Submitted to *INFORMS Journal on Computing*
manuscript (Please, provide the mansucript number!)

Formulations and Branch-and-Cut Algorithms for Multi-Vehicle Production and Inventory Routing Problems

Yossiri Adulyasak
HEC Montréal and CIRRELT, 3000 chemin de la Côte-Sainte-Catherine, Montréal, H3T 2A7, Canada, yossiri.adulyasak@hec.ca

Jean-François Cordeau
HEC Montréal and CIRRELT, 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

The inventory routing problem (IRP) and the production routing problem (PRP) are two difficult problems arising in the planning of integrated supply chains. These problems are solved in an attempt to jointly optimize production, inventory, distribution and routing decisions. Although several studies have proposed exact algorithms to solve the problems, the multi-vehicle aspect is often neglected due to its complexity. We introduce multi-vehicle PRP formulations, with and without vehicle index, to solve the problem. Several valid inequalities are proposed to strengthen the formulations. The vehicle index formulations are further improved using symmetry breaking constraints, while the non-vehicle index formulation is strengthened by several cuts. We futher consider the problem under the well-known maximum level and order-up-to level inventory replenishment systems. We develop branch-and-cut algorithms for the different formulations and extensive computational results are presented to show the effects of the inequalities and the formulations.

Key words: Integrated supply chain planning; inventory routing; production routing; multi-vehicle; branch-and-cut
History:

1. Introduction.

In a typical retail supply chain which consists of sequential activities of production, storage and distribution, each individual process is often planned and optimized using predetermined decisions from its previous activity. For example, a production planner makes production lot-sizing decisions in order to minimize production and inventory costs at the production facility. The planned lot-sizing decisions are used as inputs in subsequent steps in distribution planning. Since the decisions are limited by the plan of the former process, benefits of coordination in the planning process have been left behind. This induced the need for developing integrated supply chain operational planning systems to capture the benefits and optimize the whole system.

Operation research techniques are seen as an effective tool in optimizing supply chain operational planning decisions. Many studies have proposed integrated models to jointly optimize decisions of subsequent processes. The inventory routing problem (IRP) is one of these integrated problems that has received much attention during the past decade (see Andersson et al. (2010) for a review). This problem considers the integration of replenishment and routing plans in order to optimize total inventory and routing costs. When production lot-sizing decisions are incorporated, the problem becomes the production routing problem (PRP) and it is a generalization of the IRP (Ruokokoski et al. 2010, Adulyasak et al. 2011).

We consider the IRP and PRP with a discrete time finite horizon in this study. The IRP first appeared in the gas delivery study by Bell et al. (1983). The problem is solved using a Lagrangian relaxation method and decomposed by time period and by vehicle. Christiansen (1999) introduced an IRP application in a maritime context, called inventory pickup and delivery problem, and applied a Dantzig-Wolfe decomposition and column generation approach to solve the problem. Carter et al. (1996) and Campbell and Savelsbergh (2004) proposed efficient heuristic procedures by decomposing the IRP into the allocation problem (AP) and the vehicle routing problem (VRP). Since the IRP is a complicated combinatorial problem, several metaheuristics, e.g., tabu search (Rusdiansyah and Tsao 2005), genetic algorithm (Abdelmaguid and Dessouky 2006), greedy randomized adaptive search procedure (GRASP) (Savelsbergh and Song 2007), hybrid heuristic with combined tabu search and MIPs (Archetti et al. 2011a), and adaptive large neighborhood search (ALNS) (Coelho et al. 2012) have been proposed. Gaur and Fisher (2004) discussed a periodic IRP where the demand pattern is repeated and developed a heuristic to solve the problem.

As mentioned in Andersson et al. 2010, only few exact algorithms are proposed to solve the IRP due to its complexity. We summarize here the literature concerning the exact methods in a retailer supply chain. To represent instance sizes, we use the term $\bar{a}$c/$\bar{b}$p/$\bar{c}$v where $\bar{a}$, $\bar{b}$ and $\bar{c}$ are the number of customers, periods and vehicles, respectively (e.g., 10c/5p/2v represents an instance with 10 customers, 5 periods and 2 vehicles). Archetti et al. (2007) developed a branch-and-cut approach for the IRP with a single vehicle and analyzed three different replenishment policies for the customers. In the first policy, called order-up-to level (OU), a visited customer receives exactly the amount wjocj brings its inventory up to a predefined target stock level (TSL). The second policy, called maximum level (ML), allows delivery quantities to be any positive value but the inventory at each customer cannot exceed its maximum stock level. The third policy is similar to the ML policy but there is no maximum stock level imposed at the customers. Archetti et al. (2007) used different inequalities to strengthen the formulation for each policy and could solve instances up to 45c/3p/1v and 30c/6p/1v to optimality within two hours for the IRP with OU and ML policies. Solyalı and Süral (2011) proposed a stronger formulation for the single vehicle IRP-OU using a shortest-path network representation of the OU policy at each customer and used a similar branch-and-cut approach as presented in Archetti et al. (2007). Solyalı and Süral (2011) could solve instances up to 60c/3p/1v and 15c/12p/1v to optimality within two hours. Savelsbergh and Song (2008) considered a variant of the IRP, called IRP with continuous move, where a product is distributed from a set of plants to a set of customers by multiple vehicles. In this study, minimum delivery quantities are imposed and inventory costs are disregarded. The authors proposed a multi-commodity flow with vehicle index formulation and developed a branch-and-cut approach to solve the problem.

A closely related problem, the PRP, has also received more attention in recent years. The benefits of coordination in the PRP were first discussed in Chandra (1993) and Chandra and Fisher (1994). Similar to the IRP, most of the previous studies employed heuristic procedures to solve the problem. Several metaheuristics, such as, GRASP (Boudia et al. 2007), memetic algorithm (Boudia and Prins 2009), tabu search (Bard and Nananukul 2009, Armentano et al. 2011), and ALNS (Adulyasak et al. 2011), have been employed to solve the PRP. Archetti et al. (2011b) discussed the PRP under the ML and OU policies and developed an integer linear programming (ILP) heuristic to solve the problem.

Few studies have introduced exact algorithms or methods to compute strong lower bound for the PRP. Fumero and Vercellis (1999) developed a Lagrangian relaxation approach to obtain lower bounds and heuristic solutions for a variant of the PRP where unit transportation costs are assumed and the routing decisions can be determined using the minimum cost flow problem. A similar formulation was used by Solyalı and Süral (2009) to solve the PRP-OU , but only instances with 8c/5p/1v were solved to optimality where the longest computing time was approximately 20 hours. Ruokokoski et al. (2010) explored the performance of different lot-sizing reformulation schemes for the PRP-ML with uncapacitated production and an uncapacitated single vehicle, and further employed a branch-and-cut approach similar to Archetti et al. (2007) to solve the problem. Bard and Nananukul (2010) introduced a branch-and-price procedure for the PRP-ML with multi-vehicles. However, since the subtour elimination constraints are in the form of the Miller-Tucker-Zemlin constraints (Miller et al. 1960), it led to a poor lower bound quality and only the instances up to 10c/2p/5v are solved to optimality within 30 minutes. The emphasis of the study was instead put on a heuristic procedure using the branch-and-price framework. Archetti et al. (2011b) adopted the branch-and-cut approach as presented in Archetti et al. (2007) for the PRP-ML with uncapacitated production and a single vehicle. Several valid inequalities are also used to strengthen the formulation. However, computational testing was only performed on 14c/6p/1v instances and not all instances were solved to optimality within two hours. Table 1 presents a summary of the exact algorithms for the PRP and IRP in literature. We classify the problems along three dimensions: IRP versus PRP, the replenishment policy (ML versus OU) and the number of vehicles (single versus multiple). It clearly shows an important gap in the previous research. No exact algorithms have been proposed and tested for the multiple vehicle IRP and PRP, except for Bard and Nananukul (2010) where only relatively small instances were solved to optimality compared to the results on the single vehicle case.

Table 1 Summary of exact algorithms for the deterministic PRP and IRP with single product, single plant and multiple retailers

	Maximum Level (ML)		**Order-Up-To Level (OU)**	
Problem	**Single vehicle**	**Multiple vehicles**	**Single vehicle**	**Multiple vehicles**
IRP	Archetti et al. (2007) [$45c/3p/1v$]	-	Archetti et al. (2007) Solyalı and Süral (2011) [$60c/3p/1v$], [$15c/12p/1v$]	-
PRP	Ruokokoski et al. (2010)† Archetti et al. (2011b) [$14c/6p/1v$]‡	Bard and Nananukul (2010) [$10c/2p/5v$]	Solyalı and Süral (2009) [$8c/5p/1v$]	-

† the tests were performed only on the uncapacitated single vehicle case
‡ some instances not solved to optimality

In this paper, we consider a single product and a PRP network that consists of a production plant and multiple customers which have their own storage area. At the begining of the planing horizon, the production plant and the customers may have initial inventory. In each period, each customer must have sufficient inventory to satisfy its demand. In the case of the PRP, the plant must decide whether or not to produce the product and the quantities to be produced. If production takes place, fixed set up and unit production costs are incurred. The produced quantities can be transported by a limited number of capacitated vehicles to the customers and routing costs are charged. The product can also be stored at the plant or the customers and unit inventory holding costs are incurred. We consider the cases where the customer replenishment part is controlled by the ML and OU policies. The replenishment practice is generally in lined with Bard and Nananukul (2010) and Archetti et al. (2011b) except for the OU policy where we impose the delivery quantity for each customer by the difference between its current stock level and its TSL before demand consumption as it is clearly aligned with the concept of the OU policy, while the inventory level is imposed after demand comsumption (which typically not known in advance in practice) in Archetti et al. (2011b). It should also be noted that the replenishment practice in our problem and the IRP presented in Bertazzi et al. (2002), Archetti et al. (2007) and Solyalı and Süral (2011) are slightly different. In latter studies, the delivery to the customers must take place before the distribution facility is replenished in each period, while in our PRP, the quantity produced in period t can be delivered to customers to satisfy their demand in the same period. These two practices, however, can be converted to the other as we have shown in the Appendix. In the remainder of this paper, since the PRP is a generalization of the IRP, we prefer to use the name PRP to represent both IRP and PRP unless stated otherwise. Note that the name MVPRP isused to represent the PRP with multi-vehicle (MV) aspect.

The main contributions of our study are fourfold. First, we present strong formulations and exact algorithms for the IRP and PRP with multiple vehicles, which has received little attention in the previous literature. Several formulations are presented and branch-and-cut algorithms are proposed to solve the problem. Second, we propose several valid inequalities and symmetry breaking constraints to strengthen the formulations, and test the effect of these inequalities. Third, we adapt a previously developed ALNS procedure for the MVPRP-ML (Adulyasak et al. 2011) and extend it to the MVPRP-OU, MVIRP-ML and MVIRP-OU. Forth, we provide extensive computational results of the new formulation compared to the branch-and-cut approaches on single vehicle instances in literature.

The rest of this paper is organized as follows. Section 2 presents different formulations of the MVPRP. Section 3 describes the valid inequalities that are applied to the formulations. The details of the branch-and-cut approaches are discussed in Section 4 and the details of the heuristic algorithm to calculate upper bounds are presented in Section 5. This is followed by the discussion of computational experiments in Section 6, and by the conclusion.

2. MVPRP Formulations.

This section presents the main notation and the mathermatical formulations of the MVPRP with ML and OU policy.

2.1. Notation.

The PRP is defined on an undirected graph $G=(N,E)$ with the following notations,

Sets:

T set of time periods, indexed by $t \in \{1,\dots,l\}$, and $T' = T \cup \{l+1\}$;

N set of plant and customers, indexed by $i \in \{0,\dots,n\}$, where the plant is represented by node 0 and $N_c = N \setminus \{0\}$ is the subset of n customers;

E set of edges, $E = \{(i,j) : i,j \in N, i<j\}$;

K set of identical vehicles, indexed by $k \in \{1,\dots,m\}$;

$E(S)$ set of edges $(i,j) \in E$ that have both end points in S, where $S \subseteq N$ is a given set of nodes;

$\delta(i)$ set of edges that have one end point at node i;

Decision variables:

p_t production quantity in period t;

I_{it} inventory at node i at the end of period t;

y_t equal to 1 if there is production at the plant in period t, 0 otherwise;

z_{ikt} equal to 1 if node i is visited by vehicle k in period t, 0 otherwise;

x_{ijkt} if vehicle k travels between node i and node j in period t, 0 otherwise;

q_{ikt} quantity delivered to customer i with vehicle k in period t;

Parameters:

u unit production cost;

f fixed production setup cost;

h_i unit inventory holding cost at node i;

c_{ij} transportation cost between node i and node j;

d_{it} demand at customer i in period t;

C production capacity;

Q vehicle capacity;

L_i maximum or taget inventory level at node i;

I_{i0} initial inventory available at node i;

2.2. Multi-Vehicle Formulations for the ML Policy.

In this section, we introduce the formulations for the MVPRP-ML. The first formulation is an extension of the single vehicle PRP formulation using a vehicle index, while a new formulation without vehicle index is presented next.

2.2.1. Formulation with Vehicle Index for the ML Policy. To formulate the MVPRP-ML with vehicle index, we extend the single vehicle PRP formulation used in Archetti et al. (2007, 2011a), as follows.

$$\min \sum_{t \in T} (up_t + fy_t + \sum_{i \in N} h_i I_{it} + \sum_{(i,j) \in E} \sum_{k \in K} c_{ij} x_{ijkt}) \tag{1}$$

s.t.

$$I_{0,t-1} + p_t = \sum_{i \in N_c} \sum_{k \in K} q_{ikt} + I_{0t} \qquad \forall t \in T \tag{2}$$

$$I_{i,t-1} + \sum_{k \in K} q_{ikt} = d_{it} + I_{it} \qquad \forall i \in N_c, \forall t \in T \tag{3}$$

$$p_t \leq \min\{C, \sum_{i \in N_c} \sum_{j=t}^{l} d_{ij}\} y_t \qquad \forall t \in T \tag{4}$$

$$I_{0t} \leq L_0 \qquad \forall t \in T \tag{5}$$

$$I_{i,t-1} + \sum_{k \in K} q_{ikt} \leq L_i \qquad \forall i \in N_c, \forall t \in T \tag{6}$$

$$\sum_{i \in N_c} q_{ikt} \leq Q z_{0kt} \qquad \forall k \in K, \forall t \in T \tag{7}$$

$$\sum_{k \in K} z_{ikt} \leq 1 \qquad \forall i \in N_c, \forall t \in T \tag{8}$$

$$q_{ikt} \leq \min\{L_i, Q, \sum_{j=t}^{l} d_{ij}\} z_{ikt} \qquad \forall i \in N_c, \forall k \in K, \forall t \in T \tag{9}$$

$$\sum_{(i,j)\in\delta(i)} x_{ijkt} = 2z_{ikt} \qquad \forall i \in N, \forall k \in K, \forall t \in T \qquad (10)$$

$$\sum_{(i,j)\in E(S)} x_{ijkt} \leq \sum_{i\in S} z_{ikt} - z_{ekt} \qquad \forall S \subseteq N_c, |S| \geq 2, \forall e \in S, \forall k \in K, \forall t \in T \qquad (11)$$

$$p_t, I_{it}, q_{ikt} \geq 0 \qquad \forall i \in N, \forall k \in K, \forall t \in T \qquad (12)$$

$$y_t, z_{ikt}, \in \{0,1\} \qquad \forall i \in N, \forall k \in K, \forall t \in T \qquad (13)$$

$$x_{ijkt} \in \{0,1\} \qquad \forall (i,j) \in E : i \neq 0, \forall k \in K, \forall t \in T \qquad (14)$$

$$x_{0jkt} \in \{0,1,2\} \qquad \forall j \in N_c, \forall k \in K, \forall t \in T. \qquad (15)$$

The objective function (1) minimizes the total production, setup, inventory and routing costs. Constraints (2) and (3) are the inventory flow balance at the plant and at the customers, respectively. Constraints (4) are the setup forcing and production capacity constraints at the plant: they force the setup variable to be one if production takes place and limit the production quantity to the minimum of the production capacity and the total demand in the remaining periods. The inventory quantity at the production facility at the end of each period is limited by constraints (5) and the inventory quantities at the customers after delivery cannot exceed their inventory capacities (6). The total quantity loaded in each vehicle can be at most the vehicle capacity as specified by (7). Constraints (8) allow each customer to be visited at most once in each period. Constraints (9) allow a positive delivery quantity from vehicle k to node i in period t only if this node is visited by the vehicle in period t. Since in the ML policy, it is never optimal to carry inventory at the end of the planning horizon, the delivery quantity to a customer is limited by the minimum value between the inventory capacity at the customer, the vehicle capacity or the total demand of the customer in the remaining periods. Constraints (10) are the degree constraints. They require the number of edges incident to node i to be 2 if it is visited. Constraints (11) eliminate subtours for each vehicle.

