HEURISTIC MODELING FOR SOLVING INVENTORY ROUTING PROBLEM
WITH SIMULTANEOUS DELIVERY AND PICKUP

Ahmad Rusdiansyah¹ and De-bi Cao²
¹Logistics and Supply Chain Management (L&SCM) Laboratory,
Department of Industrial Engineering, Institut Teknologi Sepuluh Nopember (ITS),
Surabaya – Indonesia
Email: arusdianz@gmail.com
²Department of Administration Engineering,
Faculty of Science and Technology,
Keio University, Yokohama – Japan

ABSTRACT
In this paper, we propose a new model of the Inventory Routing Problem (IRP) encountered in the context of reverse logistics. The vehicles here are required not only to deliver forward goods to retailers but also to collect simultaneously returned goods at their locations. The new model will be referred to as the Inventory Routing Problem with Simultaneous Delivery and Pickup (IRPSDP). The IRPSDP is based on the existing Periodic Vehicle Routing Problem (PVRP) model. The objective of the IRPSDP is to minimize the average inventory holding and travelling costs over a given m-day period. The IRPSDP treats the visit frequency as a decision variable instead of a given parameter, to seek the best trade-off between both costs. We present a mathematical formulation and give heuristic procedures to solve the problem. Finally, we test the performance of our approach by conducting some numerical experiments.

KEY WORDS
IRP, Reverse Logistics, Simultaneous Delivery and Pickup, PVRP, Visit Frequency

1. Introduction
The growing environmental responsiveness driven by legislative and economic motives makes many industries apply reverse logistics. Reverse logistics is concerned with all logistic activities performed in the opposite direction to that of the regular supply chain for the purpose of recapturing value or ensuring proper disposal (Fleischmann, [1]). This paper deals with an issue related to integrating inventory and vehicle routing decisions in the context of reverse logistics. In the literature the model that integrates these two decisions is generally called Inventory Routing Problem (IRP). The general objective of the IRP is to minimize system-wide costs consisting of travelling and inventory holding costs. The IRP works in the supply chains with the Vendor Managed Inventory (VMI) system. Under the VMI system, the supplier assumes the responsibility of managing inventories at the retailers instead of only distributing products. The supplier has to make decisions about which retailers to visit, and how much to deliver each day during a regular time horizon.

In this paper we investigate a model of the IRP where the vehicle that is visiting a customer not only delivers a type of goods originated from the depot called “forward goods” but also has to collect and transport another type of goods, called “returned goods”, to the depot. The model be referred to as the Inventory Routing Problem with Simultaneous Delivery and Pickup (IRPSDP). It is assumed here that these two services must not be performed separately and the delivery operation is to be performed first.

In practice, such a system can be found in the industries that implement a reuse system of product carriers such as bottles, crates, and special containers. For example, in the soft drink industries in some developing countries that use returnable glass bottles (RGB) for their products, the delivery of full bottles and collection of empty bottles at a retailer location are carried out simultaneously. The system has also been a common practice in the distribution of liquid petroleum gas (LPG) using special containers.

Although the IRP has been discussed extensively in the literature, there is an open question considering the IRP model for reverse logistics. This paper attempts to fill this gap. We propose an IRPSDP model that modifies the existing Period Vehicle Routing Problem (PVRP). The reason is that the problem environment of the PVRP is appropriate for beverage industries (Dror and Ball [2]) and the periodic characteristics of the PVRP are inherently related to the inventory decisions (Gaur and Fisher, [3]). These findings motivated us to build the IRPSDP model based on the PVRP model.
2. Period Vehicle Routing Problem

The PVRP is a generalization of the well-known Vehicle Routing Problem (VRP). The PVRP is the problem of designing tours for a fleet of vehicles for all the days of a given m-day period. During the period, each retailer will be visited once or several times. The number of visits is called the visit frequency. The vehicles visit a retailer as many times as the visit frequency specified for this retailer. The visits can be performed only on the days that follow one of the so-called visit-day combinations. An example is if a retailer has been assigned to be visited three times during a six-day period, and the allowable visit-day combinations are days (1,3,5) or days (2,4,6). Thus, these visits should take place on days 1, 3, and 5, or on days 2, 4, and 6. Only one visit-day combination can be assigned to this retailer. The objective is to minimize total travelling costs over the period while subject to several constraints such as vehicle capacity and tour duration.

