) and Desaulniers et al. (2015) further adapt the SV-I1 data set and construct new instances (second data set) by dividing the fleet capacity equally between the number of vehicles. They consider $m = 2, 3, 4$ and 5 vehicles for each SV-I1 instance and develop four new multi-vehicle IRP instances (MV-I1). Dividing the vehicle capacity by 5 made two of the instances infeasible. Therefore, instead of 640 they have 638 instances in this set. The third IRP data set (SV-I2) includes bigger single-vehicle instances presented by Archetti et al. (2012). This data set includes 60 instances with 6 periods and 50, 100 and 200 nodes (20 instances for each). Similar to the second data set, Coelho and Laporte (2013b) adapt the SV-IRP instances of the third set and developed the fourth multi-vehicle IRP data set (MV-I2) which includes 240 instance. Therefore, we consider a total of 1,098 instances in four IRP data sets.

PRP instances include six data sets. Archetti et al. (2011) and Boudia et al. (2005) each introduce three data sets. Each set of Archetti et al. (2011) has 480 instances including four classes of randomly generated problem instances. The first set provides single-vehicle PRP instances (SV-A1). The other two sets include multi-vehicle instances (MV-A2 and MV-A3). Sets SV-A1, MV-A2 and MV-A3 have 14, 50, 100 customers, respectively, each with 6 periods. Each of these sets has four classes of instances. The first class includes the normal or standard instances. The second class contains high production unit and setup cost instances. The third class represents the case with high transportation costs (by multiplying the customer coordinates of the first class by a factor of 5). Finally, the fourth class includes instances with no retailer inventory costs. Boudia et al. (2005) present 30 test instances in each of their three sets: sets MV-B1, MV-B2 and MV-B3 which have 50, 100 and 200 customers, respectively, each with 20 periods. Accordingly, the total number of instances in these six PRP data sets is 1,530.

We adapt the ARP instances from the PRP data sets of Archetti et al. (2011). ARP instances include three data sets (SV-C1, MV-C2 and MV-C3) with a total number of 1,440 instances. Consequently, we are solving a total of 4,068 instances of the IRP, PRP and ARP problems. Table 9 provides an overview of the main characteristics of these instances.

Appendix B: Parameter Setting

We chose a random but varied subset of instances from the entire test bed of problems to calibrate the parameters of the algorithm. For the IRP data sets of Archetti et al. (2007) and Archetti et al. (2012), with a total number of 1,098 instances, we randomly chose two instances for each combination of the fleet size (m), period (l) and inventory cost level (h) which resulted in a total of 60 instances: 40 instances from Archetti et al. (2007) and 20 instances from Archetti et al. (2012). From the PRP data sets of Archetti et al. (2011), we randomly selected four instances from each class of instances, resulting in 16 instances from each data set and a total of 48 instances. From the PRP data sets of Boudia et al. (2005), we randomly chose four instances from MV-B1 and MV-B2, resulting in 8 instances. No instances from the MV-B3 data set were chosen since

Table 9 Overview of the benchmark data sets for the IRP, PRP and ARP

Problem	Reference	Set name	Size	l	n	m	d_i	C	L_0	L_i	I_0	I_i	Q
IRP	Archetti et al. (2007)	SV-I1	100	3	5 to 50	1	C	-	UL	C	V	V	C
			60	6	5 to 50	1	C	-	UL	C	V	V	C
		MV-I1	100	3	5 to 50	2	C	-	UL	C	V	V	C
			60	6	5 to 50	2	C	-	UL	C	V	V	C
			100	3	5 to 50	3	C	-	UL	C	V	V	C
			60	6	5 to 50	3	C	-	UL	C	V	V	C
			100	3	5 to 50	4	C	-	UL	C	V	V	C
			60	6	5 to 50	4	C	-	UL	C	V	V	C
			100	3	5 to 50	5	C	-	UL	C	V	V	C
			58	6	5 to 50	5	C	-	UL	C	V	V	C
	Archetti et al. (2012)	SV-I2	60	6	50 to 200	1	C	-	UL	C	V	V	C
		MV-I2	60	6	50 to 200	2	C	-	UL	C	V	V	C
			60	6	50 to 200	3	C	-	UL	C	V	V	C
			60	6	50 to 200	4	C	-	UL	C	V	V	C
			60	6	50 to 200	5	C	-	UL	C	V	V	C
PRP	Archetti et al. (2011)	SV-A1	480	6	14	1	C	UL	UL	C	0	V	C
		MV-A2	480	6	50	UL	C	UL	UL	C	0	V	C
		MV-A3	480	6	100	UL	C	UL	UL	C	0	V	C
	Boudia et al. (2005)	MV-B1	30	20	50	5	V	C	C	C	V	0	C
		MV-B2	30	20	100	9	V	C	C	C	V	0	C
		MV-B3	30	20	200	13	V	C	C	C	V	0	C
ARP	This paper	SV-C1	480	6	14	1	C	UL	UL	C	V	V	C
		MV-C2	480	6	50	UL	C	UL	UL	C	V	V	C
		MV-C3	480	6	100	UL	C	UL	UL	C	V	V	C
		Total	4,068										

Note. C: Constant/Capacitated, MV: Multi-vehicle, SV: Single-vehicle, UL: Unlimited, V: Varying.

they require long computing times. From the ARP data sets, we randomly selected four instances from each class of instances, resulting in 48 instances. Therefore, we perform the parameter setting experiments on 164 instances.