Archetti et al. (2007, 2011b) also strengthen the formulation using several valid inequalities. We present here the inequalities that are valid for the PRP with capacitated production. Note that we extend the original inequilities for the multi-vehicle case.

Denote by t' and t'', the earliest period that the plant must produce and the earliest period that at least once customer must be replenished to prevent stockout, respectively, i.e., $t' = \arg\min_{1\leq t\leq l}\{\sum_{i\in N_c} \max\{0, \sum_{j=1}^{t} d_{ij} - I_{i0}\} - I_{00} > 0\}$, and $t'' = \min_{i\in N_c} t''_i$ where

$t''_i = \arg\min_{1\le t\le l}\{\sum_{j=1}^{t} d_{ij} - I_{i0} > 0\}$. Let also κ be the minimum shipping quantity in t'', i.e., $\kappa = \sum_{i\in N_c}\max\{0, \sum_{j=1}^{t''} d_{ij} - I_{i0}\}$. First, these inequalities are used to prevent stockout.

$$\sum_{t=1}^{t'} y_t \ge 1 \tag{16}$$

$$\sum_{k\in K}\sum_{t=1}^{t''} z_{0kt} \ge \left\lceil \frac{\kappa}{Q} \right\rceil \tag{17}$$

Second, the inequalities below are imposed to strengthen the customer replenishment.

$$I_{i,t-s-1} \ge \sum_{j=0}^{s} d_{i,t-j}\left(1 - \sum_{k\in K}\sum_{j=0}^{s} z_{ik,t-j}\right) \qquad \forall i\in N_c, \forall t\in T, s=0,1...,t-1 \tag{18}$$

Finally, inequalities for the routing part are imposed.

$$z_{ikt} \le z_{0kt} \qquad \forall i\in N_c, \forall k\in K, \forall t\in T \tag{19}$$

$$x_{ijkt} \le z_{ikt} \text{ and } x_{ijkt} \le z_{jkt} \qquad \forall (i,j)\in E(N_c), \forall k\in K, \forall t\in T \tag{20}$$

The formulation (1)-(20) is referred to as $F(ML)|k$.

2.2.2. Formulation without Vehicle Index for the ML Policy. The previous formulation has a significant drawback since the number of variables will grow as the number of vehicles increases. Alternatively, the routing constraints can be replaced by variables without a vehicle index. The formulation is presented using the variables q, z and x with the same notations as the previous section but the vehicle index k is dropped, except for the variable z_{0t} which is changed to be an integer variable representing the number of vehicles leaving the production plant in period t. The formulation without vehicle index can be formulated as follows.

$$\min \sum_{t\in T}(up_t + fy_t + \sum_{i\in N} h_i I_{it} + \sum_{(i,j)\in E} c_{ij}x_{ijt}) \tag{21}$$

s.t. (4)-(5) and

$$I_{0,t-1} + p_t = \sum_{i\in N_c} q_{it} + I_{0t} \qquad \forall t\in T \tag{22}$$

$$I_{i,t-1} + q_{it} = d_{it} + I_{it} \qquad \forall i\in N_c, \forall t\in T \tag{23}$$

$$I_{i,t-1} + q_{it} \le L_i \qquad \forall i\in N_c, \forall t\in T \tag{24}$$

$$q_{it} \leq \min\{L_i, Q, \sum_{j=t}^{l} d_{ij}\} z_{it} \qquad \forall i \in N_c, \forall t \in T \qquad (25)$$

$$\sum_{(i,j)\in\delta(i)} x_{ijt} = 2z_{it} \qquad \forall i \in N, \forall t \in T \qquad (26)$$

$$z_{0t} \leq m \qquad \forall t \in T \qquad (27)$$

$$Q \sum_{(i,j)\in E(S)} x_{ijt} \leq Q \sum_{i\in S} (z_{it} - q_{it}) \qquad \forall S \subseteq N_c, |S| \geq 2, \forall t \in T \qquad (28)$$

$$p_t, I_{it}, q_{it} \geq 0 \qquad \forall i \in N, \forall t \in T \qquad (29)$$

$$y_t, z_{it}, \in \{0,1\} \qquad \forall i \in N_c, \forall t \in T \qquad (30)$$

$$z_{0t} \in \mathbb{Z}^+ \qquad \forall t \in T \qquad (31)$$

$$x_{ijt} \in \{0,1\} \qquad \forall (i,j) \in E : i \neq 0, \forall t \in T \qquad (32)$$

$$x_{0jt} \in \{0,1,2\} \qquad \forall j \in N_c, \forall t \in T. \qquad (33)$$

Constraints (22)-(26) are equivalent to (2)-(3), (6) and (9)-(10), respectively. Constraints (27) limit the number of vehicles leaving the production facility to the number of available vehicles in each period. Constraints (28) are the subtour elimination and vehicle capacity constraints. When we divide the inequalities by Q, these constraints have a similar form as the generalized fractional subtour elimination constraints (GFSECs) for the VRP (Toth and Vigo 2001). Unlike GFSECs in the VRP, however, we cannot round up the value of the term q_{it}/Q because it contains the q_{it} variable. We prefer to use the from (28) since initial tests indicated that the original form of GFSECs is numerically unstable due to the fractional RHS value.

We can also rewrite inequalities (17)-(20) for the non-vehicle index formulation as follows.

$$\sum_{j=1}^{t''} z_{0j} \geq \left\lceil \frac{\kappa}{Q} \right\rceil \qquad (34)$$

$$I_{i,t-s-1} \geq \sum_{j=0}^{s} d_{i,t-j} \left(1 - \sum_{j=0}^{s} z_{i,t-j}\right) \qquad \forall i \in N_c, \forall t \in T, s = 0, 1..., t-1 \qquad (35)$$

$$z_{it} \leq z_{0t} \qquad \forall i \in N_c, \forall t \in T \qquad (36)$$

$$x_{ijt} \leq z_{it} \text{ and } x_{ijt} \leq z_{jt} \qquad \forall (i,j) \in E(N_c), \forall t \in T. \qquad (37)$$

The non-vehicle index formulation together with the inequalities in this section and (16) is referred to as $F(ML)|nk$.

2.3. Multi-Vehicle Formulations for the OU Policy.

This section presents the formulations for the MVPRP-OU. Similar to the previous section, the formulations, with and without vehicle index, are proposed.

2.3.1. Formulations with Vehicle Index for the OU Policy. In the OU policy, when a customer is visited, the inventory before demand consumption must be replenished to reach its TSL. To enforce these constraints, one can add the constraints below to the formulation $F(ML)|k$ (Archetti et al. 2007, 2011b),

$$q_{ikt} \geq L_i z_{ikt} - I_{i,t-1} \qquad \forall i \in N_c, \forall k \in K, \forall t \in T. \tag{38}$$

However, unlike the ML policy where the ending inventory levels at both the production plant and customers must be zero in the optimal solution if the inventory costs are strictly positive, the ending inventory levels under the OU policy can be positive to satisfy constraints (38). Therefore, constraints (4) and (9) have to be replaced by the following constraints,

$$p_t \leq C y_t \qquad \forall t \in T \tag{39}$$

$$q_{ikt} \leq \min\{Q, L_i\} z_{ikt} \qquad \forall i \in N_c, \forall k \in K, \forall t \in T. \tag{40}$$

Constraints (3), (6), (38) and (40) represent the OU policy. To strengthen the formulation, Archetti et al. (2007) also added the inequalities below.

$$I_{i,t-1} \geq L_i \sum_{k \in K} z_{ik,t-s} - \left(\sum_{j=t-s+1}^{t} d_{ij} \right) \qquad \forall i \in N_c, \forall t \in T, s = 1, 2, ..., t-1. \tag{41}$$

The formulation (1)-(3), (5)-(7), (10)-(20), and (39)-(41) is reffered to as $WF(OU)|k$.

However, it has been shown in Solyalı and Süral (2011) that a stronger version of the IRP-OU can be obtained using a shortest-path network representation for the customer inventory replenishment part. This reformulation scheme exploits the characteristic of the OU policy that the delivery quantity for a customer visited in period t is equal to the total demand in the interval between t and the previous visit in period $v < t$. To formulate a strong MVPRP-OU formulation, we adopt the reformulation presented in Solyalı and Süral (2011) and extend it using a vehicle index. We define $d_{i0} = d_{i,l+1} = 0$ and use the following notation.

Decision variables:

λ_{ikvt} equal to 1 if node i is visited by vehicle k in period t and the previous visit is in period v;

Parameters:

g_{ivt} total delivery quantity when customer i is visited in period t and the previous visit is in period v;

e_{ivt} total inventory holding cost when customer i is visited in period t and the previous visit is in period v;

$\mu(i,t)$ the latest period after period t that the customer i can be replenished without having a stockout, $\mu(i,t) = \arg\max_{t<v\leq l+1}\{g_{itv} + d_{iv} \leq L_i\}$;

$\pi(i,t)$ the earliest period before period t that the customer i can be replenished without being stockout, $\pi(i,t) = \arg\max_{0\leq v<t}\{g_{ivt} + d_{it} \leq L_i\}$.

The parameters g_{itv} and e_{itv} can be calculated as follows.

$$g_{ivt} = \begin{cases} \sum_{j=1}^{t} d_{ij} + (L_i - I_{i0} - d_{it}) & \text{if } v = 0 \\ \sum_{j=v+1}^{t} d_{ij} & \text{if } 0 < v < t \leq l \\ 0 & \text{if } t = l+1 \end{cases}$$

$$e_{ivt} = \begin{cases} \eta_{ivt} & \text{if } t < l+1 \\ \eta_{ivt} - h_i L_i & \text{otherwise,} \end{cases}$$

where

$$\eta_{ivt} = \begin{cases} h_i\left(\sum_{j=v+1}^{t-1}(I_{i0} - \sum_{l=v}^{j} d_{il}) + (L_i - d_{it})\right) & \text{if } v = 0. \\ h_i\left(\sum_{j=v+1}^{t-1}(L_i - \sum_{l=v}^{j} d_{il}) + (L_i - d_{it})\right) & \text{if } v < t \leq l+1 \end{cases}$$

A preprocessing can be used to eliminate variables associated with infeasible delivery quanity $g_{ivt} > Q$ and $g_{ivt} > L_i$. The strong formulation, is referred to as $F(OU)|k$, is as follows.

$$\min \sum_{t\in T}(up_t + fy_t + h_0 I_{0t} + \sum_{(i,j)\in E}\sum_{k\in K} c_{ij}x_{ijkt}) + \sum_{i\in N_c}\sum_{k\in K}\sum_{t=T'}\sum_{v=\pi(i,t)}^{t-1} e_{ivt}\lambda_{ikvt} \qquad (42)$$

s.t. (5), (39), (8)-(15), and

$$I_{0,t-1} + p_t = \sum_{i \in N_c} \sum_{k \in K} \sum_{v=\pi(i,t)}^{t-1} g_{ivt}\lambda_{ikvt} + I_{0t} \qquad \forall t \in T \qquad (43)$$

$$\sum_{i \in N_c} \sum_{v=\pi(i,t)}^{t-1} g_{ivt}\lambda_{ikvt} \leq Q z_{0kt} \qquad \forall k \in K, \forall t \in T \qquad (44)$$

$$\sum_{v=\pi(i,t)}^{t-1} \lambda_{ikvt} = z_{ikt} \qquad \forall i \in N_c, \forall k \in K, \forall t \in T \qquad (45)$$

$$\sum_{k \in K} \sum_{t=1}^{\mu(i,t)} \lambda_{ik0t} = 1 \qquad \forall i \in N_c \qquad (46)$$

$$\sum_{k \in K} \sum_{v=\pi(i,v)}^{t-1} \lambda_{ikvt} - \sum_{k \in K} \sum_{v=t+1}^{\mu(i,t)} \lambda_{iktv} = 0 \qquad \forall i \in N_c, \forall t \in T \qquad (47)$$

$$\sum_{k \in K} \sum_{t=\pi(i,l+1)}^{l} \lambda_{ikt,l+1} = 1 \qquad \forall i \in N_c \qquad (48)$$

$$\lambda_{ikvt} \in \{0,1\} \qquad \forall i \in N_c, \forall k \in K, \forall v,t \in T \qquad (49)$$

The objective function (42), constraints (43) and (44) are equivalent to (1), (2) and (7), respectively. Constraints (45) provide the link between the λ_{ikvt} and z_{ikt} variables. Constraints (46)-(48) represent the shortest-path network of the OU policy at each customer. It should be noted that Solyalı and Süral (2011) used the original objective function (1) and still retain equivalent constraints (3). However, since customer inventory constraints are already controlled by the new variable λ_{ikvt}, we use the corresponding inventory cost e_{ivt} associated with variable λ_{ikvt} and drop the customer inventory constraints in this formulation.

Similar to the formulation presented in Solyalı and Süral (2011), the inequalities (19)-(20) are also added to strengthen the routing part of the formulation. We further add (16)-(17) to reinforce the production part.