The PVRP model and its solution procedure have been discussed in the literature since the 1970s. Early heuristics for solving the PVRP were proposed by Beltrami and Boudin [4] and Russel and Igo [5]. Then, Christofides and Beasley [6] presented an exact formulation of the PVRP. Chao et al. [7] developed some heuristics to solve the PVRP. Russel and Gribbin [8] proposed a multi-phase approach combining heuristic and exact methods for solving the PVRP. Cordeau et al. [9] presented a unified Tabu Search (TS) heuristic method for solving the PVRP and other related problems. A number of variants of the PVRP with different purposes is also available in the literature. Gaudioso and Palotta [10] proposed a model whose objective is to minimize vehicle fleets.

3 The Proposed Model

In the proposed IRPSDP model, we enhanced the original PVRP model which can be explained as follows.

First, the scope of the PVRP has been changed from a periodic routing problem dealing only with vehicle routing decisions, to an inventory routing problem that considers both vehicle routing decisions and inventory decisions. The objective of the IRPSDP is to minimize system-wide costs consisting of travelling and inventory holding costs instead of travelling cost only.

Second, we argue that the original PVRP, where the visit frequencies are fixed and determined by each retailer itself, can be a restriction of integrated problems such as the VMI system. On the contrary, in the proposed model, the visit frequency is viewed as a decision variable and we assume the supplier determines the visit frequency of each retailer. Along with building vehicle tours, the supplier tries to optimize the assignment of visit frequency to each retailer so that the best trade-off between both costs can be found. Such a trade-off is obvious. Shipping in small quantities with high frequencies will produce low inventory holding costs at retailers, but will lead to high travelling costs. Conversely, infrequent shipments with large quantities will incur high inventory holding costs but low travelling costs.

Third, as an application of reverse logistics, the IRPSDP deals with the feature of simultaneous delivery and pickup. At each visit to the retailers, the vehicle has to transport forward goods originated from the depot, and collect and send returned goods back to the depot.

4. Model Formulation

The IRPSDP deals with the repeated distribution of forward goods from a single depot to a set of n retailers and collection of returned goods from these retailers to the depot over a planning horizon of length m-days, called an m-day period, indexed by T={1,...,t,...,m}.

The IRPSDP can be defined on a complete and undirected graph G=(I, E), where I={0,...,i,...,n} is the vertex set and E is the edge set. Vertex set I0 consists of vertex subset I={1,...,i,...,n} corresponding to the retailers, and vertex 0 corresponding to the depot. Each arc (i,j) ∈ E refers to a non-negative travelling cost between vertex i and vertex j, c_{ij}. The travelling cost matrix is assumed to satisfy the triangle inequality, c_{ij} ≤ c_{ii} + c_{ji}, ∀i,j,r ∈ I_0. Each arc (i,j) ∈ E is also associated with a non-negative travelling time t_{ij}, ∀i,j ∈ I_0. Here, we assume that both travel cost and time are symmetrical: c_{ij}=c_{ji}, ∀i,j ∈ I_0 and t_{ij}=t_{ji}, ∀i,j ∈ I_0. We also assume that these matrices coincide: c_{ij} = t_{ij}, ∀i,j ∈ I_0.

During the m-day period, each retailer i ∈ I will receive the total quantity of forward goods, D_i, and return the total quantity of returned goods, P_i. Retailer i will be visited l_i ≤ m times with delivery sizes d_{i1}, ..., d_{im} and pickup sizes p_{i1}, ..., p_{im}, respectively, over the m-day period. d_{i} and p_{i} are defined as the delivery size and pickup size, respectively, of retailer i at the i-th visit within the period. Let F and S be the sets of allowable visit frequencies and visit-day constraints, respectively.

We assume a retailer consumes the forward goods at a stable rate during the m-day period, the inventory level will reach zero at evenly spaced intervals in time. Accordingly, the delivery size of retailer i over the m-day
period is constant and \( d_i \) can be simplified into \( d_i \) as follows.

\[
d_i = d' = \frac{D_i}{f_i}, \quad i \in I
\]

Likewise, the pickup size of retailer \( i \) over the \( m \)-day period are also equal, and \( p_i \) can be simplified into \( p_i \) as follows.

\[
p_i = p' = \frac{P_i}{f_i}, \quad i \in I
\]

A fleet of vehicles indexed by \( K = \{1, \ldots, k, \ldots, v\} \) with homogenous capacity \( Q \) serves these retailers. We define \( L_{ik} \) as the load of vehicle \( k \in K \) on day \( t \in T \) after visiting retailer \( i \in I \) and \( L_{0ik} \) as the load of vehicle \( k \in K \) on day \( t \in T \) after leaving the depot. During a tour \( L_{ik} \) must never exceed the vehicle capacity \( Q \), and never be less than 0, \( 0 \leq L_{ik} \leq Q \). To fulfil the feasibility, we assume that \( d_i, p_i \leq Q \).