The most important algorithmic parameters to set are the maximum number of iterations for the algorithm, ι^A, and the tabu search iterations coefficient, $\iota^V_{\min}$ and $\iota^V_{\max}$, for the solution of the VRP_t subproblems. The rest of the parameters are the maximum number of local optimum iterations, ι^L, the maximum number of iterations without incumbent solution improvement, ι^N, the number of consecutive iterations for which the same cost update mechanism is applied, ι^U, the maximum number of $\mathcal{M}_z^{\mathcal{R}}$ model iterations, $\iota^{\mathcal{R}}$, the right-hand-side of the LBI_z inequalities, r, the reduction in the aggregate fleet capacity in constraints (21), $1-\lambda_t$, the gap between the solution obtained using the $\mathcal{M}_z$ model and the incumbent solution, g, and the gap between the solution obtained using the $\mathcal{M}_z^{\mathcal{R}}$ model and incumbent solution, $g^{\mathcal{R}}$.

We perform an extensive study on the parameter setting and arrive at the values in Table 10. Then, we design a sensitivity analysis to make sure that the selected values are the right choice for our algorithm. Obviously, when ι^A, $\iota^V_{\min}$ and $\iota^V_{\max}$ increase we obtain better results (see Tables 11, 12 and 13). But since the same parameter setting is used for all the problems and data sets, we have an implicit limit on the number of iterations in order to spend an acceptable computing time compared to other benchmark algorithms. Our observation indicates that the algorithm has an acceptable performance with small changes in ι^s, $\iota^{\mathcal{R}}$, g and $g^{\mathcal{R}}$ while the current setting for these four parameters helps us to reduce the necessary computing time. Also, we noticed that the best $1-\lambda_t$ varies among different IRP and PRP data sets. The last column of Table 10 contains the ranges of the sensitivity analyses on the parameter values.

Table 10 Parameter setting for the algorithm applied to all problems and data sets

Par	Description	Selected value	Other values for the sensitivity analysis
ι^A	Max # of algorithm iterations	200	100, 150, 250, 300
$\iota^V_{\min}$	Minimum tabu search iterations coefficient	100	50, 200
$\iota^V_{\max}$	Maximum tabu search iterations coefficient	500	400, 600
ι^L	Max # of local optimum iterations	80 (0.4*ι^A)	60 (0.3*ι^A), 100 (0.5*ι^A)
ι^N	Max # of non-improving iterations	60 (0.3*ι^A)	40 (0.2*ι^A), 80 (0.4*ι^A)
ι^s	# of iterations before $\mathcal{M}_z^{\mathcal{R}}$ model can be used	5	4, 6
$\iota^{\mathcal{R}}$	Max # of $\mathcal{M}_z^{\mathcal{R}}$ model iterations	10	5, 15
ι^U	# of consecutive iterations to apply each mechanism	7	5, 6, 8, 9
$1-\lambda_t$	Aggregate fleet capacity reduction amount†	$2/n$	$1/n$, $3/n$, $4/n$
g	Gap of $\mathcal{S}$ obtained using $\mathcal{M}_z$ model and $\mathcal{S}^*$	3%	2%, 4%
$g^{\mathcal{R}}$	Gap of $\mathcal{S}$ obtained using $\mathcal{M}_z^{\mathcal{R}}$ model and $\mathcal{S}^*$	0.5%	0.3%, 0.4%, 0.6%

Note. Max: Maximum, Par: Parameter, $\mathcal{S}$: Current solution, $\mathcal{S}^*$: Incumbent solution
† Up to a maximum of 25%

We used the following CPLEX setting for all problems and data sets to solve the $\mathcal{M}_y, \mathcal{M}_z$ and $\mathcal{M}_z^{\mathcal{R}}$ models. We used CPLEX with one thread in all of our experiments. We disable all the CPLEX MIP cuts except *FlowCovers* and *Gomory*. We set the *AdvInd* parameter to zero to prevent CPLEX from spending time to recover the previous iteration's search tree with its built-in heuristic. The rest of the CPLEX settings follow the strategy of getting quality upper bounds faster rather than closing the optimality gap when solving the $\mathcal{M}_y, \mathcal{M}_z$ and $\mathcal{M}_z^{\mathcal{R}}$ models. We set *Dive* to 2 (probing dive for the MIP dive strategy), *OrderType* to 1 (to use decreasing costs for the MIP priority order generation in the search tree), *CoeffReduce* to 1 (to reduce only to integral coefficients when the coefficient reduction is used by CPLEX), *DGradient* to 4 (steepest-edge pricing with unit initial norms for the dual simplex pricing algorithm) and *MIP Emphasis* to 1 (to emphasize feasibility over optimality in the search tree). These allow us to terminate CPLEX sooner and execute more iterations. We set a maximum time limit of 40 seconds for CPLEX when solving the $\mathcal{M}_y, \mathcal{M}_z$ and $\mathcal{M}_z^{\mathcal{R}}$ models.