2.3.2. Formulation without vehicle index for the OU Policy. The non-vehicle index formulation for the OU policy can be formulated using the same notations as the formation $F(ML)|nk$ and the variable λ as the previous section, but without vehicle index k. The formulation, referred to as $F(OU)|nk$, is as follows.

$$\min \sum_{t \in T} (up_t + fy_t + h_0 I_{0t} + \sum_{(i,j) \in E} c_{ij} x_{ijt}) + \sum_{i \in N_c} \sum_{t=T'} \sum_{v=\pi(i,t)}^{t-1} e_{ivt} \lambda_{ivt} \quad (50)$$

s.t. (5), (39) , (26)-(27), and

$$I_{0,t-1} + p_t = \sum_{i \in N_c} \sum_{v=\pi(i,t)}^{t-1} g_{ivt} \lambda_{ivt} + I_{0t} \qquad \forall t \in T \quad (51)$$

$$\sum_{v=\pi(i,t)}^{t-1} \lambda_{ivt} = z_{it} \qquad \forall i \in N_c, \forall t \in T \quad (52)$$

$$\sum_{t=1}^{\mu(i,t)} \lambda_{i0t} = 1 \qquad \forall i \in N_c \quad (53)$$

$$\sum_{v=\pi(i,v)}^{t-1} \lambda_{ivt} - \sum_{v=t+1}^{\mu(i,t)} \lambda_{itv} = 0 \qquad \forall i \in N_c, \forall t \in T \quad (54)$$

$$\sum_{t=\pi(i,l+1)}^{l} \lambda_{it,l+1} = 1 \qquad \forall i \in N_c \quad (55)$$

$$Q \sum_{(i,j) \in E(S)} x_{ijt} \leq Q \sum_{i \in S} (z_{it} - \sum_{v=\pi(i,t)}^{t-1} g_{ivt}) \lambda_{ivt} \qquad \forall S \subseteq N_c, |S| \geq 2, \forall t \in T \quad (56)$$

$$\lambda_{ivt} \in \{0,1\} \qquad \forall i \in N_c, \forall v, t \in T \quad (57)$$

Constraints (51)-(55) are equivalent to (43), (45)-(48) and , respectively. Constraints (56) are equivalent to (28). Note that inequalities (16), (34) and (36)-(37) are also added a priori in our implementation in order to make a fair comparison to the other formulations.

2.4. Formulations for the MVIRP.

All the formulation above can be easily modify to solve the MVIRP where the production part (i.e., production setup and quantity decisions) is disregarded. First, the production setup variable y_t is set to one, $y_t = \bar{y}_t = 1, \forall t \in T$. Second, the production quantity is set to the production quantity made available in each period, denote by B_t, by replacing the production capacity constraints (4) and (39) with,

$$p_t = B_t \bar{y}_t \qquad \forall t \in T \quad (58)$$

The rest of the formulations remain unchanged.

3. Valid Inequalities.

In this section, we propose other sets of valid inequalities for the formulations. The first group of the inequalities is developed for the vehicle index formulations $F(ML)|k$, $WF(OU)|k$ and $F(OU)|k$, while the second group is used to strengthen the non-vehicle index formulations $F(ML)|nk$, $F(OU)|nk$.

3.1. Valid Inequalities for the Vehicle Index Formulations.

Denote by $\bar{m}_t$, the number of dispatched vehicles in period t. In each period t, there are two main symmetry issues concerning vehicle index. First, in vehicle dispatching, there are $\binom{m}{\bar{m}_t}$ possible options to select $\bar{m}_t$ vehicles from the fleet. Second, among the selected vehicles, there are still $\bar{m}_t!$ options to swap the routes that are assigned to each dispatched vehicle. These two types of symmetry are present in each period, and hence there can be $\left[\binom{m}{\bar{m}_1}\bar{m}_1!\right]\left[\binom{m}{\bar{m}_2}\bar{m}_2!\right]...\left[\binom{m}{\bar{m}_l}\bar{m}_l!\right]$ equivalent solutions . For example, for an instance with 3 periods and 3 vehicles, if 2 vehicles are used in each period, there are $\left[\binom{3}{2}2!\right]^3 = 216$ equivalent solutions that can be obtained by re-indexing the vehicles.

To break the first type of symmetry, we can use the symmetry breaking constraints (SBC) below to allow vehicle $k+1$ to be dispatched only if vehicle k is already dispatched.

$$\text{(SBC0)} \quad z_{0k1} \geq z_{0,k+1,t} \qquad 1 \leq k \leq m-1, \forall t \in T.$$

To resolve the second symmetry issue, we can use one of the following different versions of symmetry breaking constraints. Note that these constraints cannot be imposed together but each of them can be used in conjunction with SBC0. The first constraints break the symmetry of the routes by ordering them according to their total route costs.

$$\text{(SBC1)} \quad \sum_{(i,j)\in E} c_{ij}x_{ijkt} \geq \sum_{(i,j)\in E} c_{ij}x_{ij,k+1,t} \qquad 1 \leq k \leq m-1, \forall t \in T.$$

Alternatively, we can impose that the vehicles are ordered according to their total delivery quantity.

$$\text{(SBC2)} \quad \sum_{i\in N_c} q_{ijkt} \geq \sum_{i\in N_c} q_{ij,k+1,t} \qquad 1 \leq k \leq m-1, \forall t \in T \quad \text{for } F(ML)|k \text{ and } WF(OU)|k$$

$$\text{or} \quad \sum_{i\in N_c}\sum_{v=t+1}^{\mu(i,t)} g_{itv}\lambda_{iktv} \geq \sum_{i\in N_c}\sum_{v=t+1}^{\mu(i,t)} g_{itv}\lambda_{i,k+1,tv} \qquad 1 \leq k \leq m-1, \forall t \in T \quad \text{for } F(OU)|k.$$

We also use the lexicographic ordering constraints (Sherali and Smith 2001, Degraeve et al. 2002, Jans 2009) to assign a unique number to each possible set of customers for a route and we order the vehicles according to their assigned number. In order to ensure the uniqueness, we use powers of two for the coefficients as described in Jans (2009). This type of inequalities is efficient because a customer can be part of at most one route. The lexicographic ordering constraints can be imposed first with respect to customer one only, next with respect to customer one and two, and so on.

$$\text{(SBC3)} \quad \sum_{i=1}^{j} 2^{(j-i)} z_{ikt} \geq \sum_{i=1}^{j} 2^{(j-i)} z_{i,k+1,t} \qquad \forall j \in N_c, 1 \leq k \leq m-1, \forall t \in T.$$

We can also only use the final constraint of SBC3 including all the customers to impose a unique ordering in each period:

$$\text{(SBC4)} \quad \sum_{i=1}^{n} 2^{(n-i)} z_{ikt} \geq \sum_{i=1}^{n} 2^{(n-i)} z_{i,k+1,t} \qquad 1 \leq k \leq m-1, \forall t \in T.$$

The computational results of using different SBC are provided in Section 6.2.1.

3.2. Valid Inequalities for the Non-Vehicle Index Formulations.

To strengthen the non-vehicle index formulations, we also add the following inequalities a priori.

- For the formulation $F(ML)|nk$,

$$Qz_{0t} \geq \sum_{i \in N_c} q_{it} \qquad \forall t \in T \tag{59}$$

- For the formulation $F(OU)|nk$,

$$Qz_{0t} \geq \sum_{i \in N_c} \sum_{v=\pi(i,t)}^{t-1} g_{ivt} \lambda_{ivt} \qquad \forall t \in T \tag{60}$$

Constraints (59) and (60) impose the number of vehicles leaving the production facility must be sufficient to carry delivery quantities to all customers in each period.

Since GFSECs (28) and (56) are generally weak as described in section 2.2.2, we further strengthen the formulation by adding the subtour elimination constraints (SECs) below,

$$\sum_{(i,j) \in E(S)} x_{ijt} \leq \sum_{i \in S} z_{it} - z_{et} \quad e \in S, \forall S \subseteq N_c, |S| \geq 2, \forall t \in T \tag{61}$$

These cuts are used to prevent subtours in each period, but they do not take into account the vehicle capacity. Therefore, they have to be used together with GFSECs (28) or (56) in order to generate feasible multi-vehicle routes.

To take into account the periodic routing decisions of the MVPRP, we also add another set of constraints to strengthen the formulation, called multi-period generalized fractional subtour elimination constraints (MGFSECs), as follows.

- For the formulation $F(ML)|nk$,

$$Q\sum_{t\in R}\sum_{(i,j)\in E(S)} x_{ijt} \le Q\sum_{t\in R}\sum_{i\in S}(z_{it}-q_{it}) \quad \forall S\subseteq N_c, |S|\ge 2, ,\forall R\subseteq T \tag{62}$$

- For the formulation $F(OU)|nk$,

$$Q\sum_{t\in R}\sum_{(i,j)\in E(S)} x_{ijt} \le Q\sum_{t\in R}\sum_{i\in S}\left(z_{it}-\sum_{v=\pi(i,t)}^{t-1} g_{ivt}\lambda_{ivt}\right) \quad \forall S\subseteq N_c, |S|\ge 2, ,\forall R\subseteq T \tag{63}$$

Similar to the GFSECs (28) and (56), contraints (62) and (63) prevent subtours and ensure that the number of vehicles is sufficient to carry the delivery quantity to the set of customers S during the time period set R. These constraints are a combined version of the GFSECs and equal to GFSECs when $|R| = 1$.

Denote by $\rho(S,r)$, the minimum number of vehicles must be dispatched to carry the demands in customer set S during period 1 to r, calculated as $\rho(S,r) = \left\lceil \sum_{i\in S}\left(\sum_{t=1}^{r} d_{it} - I_{i0}\right)^+ / Q \right\rceil$. The inequalities below are also imposed.

$$\sum_{t=1}^{r}\sum_{(i,j)\in E(S)} x_{ijt} \ge 2\rho(S,r) \quad \forall S\subseteq N_c, |S|\ge 2, r\in T \tag{64}$$

These constraints ensure that total number of vehicles entering the set of customers S from period 1 to period t must be sufficient to carry the demands during that period.

The branch-and-cut algorithm that incorporates the three subtour elimination and vehicle capacity cuts is described in the next section.

4. Branch-and-cut Approaches.

Since all the formulations contain an exponentially large number of subtour elimination constraints, a natural way to solve the problems is to use the branch-and-cut technique. In this process, the subtour elimination constraints, i.e., constraints (11) for the $F(ML)|k$, $WF(OU)|k$ and $F(OU)|k$, and GFSECs for the $F(ML)|nk$ and $F(OU)|nk$, are dropped and are added iteratively when they are violated at each node of a branch-and-bound tree. In this section, we provide the details of the branch-and-cut approaches for both types of formulations. For the variable selection, we first branch on the y, z, and x variables, respectively. We use the default setting in CPLEX 12.3 to select a specific variable to branch on. The remaining parameters are set to default where CPLEX employs a best-bound-first strategy.

4.1. Branch-and-cut for the Vehicle Index Formulations.

To solve the vehicle index formulations $F(ML)|k$, $WF(OU)|k$ and $F(OU)|k$, we use an exact separation algorithm for the TSP by solving a minimum $s-t$ cut problem to detect violated TSP subtours for each vehicle in each period. This is valid since vehicle tours are separated by the vehicle index. At each node of the branch and bound tree, if a violated tour S for vehicle k in period t is found, we add the violated cuts (11) with $e = argmax_{i\in S}\{z_{ikt}\}$ to the formulation. We implemented the separation algorithm described in Ruokokoski et al. (2010) and use the minimum $s-t$ algorithm of the Concorde callable library (Applegate et al. 2011). Although this separation algorithm is different from the algorithm presented in Archetti et al. (2007, 2011a) and Solyalı and Süral (2011) where the heuristic separation algorithm for the TSP (Padberg and Rinaldi 1991) is used, it is efficient in our branch-and-cut algorithm since all the instances 14c/6p/1v in Archetti et al. (2011a) which were not solved to optimality, are all solved to optimality within few seconds and the results presented by Archetti et al. (2007) and Solyalı and Süral (2011) are all improved in this implementation while using a similar workstation.

We add a few remarks on the vehicle index formulations. First, we have tested a few possible options of the subtour elimination constraints. The first option is to use constraints (11) and when a violated cut is detected, add it only for the specific vehicle for which it was found. The second option is to add this cut for *all* vehicles in the same period instead of adding it to the specfic vehicle only. The third option is to use a combined version of constraints (11), which is also valid,

$$\sum_{k\in K}\sum_{(i,j)\in E(S)} x_{ijkt} \le \sum_{k\in K}\sum_{i\in S} z_{ikt} - \sum_{k\in K} z_{ekt} \qquad \forall S\subseteq N_c, |S|\ge 2, \forall e\in S, \forall t\in T. \tag{65}$$

The results indicated that the first option, adding only the violated constraints (11), was the best cut generation strategy among the three, while the second option was slightly worse and the third option was far worse than the other two. Second, similar to the results provided by Solyalı and Süral (2011) for the single vehicle case, the results of $WF(OU)|k$ were all worse than the $F(OU)|k$ and $F(OU)|nk$ for multiple vehicle case and we do not include the results in the experiments.

4.2. Branch-and-cut for the Non-Vehicle Index Formulation.

The three different subtour eliminations, GFSECs ((28) and (56)), SECs (61) and MGF-SECs ((62) and (63)), are used for the formulations without vehicle index. To detect SECs, we use the same separation algorithm for the vehicle index formulations as described above to find and generate the cuts for each period t. For the GFSECs, we use the four heuristic separation algorithms described in Lysgaard et al. (2004) that were proposed for the CVRP. To generate the cuts, we call the separation algorithms for each period t. Denote by $\bar{x}_{ijt}$, $\bar{q}_{it}$ and $\bar{\lambda}_{ivt}$, the current values of variables x_{ijt}, q_{it} and λ_{ivt} in the branch and bound tree. We solve the separation algorithms by setting the weights of edges to $\bar{x}_{ijt}$ and the delivery quantity for customer i to $\bar{q}_{it}$ for $F(ML)|nk$ and $\sum_{v=\pi(i,t)}^{t-1} g_{ivt}\bar{\lambda}_{ivt}$ for $F(OU)|nk$. Note that the maximum number of subsets produced by the separation algorithms for each period t is limited to n per call. Then, the violated GFSECs are added to the formulation.

For the MGFSECs, we develop a greedy heuristic separation algorithm to detect the cuts. Denote by $\bar{z}_{it}$, the current solution values of variables z_{it}. For each subset $R \subseteq T$, we calculate the value $s_i = \sum_{t\in R}(\bar{q}_{it}/\lceil \bar{z}_{it}\rceil)$ (or $s_i = \sum_{t\in R}(\sum_{v=\pi(i,t)}^{t-1} g_{ivt}\bar{\lambda}_{ivt}/\lceil \bar{z}_{it}\rceil)$ for $F(OU)|nk$) which represents the average delivery quantity to customer i per visit during the period set R. Customers are ranked according to the value of s_i in descending order and stored in an ordered list. Then, an empty set of customers S and an empty set of violated sets $\xi(S)$ are created. The separation algorithm starts by adding the first customer in the ordered list to S and checks whether the MGFSEC of the S is violated. The next customer is then added to S and the algorithm checks for the MGFSEC again, and so on. The violated MGFSECs are stored in $\xi(S)$. If a violated MGFSEC is found or all the customers in the ordered list are added to S, the set S is reset to empty and the first customer in the ordered list is removed, then the algorithm starts again by adding the first customer in the ordered list to S. This process is repeated until the ordered list is empty or n violated cuts are found. In our implementation, we consider the subset R of all two and three consecutive periods in T.