All forward goods carried by the vehicle are originated from the depot and must be distributed to the visited retailers, while all returned goods collected must be transported to the depot.

The objective of the IRPSDP is to minimize the average system-wide costs over the \( m \)-day period, which is defined as the sum of the average inventory holding cost for all retailers and the average travelling cost for all tours over the \( m \)-day period. Let \( h \) be the inventory holding cost per unit of the forward goods held per day. As \( f_i \) is treated as a decision variable, the average inventory holding cost for retailer \( i \) over the \( m \)-day period, \( C_{inv_i} \), can be formulated as a function of \( f_i \) described as follows.

\[
C_{inv_i}(f_i) = \frac{hD_i}{2f_i} \tag{3}
\]

The average inventory holding cost for all retailers over the \( m \)-day period, \( TC_{inv} \), can thus be seen as a function of \( (f_1, f_2, \ldots, f_n) \) and described as follows.

\[
TC_{inv}(f_1, f_2, \ldots, f_n) = \sum_{i \in I} C_{inv_i} \tag{4}
\]

Let \( TSP_{ik} \) denote the travelling cost of the travelling salesman tour \( H_{ik} \) carried out by vehicle \( k \in K \) on day \( t \in T \).

\[
TSP_{ik} = \sum_{i \in I} c_{y_{ijk}}, \quad t \in T, k \in K, i, j \in I_0 \tag{5}
\]

Similar to \( TC_{inv} \), we also consider that the average travelling cost for all tours over the \( m \)-day period, \( TC_{trp} \), is a function of \( (f_1, f_2, \ldots, f_n) \) and described as follows.

\[
TC_{trp}(f_1, f_2, \ldots, f_n) = \frac{1}{m} \sum_{i \in I} \sum_{t \in T} TSP_{itk}, k \in K \tag{6}
\]

The average system-wide costs over the \( m \)-day period, \( TC \), which is the objective function of the IRPSDP, can be formulated as follows.

\[
\text{Minimize } TC_{inv} + TC_{trp} \tag{7}
\]

subject to

\[
\sum_{i \in I} y_{itk} = f_i, \quad \forall i \in I, f_i \in F; t \in T; k \in K \tag{8}
\]

\[
\sum_{r = 1}^{m} y_{irk} = 1, \quad t = 0, \ldots, \left( m - \frac{m}{f_i} \right); \quad \forall i \in I; f_i \in F; k \in K \tag{9}
\]

\[
L_{0ik} - \sum_{i \in I} d_i y_{itk} = 0, \quad \forall t \in T, \quad \forall k \in K \tag{10}
\]

\[
x_{r dik} L_{r k} - \sum_{i \in I} p_i y_{itk} = 0, \quad \forall r \in I, \forall t \in T, \forall k \in K \tag{11}
\]

\[
x_{ijk}(L_{itk} - d_j + p_j - L_{jik}) = 0, \quad \forall i, j \in I; \forall t \in T; \forall k \in K \tag{12}
\]

\[
0 \leq L_{itk} \leq Q, \quad \forall i \in I, \forall t \in T, \forall k \in K \tag{13}
\]

\[
\sum_{i \in I} \sum_{j \in I} (t_{ij} + s_{ij}) x_{ijk} \leq R, \quad \forall i \in I; \forall k \in K \tag{14}
\]

\[
\sum_{i \in I_0} \sum_{j \in I_0} x_{ijk} - \sum_{j \in I_0} x_{ijk} = 0, \quad \forall r \in I_0; \forall t \in T; \forall k \in K \tag{15}
\]

\[
\sum_{j \in I} x_{ijk} \leq 1, \quad \forall r = 0; \forall t \in T; \forall k \in K \tag{16}
\]

