Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Food Engineering 70 (2005) 421–434

JOURNAL OF FOOD ENGINEERING

www.elsevier.com/locate/jfoodeng

An integrated model of the periodic delivery problems for vending-machine supply chains

Ahmad Rusdiansyah *, De-bi Tsao

Department of Industrial Engineering and Management, Tokyo Institute of Technology, Tokyo 152-8552, Japan

Received 30 September 2003; received in revised form 1 May 2004; accepted 5 May 2004
Available online 22 December 2004

Abstract

In this paper we present a model and solution procedures of the Inventory Routing Problem (IRP) encountered in vending machine supply chains working under vendor-managed inventory (VMI) scheme. The new IRP model is built based on the existing Periodic Vehicle Routing Problem with Time-windows (PVRPTW). The model will be referred to as the Integrated Inventory and Periodic Vehicle Routing Problem with Time-windows (IPVRPTW). The objective of the IPVRPTW is to minimize the sum of the average inventory holding and traveling costs during a given *m*-day period. The visit frequency is treated as a decision variable instead of as a fixed parameter. We attempt to optimize the visit frequency of each retailer and to build vehicle tours simultaneously in order to find the best trade-off between the inventory holding and traveling costs. Computational experiments are conducted on some instances taken from the literature to evaluate the performance of the proposed model.

Keywords: Vending-machine supply chains; VMI; IRP; PVRPTW; Visit frequency

1. Introduction

1.1. Vendor managed inventory

Vending machines become an essential part of the daily lives in several countries. In Japan, owing to their ubiquitous and 24-h availability, the unmanned machines are very popular especially for busy people. The Annual report of the Japan Vending Machine Manufacturers Association (2003) shows that there are more than 5.5 million vending machines that serve this country's population of about 126 million. It makes Japan's vending machine density the highest in the world. The majority of the vending machines sell beverages (47.3%), cigarettes (11.3%) and food (2.2%). Among the beverage vending machines, most of them sell soft drinks (83.7%) while the others sell milk (7.0%), coffee and chocolate (6.1%), and alcoholic beverages (3.2%).

In general, the operators of food and beverage vending machines in Japan can be classified into two groups. The first group is local merchants who operate their vending machines outside of their main business, and make the stock replenishments by themselves. The second group, which will be the focus of this research, is vending machine refilling companies who manage a network of vending machines located in dispersed places. The company is responsible for replenishing the inventories at all of the vending machines. Note that throughout this paper, terms "the supplier" and "the retailers" are used to refer to the company and the vending machines, respectively.

Traditionally, the vending machine refilling companies apply a distribution system of goods known as *driver-sell* (see Gloden & Wasil, 1987). In this system, the driver acting as a salesman has to visit vending machines to replenish the inventories (and to collect the money).

* Corresponding author. Fax: +81 30 5734 2947.
E-mail address: arusdian@ie.me.titech.ac.jp (A. Rusdiansyah).

0260-8774/$ - see front matter
doi:10.1016/j.jfoodeng.2004.05.073

In other words, the vending machines are visited without prior information on their current stock levels and demands. Accordingly, this distribution system may cause high operational costs and some problems, such as out-of-stocks.

Due to demands for efficiency, some efforts should be carried out to reduce network costs. One of the efforts to make the operations of the supply chains more efficient is to adopt a scheme called Vendor Managed Inventory (VMI).

VMI refers to a situation in which a supplier can monitor the inventory levels at its retailers, and has the liberty to decide when and how much the stock to replenish at each retailer. VMI has some advantages for both parties. The supplier has more opportunities to coordinate the deliveries; for example, by delaying and advancing deliveries according to the inventory situations at the retailers and the transportation considerations (Waller, Johnson, & Davis, 1999). Customer service levels may increase in terms of the reliability of product availability, because the supplier can use the information collected on the inventory levels at the retailers to better anticipate future demand (Kleywegt, Nori, & Savelsbergh, 2004).

The most important requirement of VMI is that the supplier can obtain accurate information of the inventory levels at the retailers in a timely and efficient way. Hence, VMI requires a reliable device to collect and transmit the necessary data between the retailers and the supplier. In order to do this, some types of vending machines currently have been equipped with an on-line messaging device that can regularly send electronic data specifying the machine's point of sale data and stock levels to the supplier (Ghiani, Laporte, & Musmanno, 2004). For example, a small telemetry device that can automatically generate and transmit information (e.g. e-mails) to the supplier based on the signals originated in sensors, switches or relays in the machine. Another type is such an Electronic Data Interchange (EDI) device using a cellular-phone-like technology. Moreover, these devices can also generate a warning message if a machine is running out of products (and exchange coins) or any malfunction is detected.