Appendix C: Subproblems for the PRP and IRP

In this section, we redefine the variables, objective, and constraints of our formulation to match the outbound PRP and IRP models in the literature. In PRP and IRP, the set of nodes $N_s = \{1, ..., n\}$, indexed by $i \in N_s$, represents the customers, $i = 0$ represents the plant and $N = N_s \cup \{0\}$ is the set of all nodes. Let K_i denote the storage capacity and F_{it} represent the inventory of the product (at the end of period t) at customer $i \in N_s$ and at the plant for $i = 0$. Let d_{itl} be the total demand of customer/retailer i from period t to the end of planning period l. The rest of the parameters and variables have a similar definition as in the ARP. The $\mathcal{M}_y$ model for the PRP is defined as follows:

$$\min \sum_{t\in T}\left(up_t + fy_t + \sum_{i\in N} h_i F_{it} + \sigma_{0t} z_{0t}\right) \tag{29}$$

s.t.

$$F_{0,t-1} + p_t = \sum_{i\in N_s} q_{it} + F_{0t} \quad \forall t\in T \tag{30}$$

$$F_{i,t-1} + q_{it} = d_{it} + F_{it} \quad \forall i\in N_s, \forall t\in T \tag{31}$$

$$p_t \le min\{C, \sum_{i\in N_s} d_{itl}\} y_t \quad \forall t\in T \tag{32}$$

$$q_{it} \le min\{K_i, Q, d_{itl}\} z_{it} \quad \forall i\in N_s, \forall t\in T \tag{33}$$

$$\sum_{i\in N_s} q_{it} \le Q z_{0t} \quad \forall t\in T \tag{34}$$

$$z_{0t} \le m \quad \forall t\in T \tag{35}$$

$$F_{it} \le K_i \quad \forall i\in N, \forall t\in T \tag{36}$$

$$p_t \ge 0, y_t \in \{0,1\}, z_{0t}\in \mathbb{Z} \quad \forall t\in T \tag{37}$$

$$F_{it} \ge 0 \quad \forall i\in N \tag{38}$$

$$q_{it} \ge 0, z_{it}\in\{0,1\} \quad \forall i\in N_s, \forall t\in T. \tag{39}$$

The objective function (29) minimizes the total production, setup, and inventory costs both at the plant and customers together with the vehicle dispatch cost. Constraints (30) and (31) ensure the inventory flow at the plant and at the customers, respectively. Constraints (32) and (33) force setup costs at the plant and vehicle visits to the customers, respectively. They also impose limits on production and shipment quantities. Constraints (34) are equivalent to constraints (19) for the ARP. Constraints (35) and (36) enforce the fleet size and storage limits at the plant and customers. The $\mathcal{M}_z$ model for the PRP is to minimize the following objective function:

$$\min \sum_{t\in T}\left(up_t + fy_t + \sum_{i\in N} h_i F_{it} + \sum_{i\in N_s} \sigma_{it} z_{it}\right), \tag{40}$$

subject to constraints (30)-(33), (35)-(39) and (41):

$$\sum_{i\in N_s} q_{it} \le \lambda_t m Q \quad \forall t\in T. \tag{41}$$

Constraints (41) are the equivalent of constraints (21) for the PRP. Having the binary decisions y_t fixed from the solution of the $\mathcal{M}_y$ model, they become constants in constraints (32) for the $\mathcal{M}_z$ model. In the $\mathcal{M}_z$ model for IRP, p_t is a parameter that makes the constraints (32) not applicable. The objective function of the $\mathcal{M}_z$ model will then be:

$$\min \sum_{t\in T}\sum_{i\in N}(h_i F_{it} + \sigma_{it} z_{it}). \tag{42}$$

To comply with the replenishment process timing assumption in Archetti et al. (2007) and Archetti et al. (2011), Adulyasak et al. (2014a) suggested constraints (43) that should be added to the $\mathcal{M}_z$ model for IRP. Constraints (44) are equivalent to the original assumption (Archetti et al. 2007, 2011), $F_{i,t-1} + q_{it} \le K_i$, and can be obtained by replacing the LHS from constraints (31). The reason for this modification is to impose them as bounds on the inventory variables rather than adding constraints to the model.

$$F_{0t} \geq p_t \quad \forall t \in T \tag{43}$$

$$F_{it} \leq K_i - d_{it} \quad \forall t \in T \tag{44}$$

Moreover, the fixed cost $\sum_{i \in N} h_i F_{i0}$ has to be added to the final solution value for the IRP instances, since Archetti et al. (2007) and Archetti et al. (2011) consider the inventory costs at the beginning of the period starting from period zero.

Appendix D: Further Analysis of the Algorithm

In addition to the 200 iterations that we fix for CCJ-DH for all problems and data sets as reported in the main paper, we let it run for $\iota^A = \{100, 150, 250, 300\}$ with different starting node visit costs. Tables 11-14 report the average gap (%), number of BUBs, number of NBS, and computing time (seconds). Moreover, we examined the effect of employing each of the three update mechanisms separately and present in the same tables the results for $\iota^A = \{200, 250\}$ iterations. The results indicate that the mixed mechanism works better than each of the cost update mechanisms separately. The exception is on the Boudia et al. (2005) data sets for which the marginal cost update mechanism outperforms the other mechanisms. CCJ-DJ is successful to find average gaps less than 0 or in other words it outperforms the state-of-the-art algorithm (ABS-H) on the MV-I2 data set in all scenarios. On the MV-A2 data set the algorithm with 100 iterations performs almost the same or better than all the previous benchmark algorithms with different starting node visit costs. The VRP route cost update mechanism leads to substantially bigger average gaps while it still is competitive on the large-scale multi-vehicle MV-I2 instances compared to the previous state-of-the-art heuristics. Overall, different CCJ-DH scenario implementations return 1257 BUBs among which 973 are NBSs on all data sets.