\[
\sum_{i \in B} x_{ijk} \leq |B| - 1, \quad \forall t \in T; \forall k \in K; B \subseteq I; |B| \geq 2 \tag{17}
\]

\[
x_{ijk} \in \{0, 1\}, \quad \forall i, j \in I_0; \forall t \in T; \forall k \in K \tag{18}
\]

\[
y_{itk} \in \{0, 1\}, \quad \forall i \in I_0; \forall t \in T; \forall k \in K \tag{19}
\]
where binary variables $x_{ijk}$ take value 1 if vehicle $k \in K$ visits vertex $j \in \mathcal{L}_o$ immediately after vertex $i \in \mathcal{L}_o$ on day $t \in T$, and to take value 0 otherwise. Also, let us define binary variables $y_{ijk}$ to take value 1 if vertex $i \in \mathcal{L}_o$ is visited by vehicle $k \in K$ on day $t \in T$, and to take value 0 otherwise. The constraints in (8) guarantee that each retailer is served as many times as the assigned visit frequency. The constraints in (9) ensure that each retailer is visited only on the days corresponding to the assigned visit-day combination and visit frequency. Moreover, equations (8) and (9) are required to fulfill the stationary-interval property. The constraints in (10) guarantee that all quantities of forward goods distributed over a tour originate from the depot, while the constraints in (11) guarantee that all quantities of returned goods collected over a tour will be sent to the depot. The constraints in (12) guarantee the schedule feasibility with respect to vehicle loads during a tour. The constraints in (13) ensure that the vehicle loads during a tour are nonnegative and never exceed the maximum vehicle capacity $Q$. The constraints in (14) guarantee that the duration of a tour never exceeds the maximum tour duration $R$. Equation (15) ensures that each vehicle that visits a retailer on a given day also leaves that retailer on the same day. Equation (16) guarantees that each vehicle can be used only once on any day. Equation (17) is a standard subtour elimination constraint. Equations (18) and (19) ensure binary values for the solution. Constraints (14)–(19) are the standard PVRP constraints adapted from Cordeau et al [1997].

5. Algorithms

We define a “node” as a visit at retailer $i$ by the vehicle in a day. Thus, each retailer $i \in \mathcal{I}$ will have $f_i \in \mathcal{F}$ node(s). We also define the moving cost of retailer $i \in \mathcal{I}$ as the insertion cost of its nodes in their new locations minus the saving obtained by removing these nodes from their current locations. There are two phases in the algorithms: the initialization and improvement phases, which can be described as follows.

Phase I: Initialization

The objective of the initialization phase is to search for the best visit frequency for each retailer. This phase consists of five steps. The step-by-step algorithm of this phase is presented as follows.

**Step 1.1** Initial condition set up

- Each retailer $i \in \mathcal{I}$ is assumed to be assigned the lowest allowable visit frequency ($f_i = 1$) and has only one node. Hence, the delivery size is equal to the total quantity of forward goods, $d_i = D_i$, while the pickup size is equal to the total quantity of returned goods, $p_i = P_i$.

- Sort the data set in nonincreasing order of $d_i$.

**Step 1.2** Seed selection

- For each tour $H_a$, which is carried by vehicle $k \in K$ on day $t \in T$, choose a seed retailer. The seed retailers selected should be geographically well scattered.

- Assign to each seed retailer a visit-day combination of $f_i = 1$ containing day $t$.

**Step 1.3** Tour initialization

- Construct $mK$ initial tours by linking each seed retailer and the depot.

**Step 1.4** Assignment of unrouted retailers

- For each unrouted retailer, compute the feasible moving cost for all adjacent vertices of the available tours. Insert the unrouted retailer into one of the tours with the minimum feasible moving cost.

- Assign to this retailer a visit-day combination associated with this tour.

- Repeat these previous stages until all retailers have been inserted into the tours.

**Step 1.5** Visit frequency optimization (see below)

- According to the heuristic used, retrieve the visit frequency optimization procedure for all $m/2$ stages.

- In the last stage, if there are no more retailers that can feasibly increase their visit frequency, stop and go to the improvement phase.

Phase II: Improvement phase

While relaxing the inventory holding cost, the improvement phase seeks a further reduction in travelling costs, which eventually decreases the objective function. The improvement phase consists of two steps: visit-day combination exchanges and tour exchanges. Both employ the tabu search method.