1.2. Inventory routing problem

Under VMI, the supplier is responsible for coordinating the inventory replenishment and vehicle routing decisions. The problem of optimal coordination of inventory replenishment and vehicle routing is known as the *Inventory Routing Problem or IRP* (Kleywegt et al., 2004). This problem is also called *the Vendor Managed Inventory Routing or VMR* (Ghiani et al., 2004).

The IRP has been discussed extensively in the literature. According to Bramel and Simchi-levi (1997) the IRP can be categorized into three types of models: single-period, multi-period, and infinite horizon. Some considerable papers considering single-period models are Federgruen and Zipkin (1984), and Golden, Assad, and Dahl (1984). In Federgruen and Zipkin (1984), the objective of the problem is to allocate the supplier's limited quantity of inventory to the customers in order to minimize the transportation cost plus the inventory related cost. Golden et al. (1984) proposed a heuristic model to maintain the customer' inventory levels at a certain level to minimize the costs. The model is built based on the urgency of each customer, which is the ratio between the tank's current inventory level and the tank size.

An important representative of the multi-period IRP models is Chien, Balakrishnan, and Wong (1989). The paper decomposes the multi-period problem into a sequence of single-period problems where the decision made in a day can affect the revenues to be received in the next days. Other significant works in this category are Bell et al. (1983), Dror, Ball, and Golden (1986) and Campbell, Clarke, and Savelsbergh (2002). These papers develop different solution approaches for solving the problems of product distribution in gas industries. Recently, Gaur and Fisher (in press) developed a multi-period IRP model for a supermarket-chain in the Netherlands.

Two papers dealing with the infinite-horizon IRP models are Anily and Federgruen (1990), and Bramel and Simchi-Levi (1995). Anily and Federgruen (1990) built heuristic algorithms to establish long-run inventory rules and routing patterns for a set of customers. The objective is to minimize the sum of the long-run average transportation and inventory costs. First, all customers are partitioned into several regions. Each region is served by a truck. A customer may belong to more than one region. Each time a customer in a region is visited, the vehicle also visits all customers in this region in an efficient tour. In Bramel and Simchi-Levi (1995), it is assumed that customers can carry an unlimited amount of inventory. The obtain a solution, the authors developed a location-based heuristic based on the capacitated concentrator location problem (CCLP) to determine fixed partitions. These partitions are then served using a procedure that is similar to that of Anily and Federgruen (1990).

Another stream of research that is closely related to the IRP is the Periodic Vehicle Routing Problem or PVRP (Gaur & Fisher, in press). In the literature, the PVRP is generally viewed as a generalization of the well-known vehicle routing problem (VRP), through extending the planning horizon from a single day to m-days. The objective of the PVRP is to minimize the traveling cost over the m-day period.

Several papers have proposed solution procedures for the PVRP such as Beltrami and Bodin (1974), Russell and Igo (1979), Christofides and Beasley (1984), Tan and Beasley (1984), Russell and Gribbin (1991), Gaudioso and Paletta (1992), Chao, Golden, and Wasil (1995), Cordeau, Gendreau, and Laporte (1997) and Baptista, Oliveira, and Zúquete (2002). An important variant of the PVRP is the Period Vehicle Routing Problem with Time-windows (PVRPTW). This problem can also be seen as a generalization of the Vehicle Routing Problem with Time-windows (VRPTW). The first article that discusses the PVRPTW is Cordeau, Laporte, and Mercier (2001). To our knowledge, there are no other works that specifically address this problem variant.

1.3. Proposed model

Although the PVRP and the PVRPTW have been discussed extensively in the literature, both problems generally deal with the delivery of products under conventional inventory management, in which the retailers monitor their own inventory levels and place orders to the supplier with their own decisions. Each retailer specifies the visit frequency and allowable visit-day combinations.

Alternatively, in this paper we propose an IRP model, which enhances the existing PVRPTW model. We have selected the PVRPTW as the basis of our proposed model because the problem environment of the PVRP (hence the PVRPTW) is essentially appropriate for applications such as beverage industries (Dror et al., 1986). Moreover, time-window constraint is normally applied in vending machine supply chains. The new model will be referred to as the Integrated Inventory and Periodic Vehicle Routing Problem with Time-windows (IPVRPTW).