The separation algorithm above can also used to generate the cuts (64). We first set $R = \{t : 1 \leq t \leq r\}, \forall r \in T$ or the set from time period 1 to $r \in T$. and use the demand d_{it} in place of the delivery quantity $\bar{q}_{it}$. The algorithm is then set to detect the cuts (64) instead of MGFSECs.

Since it is very time comsuming to solve all the separation algorithms at every node of the branch-and-bound tree, we use the following cut generation strategy for the non-vehicle index formulations. At the root node, all the separation algorithms are called to generate the GFSECs, SECs, MGFSECs and (64). At each further node of branch-and-bound tree, only the GFSECs and SECs are considered in the following sequence: (1) the seperation algorithm of the SECs is called, (2) If there is no violated SECs, the separation algorithms for the GFSECs are called.

The process as described above can significantly improve the performance of the algorithm since using GFSECs alone is inefficient due to the fractional coefficients. Using the cuts SECs can efficiently eliminate subtours and create initial vehicle route, while adding GFSECs and MGFSECs can eliminate the exceeded vehicle capacity routes. The computational result in Section 6.3 show a significant performance improvement by using the three cuts together compared to using GFSECs alone.

5. Optimization-Based Adaptive Large Neighborhood Search (Op-ALNS) Heuristic.

In this section, we present a heuristic to calculate upper bound used in the branch-and-cut algorithms. We use a heuristic based on the adaptive large neighborhood search (ALNS) framework that is proposed by Ropke and Pisinger (2006) for the VRP to solve the MVPRP. The basic idea of the ALNS is to repeatedly destroy and repair a solution using several heuristic operators to seek for improvement. These operators are probabilistically selected based on empirical scores related to the success of operators. We use an adaptation of the ALSN framework to handle the binary variables, and the remaining continuous variables are evaluated by solving a network flow model that is embedded into the operators of the ALSN. This procedure is called optimization-based adaptive large neighborhood search (Op-ALNS).

5.1. The Op-ALNS for the ML policy.

The implementation of the Op-ALNS for the MVPRP-ML can be found in Adulyasak et al. (2011). The procedure consists of two main phases, i.e., initialization and improvement.

In the **initialization** phase, a pool of different initial solutions is constructed by solving two decomposed subproblems. The first subproblem is solved to determine the production, inventory levels and assigned customers to be visited, and the second subproblem is used to determine the route for each vehicle. After the first initial solution is generated, a different initial solution is created using the local branching technique (Fischetti and Lodi 2003). For the PRP, since production setup decisions usually form an important part of the objective function, we generate initial solutions with different production setups. Denote by s, the solution index, and $\bar{y}_t^s$, the value of the production setup variable y_t in solution s. We add the local branching inequality below to the first subproblem to generate the initial solution $s+1$. The local branching constraints are added cumulatively. Note that we set the maximum number of the initial solutions in the pool to l for the MVPRP.

$$\sum_{y_t|\bar{y}_t^s=1} (1-y_t) + \sum_{y_t|\bar{y}_t^s=0} y_t \geq 1. \tag{66}$$

For the IRP, since production setups are irrelevant, we generate initial solutions with diffent customer visit decisions. Denote by $\bar{z}_{it}^s$, the value of the customer visit variable z_{it} in solution s. The local branching inequality (66) is replaced by the inequality below to generate an initial solution $s+1$,

$$\sum_{z_{it}|\bar{z}_{it}^s=1} (1-z_{it}) + \sum_{z_{it}|\bar{z}_{it}^s=0} z_{it} \geq \left\lceil 0.25 \sum_{i\in N_c}\sum_{t\in T} \bar{z}_{it}^s \right\rceil. \tag{67}$$

The inequality (67) enforces at least 25% of the total customer visits over the horizon is changed. The maximum number of the initial solutions in the pool is set to 10 for the MVIRP.

The **improvement** phase starts when maximum number of initial solutions is generated. The integer and binary variables x and z are handled by ALNS operators and the remaining continuous variables p, I and q are set by solving the minimum cost flow (MCF) problem. The process starts by selecting an initial solution from the top of the initial solution pool. At each iteration in this phase, two types of operators, called *selection* and *transformation*, are called to seek for improvement. The selection operators are used to select some customer-period combinations, called node candidates, and they are put into an ordered list. These node candidates are removed and reinserted by the transformation operators. One selection and one transformation operator are probabilistically selected at each iteration based on empirical scores. In the transformation process, once node candidates are moved according to the operator, all the binary variables are fixed and the MCF is called to determine the corresponding optimal continuous variables. The new solutions found during the transformation are accepted according to a simulated annealing criterion. The process stops after a given number of iterations have been performed. We refer to Adulyasak et al. (2011) for further details and all the parameter settings in the improvement phase.

5.2. The Op-ALNS for the OU policy.

Unlike the ML policy where the selection and transformation operators could handle infeasible routes effectively by repeatedly reallocating delivery quantity, it is much more difficult to remove and reinsert node candidates from infeasible routes in the OU policy since the delivery quantity is defined by the difference of inventory level and TSL. Thus, it is easier to start from initial solutions with feasible routes in the initialization process to ensure that feasible solutions can be obtained at the end of the process. We simply solve the formulation with vehicle index to take into account the vehicle capacity for each vehicle separately. We denote by $\sigma_i = \min\{2c_{0i}, \min_{j,k \in N, j \neq k}(c_{ij} + c_{ik})\}$, or the minimum value between the cost of making a round trip from the production facility and the cost to the nearest two neighbors of customer i. The first subproblem in the initialization phase for the MVPRP-OU and MVIRP-OU is as follows.

$$\min \sum_{t \in T} (up_t + fy_t + h_0 I_{0t} + \sum_{i \in N_c} \sum_{k \in K} \sigma_i z_{ikt}) + \sum_{t \in T'} \sum_{i \in N_c} \sum_{k \in K} \sum_{v=1}^{l+1} e_{itv} \lambda_{iktv} \quad (68)$$

subject to (4), (8), (12)-(13) (43)-(49).

The best version of the SBCs from Section sub:Inequalities-k is also added to improve the performance of the formulation. Note that the delivery quantity variable q_{ikt} is calculated as $q_{ikt} = \sum_{v=1}^{l+1} g_{itv} \bar{\lambda}_{iktv}$. After the first subproblem is solved, the routes for the vehicles are identified by solving the traveling salesman problem (TSP) for each vehicle individually. Similar to Adulyasak et al. (2011), we construct the TSP tours using the GENIUS procedure (Gendreau et al. 1992) and improve them by the 3-opt procedure (Lin 1965). Additionally, to solve the MCF for the OU policy, we set the inventory level of a visited customer i in period t equal to $L_i - d_{it}$. The remaining of the Op-ALNS algorithm for the OU policy is the same as the ML policy.

6. Computational Experiment.

The algorithms are coded in C# on MonoDevelop 2.2 using CPLEX 12.3. The experiments on the branch-and-cut approaches have been executed on a workstation with AMD Opteron 2.40GHz and 16GB of RAM, while the Op-ALNS has been executed on with a workstation with 2.10 GHz CPU and 2 GB of RAM. The maximum computing time is limited to 3600 seconds.

We report the average computing times in seconds and the average number of nodes in the column Time and Node, respectively. Columns %LB show the percentage of the final lower bounds of a specific approach with respect to the best upper bounds found among all the approaches. Boldface letters are used to indicate the best results. If all instances in a problem size are solved to optimality, we put boldface letters on the lowest total computing time, otherwise boldface letters are put on the best average percentage of lower bounds.

6.1. Details of the Instances.

We create two test sets from the instance presented in literature. The first test set consists of MVPRP instances and the second test set consists of MVIRP instances. The details of the instance sets are as follows.

6.1.1. Details of the MVPRP Instances. There are two published MVPRP datasets that are used in several studies, i.e., Boudia et al. (2005) and Archetti et al. (2011b), but both datasets were designed for heuristics and they are too large for our exact algorithms. We then consider generating a smaller dataset based on the data set from the literature for our computational experiment. Since the Archetti et al. (2011b) dataset takes into account many different aspects, e.g., inventory costs at customers, initial inventory, and varying transportation and production costs, while the Boudia et al. (2005) dataset has zero inventory cost at the customers and the problem sizes are generally too large, we decided to use a subset of the Archetti et al. (2011b) dataset to create our test set.

The Archetti et al. (2011b) dataset consists of instances with 6 periods. Each problem size contains 4 classes and each instance type has 5 instance sets with different node coordinates. The first class contains standard instances. The second and the third classes are identical to the first but with high unit production costs and high transportation costs, respectively. The fourth class consists of instances from the first and second class but with no customer inventory cost. We generate our instances using the instances from the four classes to ensure that we consider many different problem characteristics. The problem size with 50 customers is used to generate our dataset. There are four instances per problem size. The number of vehicles is set to $m = 2, 3$ for the instances with $n \leq 15$ and to $m = \lfloor n/10 \rfloor + 1$ for the instances with $n > 15$. The details of the instance generation are provided in the Appendix. We summarize the parameters of the two instance sets in table 2.

Table 2 Characteristics of the MVPRP Instance Sets

n	l	m	C	L_0	Q
10	3/6/9	2	304	152	198
10	3/6/9	3	304	152	132
15	3/6/9	2	470	235	198
15	3/6/9	3	470	235	132
20	3/6/9	3	540	270	189
25	3/6/9	3	700	350	189
30	3/6/9	4	768	384	171
35	3/6/9	4	948	474	207
40	3/6/9	5	1256	628	216

6.1.2. Details of the MVIRP Instances. We adopt the IRP instances for the single vehicle are presented in Archetti et al. (2007). This instance set was used in several literature (e.g., Archetti et al. 2007, 2011a, Solyalı and Süral 2011, Coelho et al. 2012). The instance set consists of 5 to 50 customers (with an increment of 5) with 3 periods and 5 to 30 customers with 6 periods. There are two main groups, i.e., low inventory costs and high inventory costs, and five instances per instance size in each group. To generate the MVIRP instances, we and simply divided the original vehicle capacity by the desired number of vehicles and rounded down to the nearest integer value. We use the instances with 5 to 30 customers from the instances with 3 periods and 5 to 15 customers from the instances with 6 periods in our experiment. The number of vehicles is set using the same method as the MVPRP instances. As we mentioned earlier, since the timing of the replenishment process in our paper and Archetti et al. (2007) are different, the details of the conversion are presented in the Appendix section.

6.2. Effect of Valid Inequalities.

This section presents the analysis of the inequalities in section 3. The first part discusses the effect of of the symmetry breaking constraints used to reinforce the vehicle index formulations as presented in Section 3.1 and the second part is the results of imposing valid inequalities in Section 3.2 to strengthen the non-vehicle index formulations. The experiments were conducted on the instances with $n \leq 15$.

6.2.1. Effect of Vehicle Symmetry Breaking Constraints on the Vehicle Index Formulations. We analyze the effect of symmetry breaking constraints SBC0 and SBC0 together with one of the other constraints SBC1-SBC4 for the formulation $F(ML)|k$ and $F(OU)|k$. The results on the MVPRP and MVIRP instances that are solved to optimality are shown in Table 3 and 4, respectively and the results on the instances that could not be solved to optimality are reported in Table 5 and 6, respectively. The column h_i in Table 4 and 6 indicates the group of the MVIRP instances, i.e., low (L) or high (H) inventory costs. In our test, we use the default setting in CPLEX which allows CPLEX to detect and generate its own symmetry breaking constraints.

Table 3 Results of using different symmetry breaking constraints on the MVPRP instances were solved to optimality

n	m	l	None		SBC0		SBC0+1		SBC0+2		SBC0+3		SBC0+4	
			Time	Node	Time	Node	Time	Node	Time	Node	Time	Node	Time	Node
MVPRP-ML														
10	2	3	0.5	12	0.4	3	**0.3**	3	0.4	2	**0.3**	1	0.4	1
10	3	3	3.3	88	1.7	24	1.7	13	1.7	15	**1.0**	11	1.0	14
15	2	3	7.1	109	5.0	55	**3.1**	21	5.4	58	4.7	63	4.8	66
15	3	3	259.9	3614	85.3	884	54.0	486	92.4	839	53.8	537	**47.5**	486
10	2	6	12.9	343	4.6	31	5.5	40	5.2	33	**2.5**	20	3.6	34
10	3	6	206.1	3588	48.1	675	35.6	337	37.8	423	**18.5**	221	22.7	295
15	2	6	396.4	2435	341.1	1971	159.3	719	374.6	1438	162.1	957	**159.4**	873
10	2	9	128.4	1815	77.9	981	40.2	299	56.7	513	**30.9**	274	35.6	350
10	3	9	3600.0$^{(4)}$	20059	2407.6$^{(1)}$	18138	966.9	6377	1740.9$^{(1)}$	12179	933.8	7557	**754.8**	6967
Average			512.7	3562	330.2	2529	140.7	921	257.2	1722	134.2	1071	**114.4**	1009
MVPRP-OU														
10	2	3	0.8	12	0.8	13	0.8	10	0.9	19	0.8	19	**0.7**	8
10	3	3	2.2	85	1.9	77	1.8	51	2.6	80	1.9	47	**1.6**	36
15	2	3	66.9	1775	58.7	1557	45.6	1037	86.5	2033	**31.0**	745	45.4	1064
15	3	3	99.8	1616	94.3	1395	106.4	1249	98.6	1166	29.9	328	**27.3**	287
10	2	6	1.2	7	0.9	3	0.9	5	1.0	3	1.0	4	**0.8**	3
10	3	6	3.7	24	3.1	13	3.6	15	3.4	11	2.9	11	**2.5**	8
15	2	6	25.3	165	22.1	136	23.4	101	23.6	119	**17.4**	86	23.5	123
15	3	6	332.0	1466	325.0	1372	183.8	540	320.2	1020	171.4	587	**147.9**	493
10	2	9	40.6	519	29.5	355	26.1	226	24.2	226	**17.8**	164	21.5	203
10	3	9	835.9	7228	573.5	4987	137.4	758	388.3	2448	**151.4**	915	155.5	973
15	2	9	2120.4$^{(2)}$	5766	2122.8$^{(2)}$	5766	1823.3$^{(1)}$	4649	2161.0$^{(2)}$	6125	**1871.3**	5740	2126.9$^{(2)}$	5544
Average			320.8	1696	293.9	1425	213.9	785	282.7	1204	**208.8**	786	232.1	795

$^{(-)}$ indicates the number of instances (out of 4) were not solved to optimality

Table 7 provides a summary of the time reduction factors for each approach, calculated as the average computing time spent to solve an instance size corresponding to Table 3-6 without using any of our SBCs, divided by the average computing time of using each cut strategy. A time factor equal to 2 means the algorithm spent 2 times shorter by using the SBC strategy than by using no additional SBCs.