The tabu search method explores the solution space by moving from the current solution to the most favourable solution in its neighbourhood. To avoid being caught in a local optimum or having cycling, such as visiting immediate preceding solutions, the algorithm imposes a flexible and dynamic list of forbidden moves called the tabu list. The tabu list $\Gamma$ corresponds to a set of moves that is excluded from the search process. The number of iterations that the moves are tabu-active is known as tabu length. The tabu length is based on experience. Whenever a move has been successfully performed, it is then inserted at the end of the tabu list, while at the same time the oldest element of this list is deleted. During the search process, the algorithm explores the entire neighbourhood to select the allowable non-tabu moves providing the highest savings. The step-by-step algorithm of this phase is presented as follows.
Step 2.1 Visit-day combination exchanges

The purpose of this step is to exchange the visit-day combination of a retailer with other combinations within the same visit frequency, to reduce the current travelling cost. This step considers two types of moves called combination-insertion and combination-exchange moves. It consists of four stages described as follows.

- Generate all possible moves to exchange the current visit-day combination of each retailer \(i \in \mathcal{I}\) with the other allowable visit-day combinations, excluding those listed on the tabu list.
- Select the retailer that can produce the highest reduction of the current travelling cost.
- Update the tabu list.
- Repeat these previous stages until there are no more possible moves.

Note that in this step we do not consider retailers \(i \in \mathcal{I}\) with \(f_i = m\) as they have only one visit-day combination. Also, to save computational efforts, all retailers with \(f_i = 1\) are excluded in this step and will be addressed in the next step.

Step 2.2 Tour exchanges

The purpose of this step is to improve the sequences of the tour in order to reduce the current travelling cost. This step considers three types of moves, which are the single-internal-insertion, single-external-insertion, and 2-opt exchange moves. Similar to the previous step, it consists of four stages described as follows.

- Generate all possible moves within a tour or between two tours, excluding those listed on the tabu list.
- Select the retailer that can produce the highest reduction of the current travelling cost.
- Update the tabu list.
- Repeat these previous stages until there are no more possible moves.

5.1. Visit Frequency Optimization Procedure

The objective of the Visit Frequency Optimization Procedure is to search for the best visit frequency for each retailer \(i \in \mathcal{I}\). This procedure is carried out in \(m/2\) stages. Each stage represents the target visit frequency that is defined as the visit frequency to which the current visit frequency is to be increased. For example, for the six-day period case, there are three stages. The target visit frequencies of stages 1, 2, and 3 are 2, 3, and 6, respectively. In any iteration of a stage, we evaluate all retailers that may feasibly increase their current visit frequency to the target visit frequency. We compute \(\Delta TC\), which is the sum of the change in inventory holding cost \(\Delta TC_{inv}\) and the change in travelling cost \(\Delta TC_{trp}\), led by the increase in the visit frequency of retailer \(i\) from \(f_i = \lambda\) (current visit frequency) to \(f_i = \mu\) (target visit frequency). The formulation for calculating \(\Delta TC\) can be expressed as

\[
\Delta TC_i = \omega_{inv} \Delta TC_{inv} - \omega_{trp} \Delta TC_{trp}, i \in \mathcal{I}
\]

where \(0 \leq \omega_{inv}, \omega_{trp} \leq 1\) are two user-controlled parameters of the inventory holding cost and the travelling cost, respectively, to capture the scale difference of both costs. The formulation of \(\Delta TC_{inv}\) is described as follows.

\[
\Delta TC_{inv} = \frac{hD_i}{2} \left( \frac{1}{\lambda} - \frac{1}{\mu} \right), i \in \mathcal{I}; \lambda < \mu, \lambda, \mu \in F
\]

(21)

\(\Delta TC_{trp}\) is the difference of the values of \(TC_{trp}\) before and after the increase in the visit frequency of retailer \(i\), and is shown as follows.

\[
\Delta TC_{trp} = TC_{trp}[before] - TC_{trp}[after], i \in \mathcal{I}
\]

(22)

The value of \( TC_{trp}[after] \) is obtained by finding a feasible increasing-node(i) move that produces the smallest additional moving cost. After evaluating all possibilities, we select retailer \(i^*\) in which \(\Delta TC_{i^*} = \max_{i \in \mathcal{I}} \{\Delta TC_i\}\). We then need to adjust delivery size \(d_{i^*}\) to be a new delivery size \(d_{i^*}'\), which is:

\[
d_{i^*}' = \frac{\lambda}{\mu} d_{i^*}, i \in \mathcal{I}; \lambda < \mu \in F
\]