Two major enhancements are considered in the IPVRPTW. First, the objective is to minimize system-wide costs consisting of inventory holding and traveling costs over the *m-day* period. Second, the visit frequency is treated as a decision variable, instead of as a given fixed parameter. We attempt to optimize the assignments of visit frequency to the retailers in order to find the best trade-off between inventory holding and traveling costs.

Such a trade off is important to accommodate in the model, recognizing the conflicting nature between these costs. Frequent shipments with small size deliveries results in low inventory holding costs but high traveling costs, whereas infrequent shipments require large size deliveries incurring high inventory holding costs and low traveling costs.

The optimization is carried out along with the vehicle routing process. While building the vehicle tours, we gradually increase the visit frequency of each retailer from the lowest allowable visit frequency to the highest allowable one.

The formal objective of the IPVRPTW then consists of simultaneously assigning the best visit frequency and the best visit-day combination to each retailer, and building a set of feasible tours for each day during the *m-day* period, such that all demands are fulfilled and all constraints are satisfied, and the system-wide costs over the *m-day* period is minimized. The constraints which must be satisfied include vehicle capacity, time window, and tour duration.

The remaining of this paper is organized as follows. The model formulation is provided in Section 2. We develop heuristic algorithms for solving the problem in Section 3. Section 4 shows numerical experiments. Finally, the conclusions and future research are drawn in Section 5.

2. Model description and formulation

2.1. Planning horizon

In this research we consider a finite planning horizon consists of *m-days* indexed by $T = \{1, \ldots, t, \ldots, m\}$. We adopt *the rolling horizon scenario* (Zipkin, 2000) for the practical use of our model. We assume that the finite planning horizon can be passed along the time toward an infinite planning horizon. Fig. 1 displays the visualization of the construction of delivery schedules under the rolling horizon scenario.

In this research, we are particularly concerned with the 6-day period because it can produce recognizable and practical planning calendars. The modification of the period length, however, is straightforward.

2.2. Related VRP models

In this section, we give an overview of four vehicle routing problem models related to the IPVRPTW model, which are: VRP, VRPTW, PVRP, and PVRPTW.

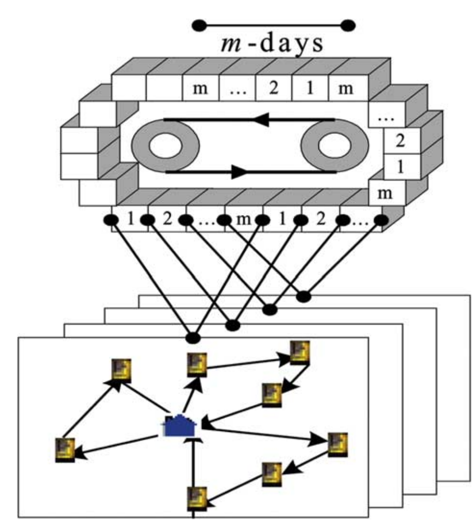

Fig. 1. Delivery schedules under the rolling horizon scenario.

The VRP is defined on a complete and undirected graph $G = (I_0,E)$ where $I_0 = \{0,\ldots,i,\ldots,n\}$ is the vertex set and E is the arc set. Vertex set I_0 consists of vertex subset $I = \{1,\ldots,i,\ldots,n\}$ which corresponds to the retailers, and vertex 0 which corresponds to the supplier's depot. Each arc $(i,j) \in E$ is associated with the non-negative traveling time between vertex i and vertex j, t_{ij}, $i, j \in I_0$. The traveling time matrix is assumed to satisfy the triangle inequality, $t_{ij} \leqslant t_{ir} + t_{rj}$ for $\forall i,j,r \in I_0$. Each arc $(i,j) \in E$ also refers to a non-negative traveling cost $c_{ij}, i,j \in I_0$. It is assumed that the traveling time and the traveling cost matrices coincide $c_{ij} = t_{ij}$, $\forall i,j \in I_0$.