Table 4 **Results of using different symmetry breaking constraints on the MVIRP instances were solved to optimality**

n	m	l	h_i	None		SBC0		SBC0+1		SBC0+2		SBC0+3		SBC0+4	
			Time	Node	Time	Node	Time	Node	Time	Node	Time	Node	Time	Node	
MVIRP-ML															
5	2	3	L	0.4	146	0.3	75	0.3	44	0.3	63	**0.2**	30	**0.2**	35
5	3	3	L	5.5	2377	1.4	361	1.0	167	1.7	391	**0.6**	130	0.7	159
10	2	3	L	3.9	277	3.1	167	3.2	170	3.1	161	**2.1**	85	2.5	131
10	3	3	L	226.5	14127	82.2	4281	44.0	1850	67.5	2955	**18.5**	847	20.3	951
15	2	3	L	27.5	639	11.4	223	11.8	194	12.1	228	**10.1**	157	10.0	168
15	3	3	L	885.0$^{(1)}$	10696	321.3	4929	174.5	2180	323.6	5329	**67.6**	1060	67.9	1047
5	2	3	H	0.4	121	0.4	87	0.3	51	0.3	53	**0.2**	20	**0.2**	28
5	3	3	H	4.4	1893	1.4	411	1.1	221	1.4	305	**0.4**	80	0.6	139
10	2	3	H	4.1	277	3.8	228	3.2	156	2.9	138	2.9	132	**1.9**	68
10	3	3	H	148.6	8841	51.3	2534	38.8	1653	61.1	2928	**16.8**	744	18.9	976
15	2	3	H	21.8	456	12.5	227	10.4	148	15.5	267	**7.5**	102	11.3	171
15	3	3	H	950.3$^{(1)}$	13449	291.9	4171	127.4	1662	274.5	4383	**62.1**	891	63.8	896
5	2	6	L	37.6	9903	23.8	5900	8.3	1752	19.5	4328	**5.5**	1158	5.8	1226
5	3	6	L	3600.0$^{(5)}$	402827	3590.4$^{(4)}$	457099	371.9	55557	2939.5$^{(3)}$	364231	186.5	34304	**172.0**	32258
10	2	6	L	1363.5	44425	1241.8	37814	416.3	11688	863.7	25449	**197.1**	6296	257.6	8335
15	2	6	L	3205.7	15731	2630.1	14530	1974.8	9604	2856.7	15499	**1403.6**	9819	1632.5	11431
5	2	6	H	32.3	8278	20.9	5027	6.4	1243	14.6	2979	**4.3**	856	5.3	1109
5	3	6	H	3600.0$^{(5)}$	422362	3169.3$^{(3)}$	424158	310.3	42621	2823.0$^{(2)}$	361287	**118.5**	21290	137.3	25457
10	2	6	H	1317.5$^{(1)}$	38066	1364.8$^{(1)}$	41694	359.1	10348	779.3	23235	**176.6**	5559	181.0	5369
15	2	6	H	3052.5$^{(1)}$	16450	2508.9$^{(1)}$	13658	1709.5$^{(1)}$	8658	2280.7$^{(1)}$	13287	1033.7	6739	**937.8**	6749
Average				924.4	50567	766.5	50879	278.6	7498	667.0	41375	**165.7**	4515	176.4	4835
MVIRP-OU															
5	2	3	L	0.2	2	0.2	0	0.2	0	0.2	0	**0.1**	0	**0.1**	0
5	3	3	L	0.2	5	0.2	3	0.3	0	0.4	4	**0.1**	0	**0.1**	0
10	2	3	L	6.1	336	5.2	270	5.1	225	6.3	313	**3.5**	149	4.1	182
10	3	3	L	24.6	806	16.4	483	20.9	531	25.2	673	**8.6**	171	8.9	205
15	2	3	L	18.4	248	17.8	238	20.2	247	20.9	288	**9.9**	102	12.7	142
15	3	3	L	166.6	1338	84.7	657	94.9	621	133.9	998	48.1	298	**35.1**	204
5	2	3	H	0.2	0	0.2	0	0.3	0	0.2	0	**0.0**	0	0.1	0
5	3	3	H	0.2	3	0.1	0	0.2	0	0.3	5	**0.1**	0	**0.1**	0
10	2	3	H	6.2	355	5.7	313	5.2	233	6.3	308	3.8	165	**3.4**	154
10	3	3	H	29.1	1010	17.0	544	13.6	331	19.5	507	**8.7**	192	9.8	246
15	2	3	H	18.1	252	14.8	209	23.2	307	23.9	352	12.2	146	**11.1**	122
15	3	3	H	172.1	1380	83.4	639	86.5	500	191.6	1244	**33.7**	216	38.0	242
5	2	6	L	3.6	244	3.9	259	3.4	113	3.3	193	**2.2**	86	2.4	106
5	3	6	L	1.7	153	1.6	142	1.0	29	1.1	36	**0.2**	0	0.3	1
10	2	6	L	1381.7	28207	863.9	19095	477.0	9439	853.5	17294	356.8	7667	**328.5**	7726
15	2	6	L	3548.6$^{(4)}$	13596	3008.2$^{(3)}$	11181	2182.6$^{(1)}$	9031	3116.4$^{(2)}$	13116	1382.5	6026	**1337.4**	6021
5	2	6	H	3.5	262	3.4	271	3.1	72	3.8	245	**1.9**	92	2.0	91
5	3	6	H	1.0	37	1.2	80	1.3	30	1.0	21	**0.2**	0	0.2	1
10	2	6	H	1361.0	27535	814.1	17833	508.5	10034	548.5	11546	327.8	7280	**300.1**	6740
15	2	6	H	3600.0$^{(4)}$	13558	2931.3$^{(3)}$	12738	2199.0$^{(1)}$	8523	3185.4$^{(3)}$	13095	**961.1**	4604	1628.9	6706
Average				517.1	4466	393.7	3248	282.3	2013	407.1	3012	**158.1**	1360	186.2	1444

$^{(-)}$ indicates the number of instances (out of 5) were not solved to optimality

Table 5 **Results of using different symmetry breaking constraints on the MVPRP instances not solved to optimality**

n	m	l	None		SBC0		SBC0+1		SBC0+2		SBC0+3		SBC0+4	
			Time	Node	Time	Node	Time	Node	Time	Node	Time	Node	Time	Node
MVPRP-ML														
15	3	6	98.8$^{(2)}$	8482	99.1$^{(2)}$	4562	99.6$^{(2)}$	3687	99.3$^{(2)}$	3633	**100.0**$^{(1)}$	4386	**100.0**$^{(1)}$	4131
15	2	9	98.7$^{(3)}$	3335	98.8$^{(2)}$	7405	99.0$^{(2)}$	3936	99.0$^{(2)}$	6677	**99.0**$^{(2)}$	4369	**99.0**$^{(2)}$	4810
15	3	9	95.9$^{(4)}$	3600	95.9$^{(4)}$	3423	95.8$^{(4)}$	2780	95.9$^{(4)}$	2705	**96.3**$^{(4)}$	3733	**96.3**$^{(4)}$	3867
MVPRP-OU														
15	9	3	96.9$^{(4)}$	3600	96.8$^{(4)}$	4227	96.9$^{(4)}$	3717	96.6$^{(4)}$	4025	**97.2**$^{(4)}$	4016	97.0$^{(4)}$	3978

$^{(-)}$ indicates the number of instances (out of 4) were not solved to optimality

The results clearly show the additional benefits of using the SBCs compared to the SBCs generated solely by CPLEX. The original formulation without any SBC provides the worst

Table 6 Results of using different symmetry breaking constraints on the MVIRP instances not solved to optimality

n	m	l	h_i	None		SBC0		SBC0+1		SBC0+2		SBC0+3		SBC0+4	
			Time	Node	Time	Node	Time	Node	Time	Node	Time	Node	Time	Node	
MVIRP-ML															
10	3	6	L	$86.1^{(5)}$	34559	$87.8^{(5)}$	36108	$88.9^{(5)}$	32218	$88.0^{(5)}$	33413	**94.3**$^{(4)}$	46785	$93.5^{(5)}$	47282
15	3	6	L	$83.4^{(5)}$	6098	$84.2^{(5)}$	6628	$85.4^{(5)}$	6154	$83.9^{(5)}$	6240	$90.7^{(5)}$	8326	**91.5**$^{(5)}$	8144
10	3	6	H	$91.2^{(5)}$	34044	$92.5^{(5)}$	34655	$93.5^{(5)}$	31797	$92.7^{(5)}$	34347	$96.3^{(3)}$	46628	**96.4**$^{(4)}$	44491
15	3	6	H	$90.9^{(5)}$	5984	$91.3^{(5)}$	6479	$92.2^{(5)}$	5540	$91.6^{(5)}$	6353	**95.9**$^{(5)}$	8344	$95.2^{(5)}$	8227
Average				87.9	20171	89.0	20967	90.0	18927	89.1	20088	**94.3**	27521	94.1	27036
MVIRP-OU															
10	3	6	L	$87.7^{(5)}$	28023	$89.1^{(5)}$	27507	$90.5^{(5)}$	25434	$88.1^{(5)}$	26042	**96.1**$^{(3)}$	24165	$95.5^{(3)}$	24035
15	3	6	L	$79.2^{(5)}$	6151	$80.9^{(5)}$	6431	$80.2^{(5)}$	5506	$80.3^{(5)}$	7085	$86.2^{(5)}$	5544	**86.5**$^{(5)}$	6319
10	3	6	H	$92.1^{(5)}$	28007	$93.1^{(5)}$	28331	$93.2^{(5)}$	23401	$92.2^{(5)}$	26769	$97.0^{(3)}$	25265	**97.3**$^{(3)}$	26527
15	3	6	H	$87.4^{(5)}$	6149	$87.9^{(5)}$	6542	$88.2^{(5)}$	5762	$87.8^{(5)}$	6506	**91.7**$^{(5)}$	6176	$90.9^{(5)}$	5437
Average				86.6	17083	87.8	17203	88.0	15026	87.1	16600	**92.7**	15287	92.5	15580

$^{(-)}$ indicates the number of instances (out of 5) were not solved to optimality

Table 7 Summary of the time reduction factors by using each SBC strategy on instances solved to optimality

	SBC0	SBC0+1	SBC0+2	SBC0+3	SBC0+4
MVPRP-ML					
Min	1.16	1.49	1.06	**1.52**	1.35
Max	4.29	5.79	5.45	**11.11**	9.07
Avg	2.12	3.12	2.28	**4.25**	3.89
MVPRP-OU					
Min	0.97	0.94	0.77	0.92	**1.00**
Max	1.46	**6.08**	2.15	5.52	5.38
Avg	1.17	1.70	1.15	**2.03**	2.01
MVIRP-ML					
Min	1.04	1.21	1.16	1.44	**1.58**
Max	3.95	7.46	3.46	**15.31**	14.89
Avg	2.31	3.38	2.29	**6.86**	6.33
MVIRP-OU					
Min	0.83	0.68	0.49	**1.49**	1.44
Max	2.06	2.15	1.49	**5.11**	4.75
Avg	1.32	1.20	0.91	**2.88**	2.79

results and adding SBC0 could generally improve the computing times and reduce the number of nodes in the branch-and-bound tree. The combination of SBC0 together with one of the other SBCs could further speed up the solving process, except for the SBC2 where some results are worse than using CPLEX cuts alone. The cut strategies SBC0+SBC3 and SBC0+SBC4 provide good results, but SBC0+SBC3 is slightly better in overall. The average time factor reductions obtained by using SBC0+SBC3 within the maximum computing time limit of one hour are 4.25, 2.03, 6.86 and 2.88 for the MVPRP-ML, MVPRP-OU, MVIRP-ML and MVPRP-OU instances, respectively. By adding SBC0+SBC3 to the $F(ML)|k$ and $F(OU)|k$ formulations, it could also solve 31 instances that could not be solved to optimality using the formulations alone. We then consider the formulations $F(ML)|k$ and $F(OU)|k$ with SBC0+SBC3 in the remaining computational experiments.

6.3. Effect of Valid Inequalities for the Non-Vehicle Index Formulations.

In this section, we test the effect of the valid inequalities and the branch and cut strategy that we implemented for the formulation $F(ML)|nk$ and $F(OU)|nk$. First, we test the effects of the valid inequalities on the lower bounds at the root node of the branch-and-bound tree. To avoid misinterpretation due to the impact of the CPLEX's cuts, we conduct the experiments without the CPLEX's cuts. The average lower bounds are shown in Table 8 and 9. The numbers presented are equal to the average lower bounds at the root node compared to the optimal solutions or the best upper bounds if the instances that were not solved to optimality. Each column shows the results of using each cut presented in Section (3.2) where the column None and All present the results without using any additional cuts and all cuts together, respectively.

Table 8 Effects of the valid inequalities on lower bounds at the root node on the MVPRP instances.

n	m	l	MVPRP-ML						MVPRP-OU					
			None	(59)	(61)	(62)	(64)	All	None	(60)	(61)	(63)	(64)	All
10	2	3	98.0	98.0	98.8	98.0	98.9	**100.0**	96.9	97.1	97.2	97.2	96.9	**97.3**
10	3	3	97.2	96.9	98.2	97.2	97.5	**98.7**	96.6	**96.8**	96.6	**96.8**	96.6	**96.8**
15	2	3	96.4	96.4	**98.2**	96.5	96.4	**98.2**	88.0	91.1	88.3	89.0	88.0	**91.8**
15	3	3	95.1	95.2	97.1	95.4	95.7	**97.6**	88.4	90.9	88.3	89.4	88.2	**91.3**
10	2	6	89.5	89.5	91.3	89.8	89.7	**91.6**	94.5	94.5	**94.8**	94.5	94.5	**94.8**
10	3	6	88.9	88.9	90.4	89.1	89.5	**91.0**	94.4	94.4	94.6	94.4	94.5	**94.7**
15	2	6	90.4	90.5	92.1	90.6	90.6	**92.4**	94.5	94.6	94.9	94.6	94.6	**94.9**
15	3	6	89.2	89.5	90.9	90.1	89.6	**91.5**	93.9	94.0	**94.3**	94.1	93.9	**94.3**
10	2	9	95.0	95.0	96.2	95.1	95.3	**96.5**	96.2	96.2	97.2	96.5	96.3	**97.3**
10	3	9	94.3	94.3	95.4	94.5	94.5	**96.3**	95.5	95.6	96.3	96.0	95.7	**96.6**
15	2	9	93.3	93.3	94.6	93.4	93.5	**94.9**	94.9	95.2	96.1	95.3	95.1	**96.5**
15	3	9	92.0	91.8	92.8	92.4	92.5	**93.7**	94.0	94.1	94.8	94.3	94.3	**95.1**
Average			93.3	93.3	94.7	93.5	93.7	**95.2**	94.0	94.5	94.5	94.4	94.0	**95.1**

The results show that adding all the cuts together generally provide the best lower bounds and it has more effect on larger instance sizes. To see the effects of using these cuts on our exact algorithms, we perform the tests and the results are shown in Table 10 and Table 11. The results of the branch-and-cut without and with additional valid inequalities in Section (3.2) are shown in Columns $F(ML)|nk$, $F(OU)|nk$ and Columns $F(ML)|nk^+$, $F(OU)|nk^+$, respectively.