(23)

Likewise, pickup size \(p_{i^*}\) is also adjusted to be a new pickup size \(p_{i^*}'\) as follows:

\[
p_{i^*}' = \frac{\lambda}{\mu} p_{i^*}, i \in \mathcal{I}; \lambda < \mu \in F
\]

(24)

At the end of this iteration, we will find that the number of retailers serviced \(\lambda\) times \(n_\lambda\) decreases by one, the number of retailers serviced \(\mu\) times \(n_\mu\) increases by one, and the total number of visits \(N\) increases by \((\mu - \lambda)\). The iteration is repeated until there are no more retailers that can feasibly increase their current visit frequency in this stage. In that case, we go to the next stage to carry out the same procedure.

5.2. Variants of Heuristics

Based on the definition of current and target visit frequencies, our heuristics can be placed in two categories. The first heuristic, called INC-1, ensures that
the search space of a stage covers all retailers whose current visit frequency equates to the immediate-lower possible number of the target visit frequency of this stage. For example, in the six-day period case, if the target visit frequency of the stage is 3, then only retailers whose visit frequency are 2 will be considered in this stage. Another type, named INC-2, is designed in a slightly different way. In INC-2, the current visit frequency is not necessarily the immediate-lower possible number of the target visit frequency. For example, in the six-day period case, if the target visit frequency of the stage is 3, the retailers whose visit frequency are 1 or 2 can be considered in this stage (see Table 1). Figures 1a and 1b visualize the transition diagrams of each heuristics.

Table1: Characteristics of Heuristics INC-1 and INC-2

<table>
<thead>
<tr>
<th>The length of period (m)</th>
<th>Stage</th>
<th>Target visit frequency</th>
<th>Heurist ic</th>
<th>Heurist ic</th>
<th>Heurist ic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>INC-1</td>
<td>INC-2</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>1, 2</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>1, 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>1, 2, 3</td>
<td></td>
</tr>
</tbody>
</table>

6. Numerical Experiments

We conducted several numerical experiments for three sets of IRPSDP instances which modified ten PVRP instances with n ranging between 48 and 288 retailers generated by Cordeau et al. [9]. we may observe that INC-2 outperforms INC-1. INC-2 provides better solutions in six instances for each of groups I and II. Meanwhile, INC-1 provides better solutions in only two and three instances for groups I and II, respectively. Both heuristics provide equivalent results in two instances for group I, and in one instance for group II. The mean ranks of INC-1 are 1.7 and 1.65 while those of INC-2 are 1.3 and 1.35 for groups I and II, respectively. The situation is different for group III, which uses a random generator for defining pickup demands. Our observation shows that INC-1 provides a better performance than INC-2. INC-1 shows lower TC in seven instances, while INC-2 in only one instance. Again, an equivalent result is obtained for the smallest size problem. The mean ranks of INC-1 and INC-2 are 1.15 and 1.85, respectively. It is most likely that the total number of visits (N) influences the quality of the results. For instance, the heuristic with higher N provides better solutions in eight out of 10 problems for group I, in nine out of 10 problems for group II, and in seven out of 10 problems for group III. In terms of computational times, the results confirm that INC-1 outperforms INC-2 for all instances. INC-2 requires higher computational time than INC-1 because of its larger searching spaces. We may also observe that the larger the data size the larger the deviation between the CPU time required by INC-1 and INC-2.
7. Conclusions

We have proposed a new Inventory Routing Problem (IRP) model encountered in reverse logistics for a supply-chain system where forward and reverse distributions are performed simultaneously. The new model is called the Inventory Routing Problem with Simultaneous Delivery and Pickup (IRPSDP), and is suitable in a Vendor Managed Inventory (VMI) system. The IRPSDP has been developed based on the existing PVRP model. The objective is to minimize the average inventory holding and travelling costs during a given m-day period. The most significant contribution of this paper is the incorporation of visit frequencies as a decision variable in the model to capture the conflicting nature of both costs. We have built a mathematical formulation of the IRPSDP and two types of heuristics to solve the problem. The results of the experiments confirm that our model could be a promising tool for the supplier under VMI system to coordinate inventory replenishment and delivery policies.

References