Each retailer $i \in I$ corresponds to a known non-negative delivery size q_i, whereas no delivery is required for the depot ($q_0 = 0$). To serve these retailers, a set of v vehicles indexed by $K = \{1,\ldots,k,\ldots,v\}$ with identical capacity C, is available at the depot. It is assumed that $q_i \leqslant C$, for $\forall i \in I$ to ensure the feasibility. The VRP consists of building a set of exactly K tours with minimum total traveling cost/traveling time, such that: (a) each retailer must be visited by at most one tour, (b) each vehicle must start and end at the depot, (c) each vehicle may perform at most one tour, (d) the load of the vehicle must not exceed the vehicle capacity during a tour.

The VRPTW is an extension of the VRP where time-window and tour duration constraints are imposed (see Bräysy (2003) for a comprehensive review on VRPTW).

Each retailer $i \in I$ is associated with a time window $[b_i,e_i]$ in which b_i is the earliest possible departure time, while e_i is the latest possible arrival time at this retailer. Each retailer $i \in I$ also corresponds to a given non-negative service time d_i. No service time is applied to the depot, $d_0 = 0$. Time windows are defined by assuming that all vehicles leave the depot at the time instant 0. The service of each retailer $i \in I$ must start within the associated time window. The vehicle may arrive before time instant b_i at the location of retailer $i \in I$, but it must wait until this time instant. Furthermore, the tour-duration during a tour must not exceed the maximum tour duration R.

The PVRP generalizes the VRP by extending the single-day period to m-day period. During the period, each retailer $i \in I$ is visited $1 \leqslant f_i \leqslant m$ times (called visit frequency). Each retailer $i \in I$ determines its *visit frequency* f_i. These visits must follow the set of allowable visit-day combinations S. For instance, if a retailer must be visited three times during a 6-day period, and the allowable visit-day combinations are $(1,4,6)$ and $(2,3,5)$, thus this retailer can only be visited in the days associated with one of these combinations.

The PVRP consists of simultaneously finding a set of K tours for each day over the period and assigning the best visit-day combination to each retailer such that all requirements are fulfilled and the total traveling costs over the m-day period is minimized. The PVRP respects to all constraints of VRP, and some additional constraints: (a) each retailer must select one allowable visit-day combination, (b) each retailer is only visited on the days associated with the selected visit-day combination, (c) each vehicle can travel between two retailers on a day if and only if both retailers are scheduled to be visited on that day, (d) each retailer can only be served at most once per day.

Similar to the relationship between the VRP and the VRPTW, the PVRPTW can be viewed as an extension of the PVRP by imposing time-window and tour-duration constraints. The PVRPTW can also be considered as a periodic version of the VRPTW by extending the single-day period to m-day period. Finally, Fig. 2 summarizes the interconnection among the problems which have been discussed so far.

2.3. Inventory model

In this section we describe the inventory model for each retailer $i \in I$. We adopt the classical Economic Lot Size model. There are several assumptions in this inventory model, such as

(a) The demand rate is specific, stable and deterministic.
(b) The order quantity needs not be an integral number of units.
(c) Initial inventory of each retailer $i \in I$ is zero.
(d) Every replenishment is made when the inventory level drops to zero.
(e) The replenishment lead time is zero duration.
(f) No shortages are allowed.

Each retailer $i \in I$ faces demand rate of λ_i units of product per day. Thus, during the m-day period, each retailer $i \in I$ has the total demand D_i formulated as follows:

$$D_i = \lambda_i m, \quad i \in I. \tag{1}$$

It is assumed that $D_i \leqslant C$ to ensure the feasibility.

During the period, retailer $i \in I$ requires to be replenished f_i times with delivery sizes, $q_{i1}, q_{i2}, \ldots, q_{ir}, \ldots, q_{if_i}$. Let u_{ir} be the interval between the rth replenishment and the $r + 1$th replenishment of retailer $i \in I$, delivery size q_{ir}, $r = 1,\ldots,f_i$ can be formulated as follows:

$$q_{ir} = \lambda_{ir} u_{ir}, \quad i \in I; \quad r = 1,\ldots,f_1; \quad f_i \in F \tag{2}$$

Fig. 3 shows the inventory pattern of retailer $i \in I$ that require 3 times of replenishments during the 6-day period.

Since some moves in the algorithms will exchange the visit-day combination and the visit frequency of retailers, we need to ensure that these changes will not affect the optimality of the inventory holding cost. Accordingly, our model requires a condition in which the inventory holding cost charged for each retailer $i \in I$ must be kept minimum during the algorithms.