Finally, we present two further sensitivity analyses on the relevant instances to evaluate the effect of the $\mathcal{M}_z^{\mathcal{R}}$ model for the multi-vehicle instances with $m < n$, and the upper bound on the number of necessary vehicles. Table 15 shows the results with and without implementing $\mathcal{M}_z^{\mathcal{R}}$ model. We performed the experiments of these two tables by setting 100 iterations and $\sigma_{it} = 0.5c_{0i}$ for CCJ-DH, similar to the scenario in the sixth column of Tables 11-14. Our observation is that the algorithm without using $\mathcal{M}_z^{\mathcal{R}}$ faces more infeasible VRP_t subproblems for these instances. Therefore, the $\mathcal{M}_z^{\mathcal{R}}$ model implementation is crucial to obtain quality solutions. Moreover, it resulted in better average gaps and more BUBs on all data sets and classes, except for MV-I2 with $m = 3$. Table 16 presents the effect of implementing the valid upper bound on the number of vehicles (Proposition 1) when CCJ-DH is applied on the multi-vehicle IRP and PRP instances. The results show that for the data sets with few vehicles available (MV-I1 and MV-I2) the time saving of applying this bound is negligible. On the instances with an unlimited number of vehicles, the time saving factor is about 3 (for MV-A2 data set with $n = 50$) to 4 (for MV-A3 data set with $n = 100$). The average gaps and number of BUBs are almost the same except for the MV-B3.

Table 11 Average gaps by different cost update mechanisms and initial node visit costs for IRP and PRP data sets (%)