We can see that applying all the inequalities in Section 3.2 could provide significant improvements in the branch-and-cut procedure for both formulations $F(ML)|nk$ and $F(OU)|nk$ in terms of lower bounds (both at the root node and final lower bounds), computing times and the number of nodes in branch and bound tree. We then use the formulations $F(ML)|nk^+$ and $F(OU)|nk^+$ in the remaining of the computational test.

Table 9 **Effects of the valid inequalities on lower bounds at the root node on the MVIRP instances.**

n	m	l	h_i	MVPRP-ML						MVPRP-OU					
				None	(59)	(61)	(62)	(64)	All	None	(60)	(61)	(63)	(64)	All
5	2	3	L	86.6	86.2	87.7	86.8	92.9	**93.9**	92.0	92.1	93.7	92.6	94.1	**95.8**
5	3	3	L	83.7	83.2	84.3	84.7	88.2	**89.5**	86.0	86.6	87.1	86.2	88.3	**88.8**
10	2	3	L	86.1	86.1	86.9	86.0	90.7	**91.3**	82.2	81.0	84.7	82.7	83.3	**85.2**
10	3	3	L	79.9	78.8	83.7	81.0	88.1	**89.1**	83.5	82.3	87.0	85.8	85.5	**89.5**
15	2	3	L	86.6	86.3	89.4	86.5	90.1	**92.1**	81.5	80.2	**86.5**	81.6	82.1	85.4
15	3	3	L	81.4	81.4	86.2	81.8	85.4	**88.9**	81.3	80.7	**87.7**	82.0	81.4	86.9
5	2	3	H	91.5	91.4	92.3	91.7	95.5	**96.0**	94.6	94.6	95.7	95.0	96.1	**97.2**
5	3	3	H	88.6	88.6	89.4	89.0	92.0	**93.0**	90.1	90.4	90.9	90.2	91.8	**92.2**
10	2	3	H	93.3	93.3	93.8	93.3	**95.7**	**95.7**	90.7	90.1	92.0	91.0	91.2	**92.4**
10	3	3	H	89.3	88.8	91.4	89.9	93.2	**94.2**	90.5	89.7	92.6	92.0	91.4	**94.3**
15	2	3	H	94.1	93.9	95.5	93.9	95.4	**96.6**	91.2	90.6	**93.6**	91.2	91.4	93.1
15	3	3	H	91.1	91.0	93.4	91.2	92.5	**94.5**	90.2	90.1	**93.8**	90.9	90.6	93.6
5	2	6	L	75.6	75.1	77.3	79.5	82.4	**83.1**	85.1	83.5	86.4	86.2	86.3	**87.8**
5	3	6	L	83.0	82.3	83.8	83.9	**87.3**	86.4	84.5	85.4	84.5	85.0	85.8	**86.6**
10	2	6	L	73.3	72.8	74.9	75.5	78.2	**79.5**	80.0	80.1	81.9	82.5	84.1	**86.8**
10	3	6	L	72.8	73.2	75.6	76.9	80.0	**82.7**	78.1	76.6	80.8	81.5	80.9	**83.2**
15	2	6	L	71.9	72.2	75.7	73.3	74.6	**79.2**	81.4	81.2	85.2	82.4	82.6	**87.6**
15	3	6	L	69.0	69.7	72.5	71.6	71.1	**77.3**	73.7	74.2	77.1	75.6	74.5	**79.4**
5	2	6	H	85.7	84.7	85.8	87.3	88.0	**88.8**	90.3	89.3	91.2	90.9	90.8	**92.1**
5	3	6	H	88.4	88.8	88.8	89.0	**91.0**	90.3	88.9	89.5	88.9	89.1	89.7	**90.3**
10	2	6	H	84.7	83.7	85.3	86.4	87.3	**88.2**	87.8	87.8	88.9	89.3	90.4	**91.7**
10	3	6	H	81.9	83.2	85.0	85.9	86.8	**89.1**	84.8	84.5	87.3	88.0	86.9	**89.4**
15	2	6	H	85.6	85.8	87.5	86.5	86.9	**89.2**	90.1	90.0	92.2	90.9	90.8	**93.4**
15	3	6	H	83.2	83.3	85.1	83.9	84.3	**86.7**	84.2	84.0	85.9	85.1	84.6	**87.7**
Average				79.6	79.6	81.4	81.6	83.1	**85.0**	84.1	83.8	85.8	85.6	85.6	**88.0**

Table 10 **Effects of the valid inequalities on the branch-and-cut algorithm on MVPRP instances.**

n	l	m	MVPRP-ML						MVPRP-OU					
			$F(ML)\|nk$			$F(ML)\|nk^+$			$F(OU)\|nk$			$F(OU)\|nk^+$		
			%LB	Time	Node	%LB	Time	Node	%LB	Time	Node	%LB	Time	Node
10	2	3	100.0	0.3	12	100.0	**0.1**	0	100.0	**0.3**	13	100.0	**0.3**	8
10	3	3	100.0	0.6	71	100.0	**0.4**	17	100.0	0.4	13	100.0	**0.3**	5
15	2	3	100.0	40.8	4203	100.0	**15.6**	918	100.0	329.8	17432	100.0	**139.8**	10223
15	3	3	100.0	275.0	30747	100.0	**58.9**	3039	100.0	967.3	39767	100.0	**428.6**	28676
10	2	6	100.0	6.9	716	100.0	**1.0**	6	100.0	0.5	5	100.0	**0.3**	0
10	3	6	100.0	40.6	4687	100.0	**19.8**	1506	100.0	**0.6**	5	100.0	0.7	6
15	2	6	$99.3^{(4)}$	3600.0	100687	**99.8**$^{(1)}$	1048.8	16217	100.0	75.3	2208	100.0	**53.1**	1333
15	3	6	$98.9^{(4)}$	3600.0	77402	**99.2**$^{(4)}$	3600.0	64402	100.0	322.4	10196	100.0	**117.8**	2802
10	2	9	100.0	182.4	10872	100.0	**53.5**	1907	100.0	15.7	566	100.0	**9.0**	200
10	3	9	$99.7^{(3)}$	2767.0	125261	100.0	**1454.5**	77948	100.0	58.3	3045	100.0	**43.6**	1789
15	2	9	$98.8^{(4)}$	3600.0	33291	**99.3**$^{(4)}$	3600.0	27380	$98.9^{(4)}$	3600.0	45068	**99.1**$^{(3)}$	3600.0	34038
15	3	9	$97.9^{(4)}$	3600.0	32155	**98.2**$^{(4)}$	3600.0	23677	**98.0**$^{(4)}$	3600.0	33266	$97.9^{(4)}$	3600.0	23592

$^{(-)}$ indicates the number of instances (out of 4) were not solved to optimality

6.4. Quality and Effectiveness of Initial Upper Bounds using the Op-ALNS.

The average results of the Op-ALNS on MVPRP and MVIRP instances are provided in Table 12 and 12, respectively. Column %Diff indicates the percent difference of the total costs obtained by the Op-ALNS from the optimal objective value or the best upper bound if the instances were not solved to optimality.

The Op-ALNS could generally provide high quality solutions on the MVPRP instances where the differences from the optimal solutions or best upper bounds are 1.2% and 0.8% for the MVPRP-ML and MVPRP-OU, respectively. The results on the MVIRP are not as

Table 11 Effects of the valid inequalities on the branch-and-cut algorithm on MVIRP instances.

n	l	m	h_i	MVIRP-ML						MVIRP-OU					
				$F(ML)\|nk$			$F(ML)\|nk^+$			$F(OU)\|nk$			$F(OU)\|nk^+$		
				%LB	Time	Node	%LB	Time	Node	%LB	Time	Node	%LB	Time	Node
5	2	3	L	100.0	0.1	57	100.0	**0.1**	45	100.0	**0.1**	8	100.0	**0.1**	5
5	3	3	L	100.0	0.3	197	100.0	**0.2**	102	100.0	**0.1**	16	100.0	**0.1**	9
10	2	3	L	100.0	5.8	1106	100.0	**4.8**	757	100.0	5.0	1069	100.0	**3.7**	445
10	3	3	L	100.0	41.8	8666	100.0	**28.9**	4358	100.0	9.3	1779	100.0	**6.0**	727
15	2	3	L	100.0	150.6	11236	100.0	**54.0**	3278	100.0	241.3	19135	100.0	**57.9**	2525
15	3	3	L	98.9$^{(1)}$	1477.5	76621	99.6$^{(1)}$	**991.9**	51154	100.0	329.7	20756	100.0	**55.7**	2248
5	2	3	H	100.0	0.1	44	100.0	**0.1**	24	100.0	**0.1**	11	100.0	**0.1**	1
5	3	3	H	100.0	0.2	153	100.0	**0.1**	61	100.0	**0.1**	23	100.0	**0.1**	3
10	2	3	H	100.0	**6.8**	1466	100.0	8.9	1569	100.0	6.0	1257	100.0	**3.5**	401
10	3	3	H	100.0	45.9	10468	100.0	**24.6**	3576	100.0	11.1	2208	100.0	**5.2**	657
15	2	3	H	100.0	149.6	10385	100.0	**53.9**	3402	100.0	391.3	24411	100.0	**70.2**	3662
15	3	3	H	99.4$^{(1)}$	1251.8	73544	100.0	**649.1**	38901	100.0	260.0	15888	100.0	**63.9**	2646
5	2	6	L	100.0	23.7	11083	100.0	**22.7**	9483	100.0	**3.3**	1296	100.0	6.2	2331
5	3	6	L	100.0	373.2	195135	100.0	**278.7**	128430	100.0	**3.1**	1056	100.0	4.2	1614
10	2	6	L	97.1$^{(2)}$	1810.9	131482	**97.9**$^{(2)}$	1671.7	98941	97.5$^{(2)}$	2219.4	155946	**98.8**$^{(1)}$	1898.7	103737
10	3	6	L	95.1$^{(4)}$	3542.6	231848	**95.4**$^{(4)}$	3599.8	194042	94.9$^{(4)}$	3500.1	216166	**95.5**$^{(4)}$	3231.2	129350
15	2	6	L	92.5$^{(5)}$	3600.0	64045	**95.4**$^{(5)}$	3600.0	52811	91.5$^{(5)}$	3600.0	56616	**95.1**$^{(5)}$	3600.0	44198
15	3	6	L	90.4$^{(5)}$	3600.0	56366	**92.1**$^{(5)}$	3600.0	41742	87.0$^{(5)}$	3600.0	53605	**89.2**$^{(5)}$	3600.0	35516
5	2	6	H	100.0	**23.7**	10590	100.0	23.8	9577	100.0	**3.8**	1532	100.0	5.9	2203
5	3	6	H	100.0	275.1	145895	100.0	**232.7**	101098	100.0	**2.7**	988	100.0	3.7	1377
10	2	6	H	98.3$^{(2)}$	1738.0	124051	**98.9**$^{(2)}$	1590.8	92794	98.4$^{(2)}$	2100.4	144600	**99.3**$^{(2)}$	2072.4	108765
10	3	6	H	96.9$^{(4)}$	3196.0	221673	**97.0**$^{(4)}$	3535.1	191522	**96.8**$^{(3)}$	3396.7	226370	96.5$^{(4)}$	3510.0	158959
15	2	6	H	96.1$^{(5)}$	3600.0	61850	**97.9**$^{(5)}$	3600.0	56568	95.5$^{(5)}$	3600.0	61377	**97.5**$^{(5)}$	3600.0	43490
15	3	6	H	94.8$^{(5)}$	3600.0	57869	**95.6**$^{(5)}$	3600.0	42722	91.7$^{(5)}$	3600.0	53524	**92.9**$^{(5)}$	3600.0	34319

$^{(-)}$ indicates the number of instances (out of 5) were not solved to optimality

Table 12 Average results for Op-ALNS on MVPRP instances.

n	l	m	MVPRP-ML		MVPRP-OU	
			Op-ALNS		Op-ALNS	
			%Diff	Time	%Diff	Time
10	2	3	0.4	4.6	0.0	4.2
10	3	3	1.1	4.3	0.0	4.4
15	2	3	0.9	6.6	2.0	5.8
15	3	3	1.0	6.6	1.0	6.7
10	2	6	0.6	7.3	0.1	9.5
10	3	6	0.4	8.4	0.1	14.0
15	2	6	1.0	13.8	0.4	14.1
15	3	6	1.6	14.0	0.7	17.6
10	2	9	1.8	13.8	1.4	18.4
10	3	9	1.8	12.9	1.3	26.7
15	2	9	1.6	24.1	1.2	28.2
15	3	9	1.6	25.7	1.6	55.7
Average			1.2	11.8	0.8	17.1

good as the MVPRP but the Op-ALNS could still provide good quality solutions within a few seconds.

We test the effect of setting the initial upper bounds produced by the Op-ALNS and the results are provided in Table 14 and 15. Column %TR designates the percentage reduction on average computing times on instances were solved to optimality (bracket is used if the average computing time increases). The results show that the average computing times on the MVPRP instances could generally reduce by setting the initial upper bounds, while there is no significant effect on the MVIRP instances. On the instances not solved to

Table 13 **Average results for Op-ALNS on MVIRP instances.**

n	l	m	h_i	MVIRP-ML		MVIRP-OU	
				Op-ALNS		Op-ALNS	
				%Diff	Time	%Diff	Time
5	2	3	L	0.2	3.5	2.2	3.4
5	3	3	L	1.4	3.7	0.8	3.8
10	2	3	L	4.5	5.5	8.3	6.6
10	3	3	L	4.4	6.0	6.7	7.4
15	2	3	L	4.7	7.9	8.8	8.1
15	3	3	L	7.8	10.3	13.0	13.7
5	2	3	H	1.0	3.2	0.0	3.3
5	3	3	H	1.1	3.7	0.5	4.0
10	2	3	H	2.2	5.3	2.0	6.2
10	3	3	H	2.8	6.0	2.2	7.1
15	2	3	H	2.6	7.8	5.6	9.4
15	3	3	H	2.6	8.5	4.9	13.9
5	2	6	L	4.0	5.7	1.6	5.9
5	3	6	L	3.8	6.8	1.7	7.6
10	2	6	L	6.6	9.7	8.7	9.7
10	3	6	L	6.6	11.4	7.7	12.0
15	2	6	L	6.0	18.0	12.1	16.5
15	3	6	L	7.8	18.8	10.7	20.6
5	2	6	H	2.6	5.7	2.3	5.7
5	3	6	H	2.7	6.6	1.0	7.5
10	2	6	H	3.0	9.8	6.0	9.1
10	3	6	H	4.1	11.4	6.4	11.6
15	2	6	H	2.5	17.6	6.8	15.7
15	3	6	H	3.1	17.3	5.4	19.7
Average				3.7	8.8	5.2	9.5

optimality, One possible reason is that the quality of the upper bounds on the MVIRP instances is not as good as the MVPRP instances.