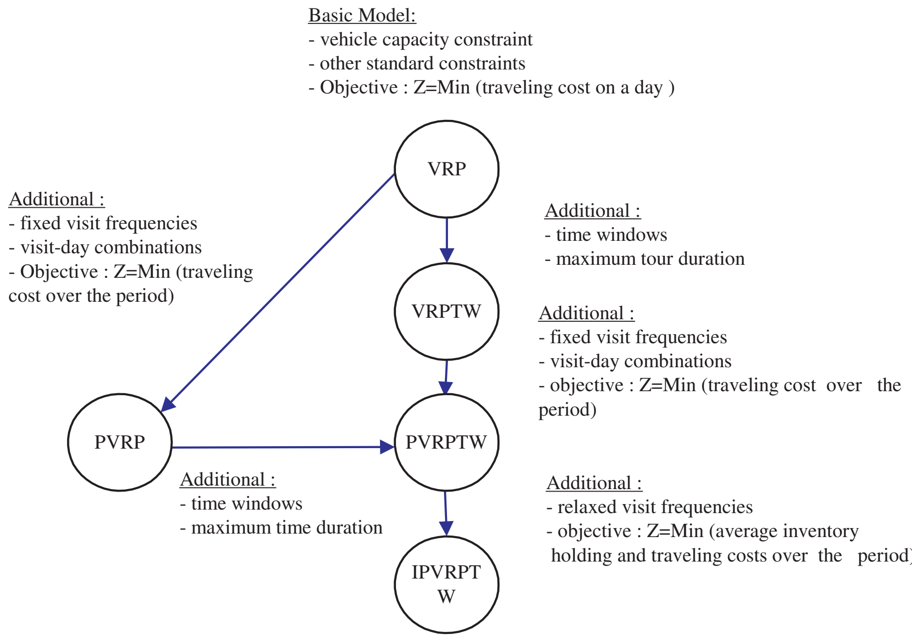

Fig. 2. Interconnection among the problems.

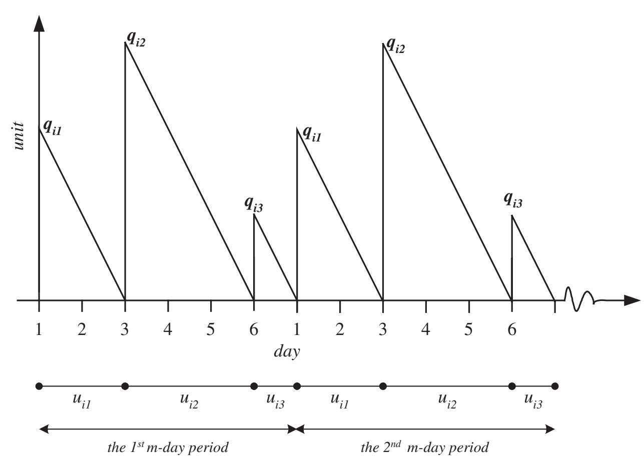

Fig. 3. Inventory pattern of retailer i with $f_i = 3$.

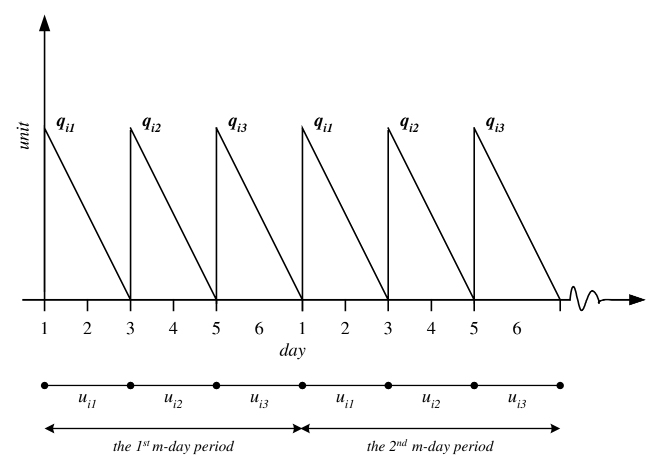

Fig. 4. Inventory pattern of retailer i with $f_i = 3$ satisfying the stationary interval property.