					Mixed M† and T† Mechanisms															M†		T†		V†		
					σ_{it}‡ $= c_{0i}/2$					σ_{it}‡ $= c_{0i}$					σ_{it}‡ $= 2c_{0i}$					σ_{it}‡ $= c_{0i}/2$						
Prob	Set	m	Class	Size	100	150	200	250	300	100	150	200	250	300	100	150	200	250	300	200	250	200	250	200	250	OB††
IRP	SV-I1	1	-	160	1.66	1.61	1.62	1.59	1.57	1.61	1.57	1.57	1.56	**1.53**	1.65	1.6	1.61	1.58	1.56	3.22	2.92	2.08	2	5.3	5.16	1.39
	MV-I1	2	-	160	2.81	2.68	2.6	2.6	2.52	2.66	2.54	2.46	2.51	**2.43**	2.73	2.68	2.59	2.61	2.5	4.06	3.61	2.82	2.76	6.26	6.05	1.8
		3	-	160	2.54	2.51	2.4	2.35	2.24	2.44	2.44	2.36	2.3	**2.19**	2.49	2.45	2.42	2.33	2.21	3.73	3.62	2.77	2.68	6.97	6.49	1.67
		4	-	160	2.89	2.71	2.55	2.51	2.51	2.88	2.71	2.56	2.52	2.47	2.89	2.72	2.5	2.48	**2.39**	4.45	4.14	2.83	2.83	7.05	6.85	1.87
		5	-	158	3.01	2.91	2.75	2.68	2.56	2.89	2.8	2.72	2.58	**2.49**	2.99	2.92	2.73	2.65	2.6	4.66	4.57	3.01	3.01	7.58	7.48	2.05
	SV-I2	1	-	60	3.71	3.55	3.51	3.46	3.39	3.67	3.58	3.49	**3.38**	**3.38**	3.61	3.57	3.49	3.4	3.43	4.82	4.54	4.79	4.74	4.86	4.78	2.9
	MV-I2	2	-	60	-1.59	-1.68	-1.82	**-2.05**	-1.94	-1.6	-1.69	-1.74	-1.83	-1.97	-1.52	-1.78	-1.88	-1.75	-1.95	-0.61	-0.8	-0.68	-0.82	-0.11	-0.22	-2.67
		3	-	60	-2.55	-2.72	-2.86	-2.95	-3	-2.64	-2.91	-2.8	**-3.04**	-2.93	-2.24	-2.67	-2.91	-3	-2.99	-2.56	-2.75	-1.77	-1.83	-1.29	-1.5	-3.92
		4	-	60	-4.48	-4.57	-4.73	-4.73	**-4.85**	-4.45	-4.7	-4.77	-4.81	-4.78	-4.41	-4.52	-4.6	-4.74	-4.74	-3.82	-3.9	-3.69	-3.84	-3.12	-3.07	-5.63
		5	-	60	-4.65	-4.69	-4.9	-4.85	-4.86	-4.56	-4.46	-4.64	-4.7	-4.77	-4.42	-4.76	-4.84	-4.98	**-5.04**	-4.2	-4.26	-4.4	-4.5	-3.05	-3.18	-5.78
PRP	SV-A1	1	1	120	0.35	0.28	0.24	0.2	**0.15**	0.34	0.29	0.25	0.22	0.17	0.35	0.29	0.26	0.22	0.16	0.39	0.38	0.5	0.49	0.95	0.96	0.11
		1	2	120	0.06	0.04	0.03	**0.03**	**0.03**	0.06	0.04	0.04	0.03	**0.03**	0.06	0.04	0.04	0.03	**0.03**	0.07	0.06	0.08	0.08	0.16	0.15	0.01
		1	3	120	1.82	1.61	1.49	1.3	1.21	1.87	1.69	1.52	1.29	1.2	1.84	1.69	1.48	1.25	**1.17**	2.03	1.93	2.41	2.32	4.66	4.22	0.61
		1	4	120	0.18	0.15	0.13	0.14	**0.08**	0.18	0.16	0.14	0.15	0.09	0.18	0.15	0.14	0.14	**0.08**	0.24	0.2	0.31	0.3	0.52	0.5	0.05
	MV-A2	UL	1	120	-0.05	-0.05	-0.06	-0.05	**-0.06**	-0.05	-0.05	-0.05	-0.06	**-0.06**	-0.04	-0.04	-0.05	-0.05	-0.05	0.04	0.03	0.1	0.1	0.32	0.31	-0.09
		UL	2	120	0.02	0.02	**0.01**	**0.01**	**0.01**	0.02	**0.02**	**0.02**	**0.01**	**0.01**	0.03	0.02	**0.02**	**0.01**	**0.01**	0.02	0.02	0.06	0.06	0.11	0.11	-0.01
		UL	3	120	-0.1	-0.17	-0.2	-0.24	**-0.3**	-0.17	-0.22	-0.25	-0.28	-0.29	-0.11	-0.2	-0.26	-0.27	-0.28	0.27	0.19	-0.06	-0.11	1.11	1	-0.41
		UL	4	120	-0.02	-0.02	-0.03	-0.03	**-0.04**	-0.02	-0.02	-0.03	-0.03	**-0.04**	-0.01	-0.03	-0.02	-0.03	-0.03	-0.03	-0.03	0.02	0.01	0.11	0.11	-0.07
	MV-A3	UL	1	120	0.21	0.2	0.18	0.17	0.18	0.19	0.19	0.17	0.17	**0.17**	0.19	0.18	0.18	0.17	**0.17**	0.23	0.23	0.53	0.5	0.58	0.61	0.13
		UL	2	120	0.18	0.18	**0.18**	**0.18**	**0.18**	0.18	**0.18**	**0.18**	**0.18**	**0.18**	0.18	0.18	**0.18**	**0.18**	**0.18**	0.19	0.18	0.21	0.21	0.23	0.23	0.16
		UL	3	120	1.15	1.11	1.07	1.02	**0.96**	1.26	1.26	1.11	1.02	1.02	1.23	1.14	1.09	1.03	1.02	1.46	1.47	1.64	1.6	2.26	2.15	0.79
		UL	4	120	0.11	0.11	0.11	0.1	0.1	0.11	0.11	0.11	0.1	**0.1**	0.11	0.11	0.11	0.1	**0.1**	0.1	**0.1**	0.13	0.13	0.17	0.16	0.08
	MV-B1	5	-	30	0.89	0.8	0.78	0.77	0.74	0.96	0.92	0.89	0.84	0.83	1.46	1.26	1.34	1.31	1.28	0.74	**0.71**	1.78	1.76	2.07	2.1	0.5
	MV-B2	9	-	30	1.28	1.22	1.21	1.14	1.12	1.33	1.33	1.32	1.24	1.19	1.46	1.31	1.28	1.22	1.21	1.08	**1.04**	1.93	1.94	2.32	2.28	0.85
	MV-B3	13	-	30	0.22	0.21	0.18	0.16	0.17	0.27	0.22	0.2	0.19	0.2	0.26	0.19	0.17	0.18	0.15	-0.04	**-0.05**	1.91	1.91	2.19	2.13	-0.1

Note. The best average gap at each row is presented with the bold font.
† Transportation cost update mechanisms. M: Marginal, T: TSP route cost share, V: VRP route cost share. ‡ Initial values for node visit cost.
†† Overall best obtained solution for each instance is taken into account.

Table 12 Number of BUBs by different cost update mechanisms and initial node visit costs for IRP and PRP data sets