Table 14 **Effect of initial upper bounds on MVPRP instances.**

n	l	m	MVIPRP-ML						MVPRP-OU					
			$F(ML)\|k$			$F(ML)\|nk$			$F(OU)\|k$			$F(OU)\|nk$		
			%LB	Time	%TR	%LB	Time	%TR	%LB	Time	%TR	%LB	Time	%TR
10	2	3	100.0	0.2	20.1	100.0	**0.1**	6.0	100.0	0.5	35.3	100.0	**0.2**	43.2
10	3	3	100.0	0.8	13.0	100.0	**0.3**	25.1	100.0	0.8	57.4	100.0	**0.2**	36.5
15	2	3	100.0	**3.8**	18.6	100.0	8.9	42.6	100.0	**25.2**	18.7	100.0	86.1	38.4
15	3	3	100.0	41.7	22.5	100.0	**37.3**	36.7	100.0	**26.7**	10.8	100.0	338.2	21.1
10	2	6	100.0	2.7	(5.7)	100.0	**0.8**	16.5	100.0	0.7	29.0	100.0	**0.3**	1.1
10	3	6	100.0	18.5	(0.0)	100.0	**17.3**	12.2	100.0	2.1	27.5	100.0	**0.3**	54.1
15	2	6	100.0	**165.0**	(1.8)	99.7$^{(1)}$	1081.1	-	100.0	**13.7**	21.4	100.0	60.6	(14.3)
15	3	6	**99.9**$^{(1)}$	1818.3	-	99.2$^{(4)}$	3600.0	-	100.0	**123.1**	28.2	100.0	126.1	(7.0)
10	2	9	100.0	**31.5**	(1.8)	100.0	34.5	35.4	100.0	23.2	(30.1)	100.0	**5.6**	37.6
10	3	9	100.0	**664.0**	28.9	100.0	1336.5	8.1	100.0	115.6	23.7	100.0	**48.1**	(10.2)
15	2	9	99.1$^{(2)}$	2167.2	-	**99.3**$^{(4)}$	3600.0	-	**99.8**$^{(1)}$	1529.9	-	99.1$^{(4)}$	3600.0	-
15	3	9	96.3$^{(4)}$	3600.0	-	**98.1**$^{(4)}$	3600.0	-	97.3$^{(4)}$	3600.0	-	**97.8**$^{(4)}$	3600.0	-
Average			99.6	709.5	10.4	**99.7**	1109.7	22.8	**99.8**	455.1	22.2	99.7	655.5	20.0

$^{(-)}$ indicates the number of instances (out of 4) were not solved to optimality

Table 14 and 15 also provide insights about the performance of the vehicle index and non-vehicle index formulations. The vehicle index formulations $F(ML)|k$ and $F(OU)|k$ could find more optimal solutions and generally spend less computing times on average compared to the non vehicle-index formulations $F(ML)|nk$ and $F(OU)|nk$. There is no

Table 15 Effect of initial upper bounds on MVIRP instances.

n	l	m	h_i	MVIRP-ML						MVIRP-OU					
				$F(ML)\|k$			$F(ML)\|nk$			$F(OU)\|k$			$F(OU)\|nk$		
				%LB	Time	%TR	%LB	Time	%TR	%LB	Time	%TR	%LB	Time	%TR
5	2	3	L	100.0	0.2	(0.9)	100.0	**0.1**	0.0	100.0	**0.1**	0.0	100.0	**0.1**	0.0
5	3	3	L	100.0	0.5	16.2	100.0	**0.2**	0.0	100.0	**0.1**	0.0	100.0	**0.1**	0.0
10	2	3	L	100.0	**2.1**	0.1	100.0	3.1	35.4	100.0	3.4	2.9	100.0	**3.1**	16.2
10	3	3	L	100.0	**14.2**	23.2	100.0	22.5	22.1	100.0	8.1	5.8	100.0	**4.8**	20.0
15	2	3	L	100.0	**8.1**	20.1	100.0	29.4	45.6	100.0	**9.4**	5.1	100.0	31.9	44.9
15	3	3	L	100.0	**69.7**	(3.1)	99.2$^{(1)}$	1031.2	-	100.0	**31.4**	34.7	100.0	72.2	(29.6)
5	2	3	H	100.0	0.2	(9.1)	100.0	**0.1**	0.0	100.0	**0.0**	0.0	100.0	0.1	0.0
5	3	3	H	100.0	0.5	(16.0)	100.0	**0.1**	0.0	100.0	**0.1**	0.0	100.0	**0.1**	0.0
10	2	3	H	100.0	**2.5**	12.3	100.0	6.2	30.3	100.0	3.4	10.5	100.0	**3.2**	8.6
10	3	3	H	100.0	**15.9**	5.4	100.0	37.6	(52.8)	100.0	7.6	12.6	100.0	**6.0**	(15.4)
15	2	3	H	100.0	**7.2**	3.4	100.0	46.7	13.4	100.0	**8.9**	27.0	100.0	56.1	20.1
15	3	3	H	100.0	**60.7**	2.2	100.0	741.9	(14.3)	100.0	**30.4**	9.8	100.0	45.1	29.4
5	2	6	L	100.0	**5.4**	1.4	100.0	21.6	4.8	100.0	**2.0**	9.1	100.0	5.7	8.1
5	3	6	L	100.0	**187.3**	(0.4)	100.0	342.7	(23.0)	100.0	**0.2**	0.0	100.0	3.9	7.1
10	2	6	L	100.0	**189.2**	4.0	98.2$^{(2)}$	1618.2	-	100.0	**235.7**	33.9	99.0$^{(1)}$	1761.2	-
10	3	6	L	94.6$^{(4)}$	3504.4	-	**95.6**$^{(4)}$	3281.6	-	**96.6**$^{(3)}$	2848.4	-	95.8$^{(5)}$	3600.0	-
15	2	6	L	**99.8**$^{(1)}$	1319.1	-	95.6$^{(5)}$	3600.0	-	100.0	**1287.5**	6.9	95.3$^{(5)}$	3600.0	-
15	3	6	L	91.7$^{(5)}$	3600.0	-	**92.2**$^{(5)}$	3600.0	-	86.5$^{(5)}$	3600.0	-	**89.4**$^{(5)}$	3600.0	-
5	2	6	H	100.0	**4.0**	7.8	100.0	22.0	7.6	100.0	**1.9**	0.0	100.0	5.9	0.0
5	3	6	H	100.0	**108.0**	8.9	100.0	236.0	(1.4)	100.0	**0.2**	0.0	100.0	4.5	(21.6)
10	2	6	H	100.0	**158.5**	10.3	98.8$^{(2)}$	1588.0	-	100.0	**327.0**	0.2	99.5$^{(1)}$	1776.2	-
10	3	6	H	96.7$^{(3)}$	3184.2	-	**97.2**$^{(5)}$	3600.0	-	**97.5**$^{(3)}$	2959.3	-	97.0$^{(4)}$	3443.2	-
15	2	6	H	100.0	**1282.7**	(24.1)	97.9$^{(5)}$	3600.0	-	100.0	**1364.1**	(41.9)	97.5$^{(5)}$	3600.0	-
15	3	6	H	**95.8**$^{(5)}$	3600.0	-	95.6$^{(5)}$	3600.0	-	91.7$^{(5)}$	3600.0	-	**92.9**$^{(5)}$	3600.0	-
Average				**99.1**	721.9	3.3	98.8	1126.2	4.5	**98.8**	680.4	5.8	98.6	1051.0	5.5

$^{(-)}$ indicates the number of instances (out of 5) were not solved to optimality

significant difference on the average optimality gaps of the two formulation schemes within the computational time limit of one hour.

6.5. Results on Larger Instances.

***The discussion and full results will be provided.** In overall, it seems that the non-vehicle index formulations are less sensitive to the size of the problems and could produce better lower bounds within one hour.

Table 16 Average results on larger MVIRP instances.

n	m	l	MVIPRP-ML						MVPRP-OU					
			$F(ML)\|k$		$F(ML)\|nk$		Op-ALNS		$F(OU)\|k$		$F(OU)\|nk$		Op-ALNS	
			%LB	Time	%LB	Time	%DIFF	Time	%LB	Time	%LB	Time	%DIFF	Time
20	3	3	100.0	89.2	100.0	**83.1**	3.4	10.0	100.0	1326.3	TBD		1.3	8.4
25	3	3	100.0	**172.2**	100.0	632.8	2.0	14.3	96.9	3600.0			1.8	14.3
30	4	3	99.4	3143.6	**99.6**	2973.2	2.2	28.1	93.9	3600.0			0.0	23.3
35	4	3	98.0	3600.0	**98.6**	3600.0	2.0	43.0	93.8	3600.0			0.0	37.7
40	5	3	95.9	3600.0	**96.9**	3600.0	0.2	67.3	93.0	3600.0			0.0	52.5
20	3	6	100.0	**1782.6**	99.3	1811.8	1.5	20.6	100.0	520.0			0.5	32.7
25	3	6	98.6	2178.5	**98.7**	2527.6	1.0	34.2	97.8	3600.0			0.8	48.4
30	4	6	96.1	3600.0	**97.8**	3600.0	0.4	59.3	97.2	3600.0			0.0	90.3
20	3	9	97.3	3600.0			0.1	39.2	98.2	3600.0			0.1	70.4
Average			98.4	2418.5			1.4	35.1	96.8	3005.1			0.5	42.0

6.6. Performances of the Exact Algorithms on Single Vehicle Instances.

***The results and the discussion on this section will be updated.** In this section, we test the performance of our branch-and-cut algorithms on the single vehicle PRP and IRP in literature. Since the SBCs are dropped when $m < 2$, the formulations $F(ML)|k$ and $F(OU)|k$ become equivalent to the formulation proposed by Archetti et al. (2007, 2011b) and Solyalı and Süral (2011), respectively. We add a few remarks on the two formulation schemes on the single vehicle case. Without additional valid inequalities, the constraints in the $F(ML)|k$ and $F(OU)|k$ that do not appear in the $F(ML)|nk$ and $F(OU)|nk$ are constraints (7) for the ML policy or (44) for the OU policy and constraints (11), while the constraints (28) or (56) are used in place of these two constraints for the ML and OU policy, respectively. With the valid inequalities in Section 3.2 for the single vehicle case, the inequalities (59), (60) and (61) are equilvalent to (7), (44) and (11), respectively. Thus, the formulations $F(ML)|nk$ and $F(OU)|nk$ become equilvalent to $F(ML)|k$ and $F(OU)|k$ plus additional valid inequalities.

We test the single vehicle using the MVPRP and MVIRP instances in the previous section. For the single vehicle PRP, we simply set the number of vehicles to one, and use the combined vehicle capacity, i.e., $Q = Q_s = mQ_m$, where Q_s and Q_m are the vehicle capacity used in the single and multiple vehicle instances, respectively. For the instance sizes that have different number of vehicles (i.e., instances with $n = 10, 15$, we use the Q_m from the instances with lower number of vehicles to calculate the capacity Q_s. For the single vehicle IRP, the instances are actually the original single vehicle IRP instances presented in Archetti et al. (2007). The results on the single vehicle PRP and IRP are shown in Table 17 and 18, respectively.

We can see that, compared to the multi-vehicle instances, the single vehicle instances are much easier to solve compared to the multiple vehicle case. There is no significant difference in the performance of using different formulations.

We also test the performance of the algorithm on the single vehicle PRP instances with uncapacitated production in Archetti et al. (2011b) which consists of 480 PRP instances with 14 customers and 6 periods. Since the instances were designed to test a heuristic, several different parameter settings in terms of inventory, production and transportation

Table 17 Average results on the single vehicle PRP instances (results are not updated).

n	l	$F(ML)\|k$			$F(ML)\|nk$			$F(OU)\|k$			$F(OU)\|nk$		
		Time	*Nodes*	*Cuts*	*Time*	*Nodes*	*Cuts*	*Time*	*Nodes*	*Cuts*	*Time*	*Nodes*	*Cuts*
10	3	**0.1**	5	10	**0.1**	5	36	**0.2**	1	14	**0.2**	4	19
10	6	**0.6**	3	52	**0.6**	4	54	**0.3**	1	16	**0.3**	1	22
10	9	**1.3**	18	83	1.7	27	97	**1.0**	11	52	1.4	7	68
15	3	**0.3**	2	34	**0.3**	2	62	**1.3**	47	90	1.5	44	104
15	6	**2.6**	5	147	3.8	10	141	**0.4**	1	17	0.6	1	19
15	9	31.0	222	410	**29.2**	219	406	**12.0**	134	173	18.1	168	234
20	3	**0.5**	8	45	1.3	264	219	**1.3**	30	65	1.5	34	61
20	6	**4.4**	2	171	6.5	3	181	**2.3**	57	39	4.1	133	187
20	9	**22.4**	87	265	29.0	77	300	13.5	42	167	20.7	44	217
25	3	**1.0**	139	40	1.6	28	96	15.2	243	284	**11.7**	175	279
25	6	**14.9**	5	265	23.6	3	286	13.1	35	149	**10.5**	22	111
25	9	558.0	976	859	**386.3**	765	811	**69.2**	146	316	86.8	130	382
30	3	**9.2**	12	195	9.5	13	198	**36.5**	253	478	38.9	263	611
30	6	**86.6**	112	575	111.9	159	546	**3.3**	1	49	**3.3**	1	51
30	9	1095.9	652	1676	**1062.1**	641	1504	**127.7**	55	544	170.5	51	617
35	3	**10.7**	16	159	18.7	9	210	268.5	1283	1112	**218.5**	1141	1200
35	6	**130.3**	126	481	153.0	149	502	**7.0**	10	74	9.3	167	165
35	9	1224.2$^{(1)}$	363	1697	1413.1$^{(1)}$	294	1804	**247.2**	75	683	411.0	161	752
40	3	**20.1**	5	161	27.5	5	197	162.7	489	712	**145.1**	488	758
40	6	**110.2**	3	461	146.6	4	425	**63.3**	23	238	87.6	82	301
40	9	1426.4$^{(1)}$	213	1660	1366.5$^{(1)}$	165	1635	**817.9**	219	950	1393.5	330	1342

$^{(-)}$ indicates the number of instances not solved to optimality

Table 18 Average results on the single vehicle PRP instances (results are not updated).