					Mixed M† and T† Mechanisms															M†		T†		V†		
					σ_{it}‡ $= c_{0i}/2$					σ_{it}‡ $= c_{0i}$					σ_{it}‡ $= 2c_{0i}$					σ_{it}‡ $= c_{0i}/2$						
Prob	Set	m	Class	Size	100	150	200	250	300	100	150	200	250	300	100	150	200	250	300	200	250	200	250	200	250	OB††
IRP	SV-I1	1	-	160	30	**33**	31	31	31	32	**33**	**33**	30	**33**	30	**33**	32	32	**33**	30	32	28	28	14	16	39
	MV-I1	2	-	160	24	27	**28**	**28**	**28**	23	26	27	27	27	25	26	26	26	**28**	16	16	20	21	10	11	35
		3	-	160	31	32	31	31	34	32	32	33	32	34	32	31	31	31	**36**	22	25	23	24	6	7	42
		4	-	160	27	29	29	33	31	26	28	29	32	30	28	27	31	**34**	**34**	15	18	26	25	9	9	39
		5	-	158	33	34	38	38	38	33	36	36	**39**	**39**	34	33	37	38	37	24	26	30	29	11	11	48
	SV-I2	1	-	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MV-I2	2	-	60	40	40	40	40	40	40	40	40	**41**	**41**	40	40	40	**41**	**41**	35	39	38	37	36	36	43
		3	-	60	44	44	45	46	48	46	46	46	**49**	48	44	45	45	45	46	46	46	43	43	41	40	51
		4	-	60	50	49	51	50	50	52	53	**54**	**54**	53	48	49	52	51	50	48	47	48	48	44	45	58
		5	-	60	57	57	**58**	57	**58**	54	54	55	55	56	54	55	57	56	56	52	52	57	56	49	46	60
PRP	SV-A1	1	1	120	13	12	19	20	23	14	11	19	19	21	17	13	19	21	**25**	12	16	3	3	1	1	39
		1	2	120	14	17	20	**23**	20	14	15	19	21	22	14	15	17	21	19	12	15	3	7	1	1	46
		1	3	120	3	3	**5**	4	3	2	3	4	4	3	2	3	4	4	3	**5**	**5**	2	2	0	0	10
		1	4	120	26	32	32	38	39	28	33	38	35	**41**	28	31	34	39	**41**	29	33	8	8	2	2	65
	MV-A2	UL	1	120	101	100	101	100	**102**	98	100	98	100	**102**	97	99	96	97	98	81	81	84	84	57	57	110
		UL	2	120	39	41	47	49	**53**	38	43	45	44	46	32	40	44	44	47	44	47	25	25	7	9	79
		UL	3	120	87	91	92	96	**98**	90	91	94	96	95	89	93	97	97	**98**	63	70	82	87	37	43	102
		UL	4	120	71	76	75	**80**	79	69	76	75	**80**	78	69	77	76	**80**	78	71	73	38	42	24	26	94
	MV-A3	UL	1	120	73	74	**79**	78	78	74	76	**79**	**79**	78	73	77	78	77	**79**	67	66	42	43	25	28	82
		UL	2	120	39	36	46	48	50	39	40	47	46	**53**	32	36	44	45	50	45	42	22	21	5	6	82
		UL	3	120	26	26	28	**30**	**30**	21	19	25	26	28	23	26	26	**30**	**30**	25	23	25	26	11	11	35
		UL	4	120	47	46	46	53	52	46	46	47	51	53	45	48	48	53	**57**	53	56	28	26	16	17	71
	MV-B1	5	-	30	1	1	2	2	**3**	2	2	**3**	**3**	**3**	0	2	2	2	**3**	2	**3**	0	0	0	0	5
	MV-B2	9	-	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0	1
	MV-B3	13	-	30	12	12	12	14	15	9	11	12	11	10	10	12	14	11	12	18	**19**	0	0	0	0	21
Total (All Instances)				2628	888	912	955	989	1003	882	914	958	974	994	866	911	950	975	1001	815	851	675	685	406	422	1257
Total (LSMV‡† Instances)				1290	687	693	722	743	756	678	697	720	735	744	656	699	719	729	745	650	665	532	538	352	364	894

Note. The largest number of best solutions obtained at each row is presented with the bold font.
† Transportation cost update mechanisms. M: Marginal, T: TSP route cost share, V: VRP route cost share. ‡ Initial values for node visit cost.
†† Overall best obtained solution for each instance is taken into account. ‡† Large-scale multi-vehicle.

Table 13 Number of new best solutions found by different cost update mechanisms and initial node visit costs for IRP and PRP data sets

					Mixed M† and T† Mechanisms															M†		T†		V†		
					$\sigma_{it}{}^{\ddagger} = c_{0i}/2$					$\sigma_{it}{}^{\ddagger} = c_{0i}$					$\sigma_{it}{}^{\ddagger} = 2c_{0i}$					$\sigma_{it}{}^{\ddagger} = c_{0i}/2$						
Prob	Set	m	Class	Size	100	150	200	250	300	100	150	200	250	300	100	150	200	250	300	200	250	200	250	200	250	OB††
IRP	SV-I1	1	-	160	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MV-I1	2	-	160	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		3	-	160	14	13	13	14	14	**15**	13	**15**	**15**	14	**15**	12	13	13	**15**	11	11	10	11	3	3	19
		4	-	160	21	21	22	21	21	20	20	22	20	20	21	21	22	**23**	**23**	11	14	18	18	6	6	26
		5	-	158	27	27	31	32	31	27	29	29	**33**	32	28	26	30	32	30	20	21	26	26	10	10	39
	SV-I2	1	-	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MV-I2	2	-	60	40	40	40	40	40	40	40	40	**41**	**41**	40	40	40	**41**	**41**	35	39	38	37	36	36	43
		3	-	60	44	44	45	46	48	46	46	46	**49**	48	44	45	45	45	46	46	46	43	43	41	40	51
		4	-	60	50	49	51	50	50	52	53	**54**	**54**	53	48	49	52	51	50	48	47	47	48	44	45	58
		5	-	60	57	57	**58**	57	**58**	54	54	55	55	56	54	55	57	56	56	52	52	57	56	49	46	60
PRP	SV-A1	1	1	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1	2	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1	3	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		1	4	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	MV-A2	UL	1	120	98	97	**99**	97	**99**	95	97	96	98	**99**	95	97	95	96	97	80	80	81	80	52	55	108
		UL	2	120	39	41	46	48	**53**	38	43	44	44	46	32	40	44	44	47	44	47	25	25	7	9	79
		UL	3	120	83	88	89	92	**95**	86	88	92	94	92	86	91	94	94	**95**	63	70	80	85	35	40	100
		UL	4	120	71	76	74	78	**79**	69	76	74	**79**	78	69	77	75	**79**	78	71	72	37	41	24	25	93
	MV-A3	UL	1	120	73	73	**79**	78	78	74	76	**79**	**79**	78	73	77	78	77	**79**	67	66	42	42	25	27	82
		UL	2	120	39	36	46	48	50	39	40	47	46	**53**	31	35	44	44	49	45	42	22	21	5	6	82
		UL	3	120	26	26	28	**30**	**30**	21	19	25	26	28	23	26	26	**30**	**30**	25	23	24	25	11	11	35
		UL	4	120	46	46	46	52	52	45	46	47	50	53	44	48	48	52	**57**	53	55	27	25	15	16	71
	MV-B1	5	-	30	1	1	2	2	**3**	2	2	**3**	**3**	**3**	0	2	2	2	**3**	2	**3**	0	0	0	0	5
	MV-B2	9	-	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**1**	0	0	0	0	1
	MV-B3	13	-	30	12	12	12	14	15	9	11	12	11	10	10	12	14	11	12	18	**19**	0	0	0	0	21
Total (All Instances)				2628	741	747	781	799	816	732	753	780	797	804	713	753	779	790	808	691	708	577	583	363	375	973
Total (LSMV‡‡ Instances)				1290	679	686	715	732	750	670	691	714	729	738	649	694	714	722	740	649	662	523	528	344	356	889

Note. The largest number of best solutions obtained at each row is presented with the bold font.
† Transportation cost update mechanisms. M: Marginal, T: TSP route cost share, V: VRP route cost share. ‡ Initial values for node visit cost.
†† Overall best obtained solution for each instance is taken into account. ‡‡ Large-scale multi-vehicle.

Table 14 Average running time for different cost update mechanisms and initial node visit costs for IRP and PRP data sets (seconds)

					Mixed M† and T† Mechanisms															M†		T†		V†	
					$\sigma_{it}{}^{\ddagger} = c_{0i}/2$					$\sigma_{it}{}^{\ddagger} = c_{0i}$					$\sigma_{it}{}^{\ddagger} = 2c_{0i}$					$\sigma_{it}{}^{\ddagger} = c_{0i}/2$					
Prob	Set	m	Class	Size	100	150	200	250	300	100	150	200	250	300	100	150	200	250	300	200	250	200	250	200	250
IRP	SV-I1	1	-	160	24	36	48	60	72	25	36	51	60	79	27	36	48	60	73	44	56	55	71	51	65
	MV-I1	2	-	160	39	52	68	85	99	37	60	69	88	118	36	52	68	84	98	77	95	69	92	67	77
		3	-	160	37	52	69	108	125	37	52	67	83	104	38	52	72	84	100	75	94	69	95	54	78
		4	-	160	37	53	74	84	99	38	61	77	86	102	38	54	71	94	115	78	88	68	86	45	70
		5	-	158	36	52	65	84	98	40	54	66	91	99	37	49	68	85	94	64	79	65	80	58	68
	SV-I2	1	-	60	3363	4998	6668	8295	9209	3361	5019	6831	8375	9259	3423	5061	6725	8300	9397	4394	5436	7795	9472	7617	9169
	MV-I2	2	-	60	2781	4122	5657	6914	8029	2889	4116	5351	6494	8172	3011	4718	6045	6860	8397	4617	5890	5943	7492	6473	7957
		3	-	60	2120	3513	4209	5386	6624	2088	3272	4397	5325	6923	2144	3219	4450	5478	6401	4441	6008	4159	5204	5354	6683
		4	-	60	2780	4053	5132	6236	7046	3102	4126	5289	6371	7246	2916	4199	5333	6502	7222	5918	7204	5593	6668	6880	8160
		5	-	60	3232	4319	5527	6290	7101	2983	3977	5167	6213	6952	2911	4122	5113	6211	7163	6249	7745	4733	5807	6925	8119
PRP	SV-A1	1	1	120	8	13	18	24	30	8	13	18	24	30	8	13	18	24	30	17	22	25	33	13	17
		1	2	120	8	13	18	23	29	8	13	18	23	29	8	13	18	23	29	18	22	23	30	12	15
		1	3	120	8	12	17	22	28	8	12	17	22	28	8	14	17	22	28	14	18	25	33	9	12
		1	4	120	8	13	18	24	30	8	13	18	23	29	8	13	18	23	29	15	19	25	33	15	20
	MV-A2	UL	1	120	201	300	400	497	600	201	299	400	500	598	202	301	434	503	605	522	714	365	460	301	378
		UL	2	120	170	258	344	432	524	170	258	347	433	524	169	257	355	431	662	388	525	331	422	257	323
		UL	3	120	155	233	309	385	465	157	234	313	385	467	158	236	356	394	473	353	436	288	358	176	200
		UL	4	120	215	326	434	540	661	217	329	436	525	660	216	325	470	541	654	468	669	426	656	394	448
	MV-A3	UL	1	120	1103	1612	2125	2644	3162	1084	1636	2131	2642	3174	1080	1886	2124	2637	3458	2933	3495	1720	2204	1848	1965
		UL	2	120	961	1454	1947	2432	2920	962	1461	1946	2422	2923	962	1453	1938	2431	3323	2134	2614	1791	2248	1583	1958
		UL	3	120	729	1088	1461	1802	2173	730	1093	1456	1809	2165	757	1224	1473	1846	2681	1767	2171	1485	1883	916	1147
		UL	4	120	1112	1664	2213	2767	3317	1110	1663	2214	2753	3285	1106	1661	2223	3225	3597	2293	2844	2150	2703	1985	2463
	MV-B1	5	-	30	1825	2622	3559	4446	5326	1840	2642	3546	4480	5361	1729	2494	3408	4319	5090	1347	1698	6760	8447	8602	10581
	MV-B2	9	-	30	5368	7460	9811	11773	13911	5888	8203	10876	12812	15280	5619	7925	9894	12401	14623	9488	12849	13256	16648	14130	17406
	MV-B3	13	-	30	8344	12044	15891	19329	22858	8294	11878	15747	19310	23012	8191	11705	15621	18950	22520	17955	22625	22340	27584	23869	28820