n	l	Class	$F(ML)\|k$			$F(ML)\|nk$			$F(OU)\|k$			$F(OU)\|nk$		
			Time	Node	Cut	Time	Node	Cut	Time	Node	Cut	Time	Node	Cut
5	3	Low	0.0	13	6	**0.0**	10	16	0.0	0	8	**0.0**	0	10
10	3	Low	0.2	0	21	**0.1**	3	19	0.5	21	62	**0.5**	14	65
15	3	Low	**0.4**	11	30	0.7	257	189	**1.7**	48	109	2.0	39	125
20	3	Low	**1.9**	37	86	2.3	43	84	**5.0**	111	226	6.8	56	268
25	3	Low	4.5	223	99	**3.8**	243	264	**9.8**	61	261	17.4	124	382
30	3	Low	**14.0**	163	159	30.8	1090	805	**28.6**	97	502	29.9	70	456
35	3	Low	**4.2**	32	59	4.4	8	62	**67.0**	171	602	107.1	168	740
40	3	Low	**32.3**	118	174	40.9	289	411	**119.9**	132	775	181.9	198	874
45	3	Low	**65.9**	62	237	67.1	99	202	272.2	349	1049	**219.2**	172	715
50	3	Low	239.9	121	590	**184.4**	111	407	677.0	243	1671	**641.1**	206	1530
5	3	High	0.0	15	4	**0.0**	15	18	**0.0**	1	9	**0.0**	0	11
10	3	High	0.2	3	26	**0.1**	4	20	**0.5**	17	56	0.7	17	72
15	3	High	0.6	38	38	**0.5**	43	65	**1.8**	52	127	2.2	54	145
20	3	High	**2.0**	30	85	2.6	33	82	**5.0**	57	243	5.5	62	224
25	3	High	**3.6**	182	83	**3.6**	171	204	**9.7**	51	296	19.0	66	388
30	3	High	**12.6**	130	153	18.8	344	363	**31.4**	103	540	37.9	138	540
35	3	High	**4.0**	16	73	7.0	73	106	**61.2**	148	620	76.1	118	611
40	3	High	**31.5**	55	213	32.7	698	722	**128.4**	148	804	134.3	128	775
45	3	High	56.1	236	230	**48.2**	110	225	**188.4**	170	808	236.2	181	816
50	3	High	194.4	132	499	**190.4**	144	495	664.1	267	1666	**484.8**	175	1183
5	6	Low	**0.2**	28	20	**0.2**	22	27	**0.4**	25	34	0.7	32	46
10	6	Low	1.4	36	82	**1.1**	15	76	**2.3**	123	132	2.6	78	144
15	6	Low	5.6	76	189	**5.0**	50	187	**6.0**	67	256	6.9	78	246
20	6	Low	37.4	293	469	**31.5**	266	415	**58.6**	511	688	68.0	628	772
25	6	Low	**48.7**	175	429	56.8	203	495	83.5	259	705	**65.0**	140	649
30	6	Low	283.5	691	985	**280.0**	780	1101	**247.1**	331	1287	259.0	385	1314
5	6	High	**0.2**	16	20	**0.2**	15	27	**0.3**	20	29	0.5	30	43
10	6	High	1.1	23	75	**1.0**	11	67	**2.1**	60	137	2.6	72	137
15	6	High	**5.0**	62	191	6.0	65	215	**5.7**	67	237	6.3	59	250
20	6	High	**18.5**	107	354	22.3	128	373	65.0	528	755	**53.8**	432	733
25	6	High	**34.1**	86	412	36.7	75	404	**42.6**	98	536	71.3	185	633
30	6	High	**120.2**	153	729	135.5	151	866	292.5	442	1347	**226.1**	305	1262

costs are used to generate the test set. In Archetti et al. (2011b), they consider the PRP-ML with uncapacitated production where the inequalities below that are valid for the uncapacitated lot-sizing problem are also added,

$$I_{t-1} \leq \sum_{i \in N_c} \sum_{j=t}^{l} d_{ij}(1 - y_t) \quad \forall t \in T \tag{69}$$

$$y_t \geq \frac{f}{h_0 j}(y_{t-j} + y_t - 1) \quad 2 \leq t \leq l, 1 \leq j \leq t-1. \tag{70}$$

These inequalities are also added in the $F(ML)|k$ and $F(ML)|nk$ formulations. The results are shown Table 17. Since there are 480 instances, we report the average results of each class. Column Cuts shows the number of cuts generated in the branch-and-cut process. Note that all instances, which were not solved to optimality in Archetti et al. (2011b), are solved to optimality in this implementation.

Table 19 Results on Archetti et al. (2011b) PRP-ML instances with 14 customers and 6 periods.

Class	$F(ML)\|k$			$F(ML)\|nk$		
	Time	Node	Cut	Time	Node	Cut
I	6.8	143	209	**6.7**	115	230
II	5.5	117	183	**5.1**	83	192
III	11.9	303	256	**11.2**	239	264
IV	8.9	208	232	**7.8**	152	235
Avg	8.3	193	220	**7.7**	147	230

The $F(ML)|nk$ could provide slightly better results due to additional valid inequalities added to the formulation and the average number of nodes in the branch-and-bound tree is also reduced.

7. Conclusion.

In this study, we discuss the multi-vehicle aspect in the production routing problem (PRP). Two strong formulations, one with vehicle index and the other without vehicle index, are introduced. We propose several valid inequalities including symmetry breaking constraints to strengthen the formulations and develop branch-and-cut approaches to solve the problems. The computational experiments are performed with three different production strategies. The results show the performances of the strong formulation with vehicle index on smaller number of customers where the strong formulation without vehicle index could produce better lower bounds in large instances. The experiments on single vehicle PRP and IRP instances also show that, the inequalities added to the formulation without vehicle index could also provide slightly better computational performance on the ML policy.

Acknowledgments

Appendix A: Details of the MVPRP Instance Generation.

The MVPRP instances are generated using the instances presented in Archetti et al. (2011b). We select the instances with 50 to generate our test bed. The details are as follows.

1. **Instance size**. We create instances from $n = 10 - 40$, to $10 - 30$ and to $10 - 20$ customers with an incremental of 5 for the time periods $l = 3, 6$ and 9, respectively. The instances with $n = 10, 15$ consists of two subsets of instances with the different number of vehicles.

2. **Customers**: To generate a new instance with n customers, we select customer numbers from 1 to n from the Archetti et al. instance. For the instances with 3 periods, initial inventory levels at customers are all reduced by the factor of 2 to prevent the case where initial inventory levels are already sufficient to satisfy demand during the planning horizon. Also, since the target stock level in Archetti et al. test set is defined by the inventory after consumption. We simply set the $L_i = \bar{L}_i + d_{it}$ where $\bar{L}_i$ is the original value in Archetti et al. test set. This does not have any effect since the values of d_{it} for each customer in Archetti et al. test set are constant during the planning horizon.

3. **Vehicles**. Since the number of vehicles in the original instances are unlimited, we set the number of vehicles equal to $m = \bar{m} = \lfloor n/10 \rfloor + 1$. For the instances with $n = 10, 15$, we also create the other instances with an additional vehicle, i.e., $m = \bar{m} + 1$. Similar to Archetti et al., we set the vehicle capacity related to the maximum inventory level of customers. The vehicle capacity is calculated as $Q = \left\lfloor 1.5\bar{m} \max_{i \in N_c} \{\bar{L}_i\}/m \right\rfloor$. In our test, this setting is appropriate since smaller vehicle capacity could lead to infeasible solutions, especially for the MVPRP-OU.

4. **Instance number.** We generate four instances per one instance size. Each instance is brought from a different instance class in Archetti et al. dataset to ensure that different characteristics are captured. We select the original instance type 1, 25, 49 and 73 to create our test bed.

5. **Production capacity.** In Archetti et al. dataset, they considered uncapacitated production and unlimited inventory capacity at the plant. Since we also consider production capacity in our study, the production capacity is set to $C = \left\lfloor 2\bar{d}/l \right\rfloor$ and plant inventory capacity is set to $L_0 = \lfloor C/2 \rfloor$. In our preliminary test, we found that a smaller production capacity could lead to stockout in the MVPRP-OU.

Appendix B: Details of the IRP Instance Conversion.

The IRP instances in Archetti et al. (2007) were created using different inventory replenishment practice. In Archetti et al. 2007, the inventory level is considered at the beginning of the period. We denote by s_{it}, the begining inventory level at node i in period t and we can observe that

$$s_{it} = I_{i,t-1} \quad \forall i \in N, \forall t \in T. \tag{71}$$

The following modifications are used to convert the instances in Archetti et al. (2007) for our formulation. For ease of presentation, we use the variable without vehicle index and they can be easily converted to vehicle index formulations using $q_{it} = \sum_{k \in K} q_{ikt}$.

First, the supplier can only use the inventory at the beginning of the period to replenish the customers and the supplier must ensure that the total amount shipped to retailers in a period cannot exceed the available amount at the supplier in the beginning of that period, i.e.,

$$s_{0t} \geq \sum_{i \in N_c} q_{it} \quad \forall t \in T. \tag{72}$$

Replacing s_{0t} by $I_{0,t-1}$ and from (22) and (58), the constraints can be rewritten as,

$$I_{0t} \geq B_t \quad \forall t \in T, \tag{73}$$

and these constraints are added to our formulations.

Second, since the inventory costs are charged at the begining of the period starting from period one where $s_{i1} = I_{i0}$ or the initial inventory levels. The fixed cost $\sum_{i \in N} h_i I_{i0}$ is simply added to the formulations.

The other parts of the formulations remain unchanged.

References

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

Adulyasak, Y., J.-F. Cordeau, R. Jans. 2011. Optimization-based adaptive large neighborhood search for the production routing problem. GERAD Tech. Rep. G-2011-52, HEC Montréal, Canada.

Andersson, H., A. Hoff, M. Christiansen, G. Hasle, A. Løkketangen. 2010. Industrial aspects and literature survey: Combined inventory management and routing. *Comput. Oper. Res.* **37** 1515–1536.

Applegate, D., R. Bixby, V. Chvátal, W. Cook. 2011. Concorde tsp solver Http://www.tsp.gatech.edu/concorde.html.

Archetti, C., L. Bertazzi, A. Hertzand, M. G. Speranza. 2011a. A hybrid heuristic for an inventory routing problem. *INFORMS J. Comput.* **Articles in Advance** 1–16.

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

Archetti, C., L. Bertazzi, G. Paletta, M. G. Speranza. 2011b. Analysis of the maximum level policy in a production-distribution system. *Comput. Oper. Res.* **38** 1731–1746.

Armentano, V. A., A. L. Shiguemoto, A. Løkketangen. 2011. Tabu search with path relinking for an integrated production-distribution problem. *Comput. Oper. Res.* **38** 1199–1209.

Bard, J. F., N. Nananukul. 2009. Heuristics for a multiperiod inventory routing problem with production decisions. *Comput. Ind. Eng.* **57** 713–723.

Bard, J. F., N. Nananukul. 2010. A branch-and-price algorithm for an integrated production and inventory routing problem. *Comput. Oper. Res.* **37** 2202–2217.

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

Bertazzi, L., G. Paletta, M. G. Speranza. 2002. Deterministic order-up-to level policies in an inventory routing problem. *Transportation Sci.* **36** 119–132.

Boudia, M., M. A. O. Louly, C. Prins. 2005. Combined optimization of production and distribution. *CD-ROM proceeding of the international conference on industrial engineering and systems management (IESM'05), Marrakech, Morocco*.

Boudia, M., M. A. O. Louly, C. Prins. 2007. A reactive GRASP and path relinking for a combined production-distribution problem. *Comput. Oper. Res.* **34** 3402–3419.

Boudia, M., C. Prins. 2009. A memetic algorithm with dynamic population management for an integrated production-distribution problem. *Eur. J. Oper. Res.* **195** 703–715.

Campbell, A. M., M. W. P. Savelsbergh. 2004. A decomposition approach for the inventory-routing problem. *Transportation Sci.* **38** 488–502.

Carter, M., J. Farvolden, G. Laporte, J. Xu. 1996. Solving an integrated logistics problem arising in grocery distribution. *INFOR* **34** 290–306.

Chandra, P. 1993. A dynamic distribution model with warehouse and customer replenishment requirements. *J. Oper. Res. Soc.* **44** 681–692.

Chandra, P., M.L. Fisher. 1994. Coordination of production and distribution planning. *Eur. J. Oper. Res.* **72** 503–517.

Christiansen, M. 1999. Decomposition of a combined inventory and time constrained ship routing problem. *Transportation Sci.* **33** 3–16.

Coelho, L. C., J.-F. Cordeau, G. Laporte. 2012. The inventory-routing problem with transshipment. *Comput. Oper. Res.* **Article In Press**.

Degraeve, Z., W. Gochet, R. Jans. 2002. Alternative formulations for a layout problem in the fashion industry. *Eur. J. Oper. Res.* **143** 80–93.

Fischetti, M., A. Lodi. 2003. Local branching. *Math. Programming* **98** 23–47.

Fumero, F., C. Vercellis. 1999. Synchronized development of production, inventory, and distribution schedules. *Transportation Sci.* **33** 330–340.

Gaur, V., M. L. Fisher. 2004. A periodic inventory routing problem at a supermarket chain. *Oper. Res.* **52** 813–822.

Gendreau, M., A. Hertz, G. Laporte. 1992. New insertion and postoptimization procedures for the traveling salesman problem. *Oper. Res.* **40** 1086–1094.

Jans, R. 2009. Solving lot-sizing problems on parallel identical machines using symmetry-breaking constraints. *INFORMS J. Comput.* **21** 123–136.

Lin, S. 1965. Computer solutions of the traveling salesman problem. *Bell Syst. Tech. J.* **44** 2245–2269.

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

Miller, C. E., A. W. Tucker, R. A. Zemlin. 1960. Integer programming formulation of traveling salesman problems. *J. Assoc. Comput. Mach.* **7** 326–329.

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

Ropke, S., D. Pisinger. 2006. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Sci.* **40** 455–472.

Ruokokoski, M., O. Solyalı, J.-F. Cordeau, R. Jans, H. Süral. 2010. Efficient formulations and a branch-and-cut algorithm for a production-routing problem. GERAD Tech. Rep. G-2010-66, HEC Montréal, Canada.

Rusdiansyah, A., D. Tsao. 2005. An integrated model of the periodic delivery problems for vending-machine supply chains. *Journal of Food Engineering* **70** 421–434.

Savelsbergh, M. W. P., J.-H. Song. 2007. Inventory routing with continuous moves. *Comput. Oper. Res.* **34** 1744–1763.

Savelsbergh, M. W. P., J.-H. Song. 2008. An optimization algorithm for the inventory routing problem with continuous moves. *Comput. Oper. Res.* **35** 2266–2282.

Sherali, H. D., J. C. Smith. 2001. Improving discrete model representations via symmetry considerations. *Management Sci.* **47** 1396–1407.

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

Solyalı, O., H. Süral. 2011. A branch-and-cut algorithm using a strong formulation and an a priori tour-based heuristic for an inventory-routing problem. *Transportation Sci.* **45** 335–345.

Toth, P., D. Vigo. 2001. An overview of vehicle routing problems. P. Toth, D. Vigo, eds., *The Vehicle Routing Problems*. Society for Industrial and Applied Mathematics, 1–26.