† Transportation cost update mechanisms. M: Marginal, T: TSP route cost share, V: VRP route cost share.
‡ Initial values for node visit cost.

Table 15 **Effect of implementing $\mathcal{M}_z^{\mathcal{R}}$ model in CCJ-DH on relevant IRP and PRP instances**

					Gap (%)		# BUB		# NBS		Time (sec)	
Prob	Set	m	Class	Size	$\mathcal{M}_z^{\mathcal{R}}$	NM†	$\mathcal{M}_z^{\mathcal{R}}$	NM†	$\mathcal{M}_z^{\mathcal{R}}$	NM†	$\mathcal{M}_z^{\mathcal{R}}$	NM†
IRP	MV-I1	2	-	160	**2.81**	3.62	**24**	8	0	0	39	**37**
		3	-	160	**2.54**	3.42	**31**	6	**14**	3	**37**	38
		4	-	160	**2.89**	3.42	**27**	14	**21**	11	**37**	38
		5	-	158	**3.01**	3.39	**33**	18	**27**	16	**36**	**36**
	MV-I2	2	-	60	**-1.59**	-1.56	**40**	**40**	**40**	**40**	2781	**2698**
		3	-	60	**-2.55**	-2.39	44	**45**	44	**45**	2120	**1842**
		4	-	60	**-4.48**	-4.13	**50**	44	**50**	44	2780	**1761**
		5	-	60	**-4.65**	-4.54	**57**	54	**57**	54	3232	**1917**
PRP	MV-B1	5	-	30	**0.89**	2.51	**1**	0	**1**	0	1825	**1778**
	MV-B2	9	-	30	**1.28**	2.15	0	0	0	0	5368	**4318**
	MV-B3	13	-	30	**0.22**	0.85	**12**	1	**12**	1	**8344**	9171

Note. Other data sets include only single-vehicle or unlimited multi-vehicle instances.
† Without implementing $\mathcal{M}_z^{\mathcal{R}}$ model.

Table 16 **Effect of valid upper bound on the number of vehicles on the algorithm's performance when applied to multi-vehicle IRP and PRP instances**

					Gap (%)		# BUB		# NBS		Time (sec)	
Prob	Set	m	Class	Size	TB†	NB‡	TB†	NB‡	TB†	NB‡	TB†	NB‡
IRP	MV-I1	2	-	160	**2.81**	2.89	**24**	22	0	0	39	**38**
		3	-	160	2.54	**2.53**	31	**33**	**14**	13	**37**	**37**
		4	-	160	**2.89**	2.95	**27**	26	**21**	20	**37**	39
		5	-	158	**3.01**	3.09	**33**	29	**27**	25	**36**	36
	MV-I2	2	-	60	**-1.59**	-1.5	**40**	39	**40**	39	2781	**2761**
		3	-	60	**-2.55**	-2.53	**44**	**44**	**44**	**44**	**2120**	2282
		4	-	60	-4.48	**-4.49**	**50**	49	**50**	49	**2780**	2976
		5	-	60	**-4.65**	-4.41	**57**	55	**57**	55	**3232**	3233
PRP	MV-A2	UL	1	120	**-0.05**	-0.03	**101**	94	**98**	93	**201**	643
		UL	2	120	**0.02**	**0.02**	**39**	33	**39**	33	**170**	562
		UL	3	120	**-0.1**	**-0.1**	87	**90**	83	**87**	**155**	480
		UL	4	120	**-0.02**	**-0.02**	71	**77**	71	**77**	**215**	655
	MV-A3	UL	1	120	0.21	**0.18**	73	**76**	73	**76**	**1103**	5201
		UL	2	120	**0.18**	**0.18**	39	**44**	39	**44**	**961**	4441
		UL	3	120	**1.15**	1.18	**26**	**26**	**26**	**26**	**729**	3333
		UL	4	120	**0.11**	**0.11**	47	**52**	46	**52**	**1112**	4518
	MV-B1	5	-	30	0.89	**0.86**	**1**	**1**	**1**	**1**	**1825**	1980
	MV-B2	9	-	30	1.28	**1.26**	0	0	0	0	**5368**	6911
	MV-B3	13	-	30	0.22	**0.05**	12	**18**	12	**18**	**8344**	15750

Note. Other data sets include only single-vehicle instances and no tighter bound is applicable.
† Tight bound (Eq. 23) is applied.
‡ No bound is applied.