To satisfy this requirement, we follow the *stationary interval property* explained in Bramel and Simchi-levi (1997) where the inventory cost of retailer $i \in I$, which requires to be replenished f_i times over a finite period, is minimum if the inventory replenishments are performed at equal interval with equal delivery size. The property is described as follows:

$$u_{i1} = u_{i2} = \cdots = u_{if_i} = \frac{m}{f_i}, \quad i \in I. \tag{3}$$

Hence, q_{ir} can be represented as q_i only and formulated as follows:

$$q_i = q_{ir} = \lambda_i \frac{m}{f_i} = \frac{D_i}{f_i} \quad i \in I; \quad r = 1, \ldots, f_i \tag{4}$$

Fig. 4 shows the inventory pattern of retailer i that require 3 times of replenishments during the 6-day period satisfying the *stationary interval property*.

In our model, we assume that there is no ordering cost incurred. This assumption is logical in situations encountered in vending machine supply chains using EDI device. As written by Silver, Pike, and Peterson (1998), the ordering cost can be significantly reduced to be very low, owing to the use of electronic transactions.

Moreover, to fulfill the stationary interval property we also need to define the sets of allowable visit frequencies F and visit-day combinations S. In the algebra, the constant intervals between any two replenishments during the m-day period can only be achieved, if F consists of the *divisors* (factors) of the period length m. The

Table 1
The allowable visit frequencies and visit-day combinations for the 6-day period

F	S					
	#1	#2	#3	#4	#5	#6
1	100000	010000	001000	000100	000010	000001
2	100100	010010	001001			
3	101010	010101				
6	111111					

associated S then can be easily determined. Table 1 show the allowable visit frequencies and visit-day combinations for $m = 6$. The six consecutive digits used in this table correspond to the days of visits, where 1 indicates that there exists a visit in the associated day and 0 indicates otherwise.

2.4. Mathematical formulation

The following is the notation that will be used in our model:

- m — period length
- c_{ij} — traveling cost of traversing edge (i,j)
- t_{ij} — traveling time of traversing edge (i,j)
- h — inventory holding cost per unit of the goods held per day
- $C_{\text{inv}(i)}$ — average period inventory holding cost of retailer $i \in I$
- TC_{inv} — average inventory holding cost for all retailers over the m-day period
- TC_{trp} — average traveling cost for all tours over the m-day period
- TC — average system-wide costs over the m-day period
- D_i — total demand of retailer $i \in I$ over the m-day period
- F — set of allowable visit frequencies
- S — set of allowable visit-day combinations
- q_i — the delivery size of retailer $i \in I$ at any visit
- b_i — the earliest possible departure time at vertex $i \in I_0$
- e_i — the latest possible arrival time at vertex $i \in I_0$
- s_{itk} — the start of service at vertex $i \in I_0$ by vehicle $k \in K$ on day $t \in T$
- d_i — the service time required when a vehicle visits vertex $i \in I_0$
- C — maximum capacity of any vehicle
- R — maximum tour duration of any tour
- H_{tk} — the tour served by vehicle $k \in K$ on day $t \in T$
- x_{ijtk} — $\begin{cases} 1 & \text{if vehicle } k \in K \text{ visits vertex } j \in I_o \text{ just after vertex } i \in I_o \text{ on day } t \in T. \\ 0 & \text{otherwise.} \end{cases}$
- y_{it} — $\begin{cases} 1 & \text{if vertex } i \in I_o \text{ is visited by any vehicle on day } t \in T. \\ 0 & \text{otherwise.} \end{cases}$
- f_i — visit frequency of retailer $i \in I$ during the m-day period

2.4.1. Objective function

The objective function of the IPVRPTW is to minimize the average system-wide costs (TC) which is the sum of the average inventory holding cost (TC_{inv}) and the average traveling cost (TC_{trp}) over the m-day period.

As stated earlier, unlike in the PVRP or the PVRPTW, f_i, the visit frequency of retailer $i \in I$ in the IPVRPTW, is now treated as a decision variable. The average inventory holding cost of retailer $i \in I$ over the m-day period, $C_{\text{inv}(i)}$ then can be formulated as a function of f_i.

$$C_{inv(i)}(f_i) = \frac{hD_i}{2f_i}, \quad i \in I;\ f_i \in F \tag{5}$$

Accordingly, TC_{inv} can be seen as a function of $(f_1, f_2, \ldots, f_n)$.

$$TC_{\text{inv}}(f_1, f_2, \ldots, f_n) = \sum_{\forall i \in I} \frac{hD_i}{2f_i}, \quad f_i \in F \tag{6}$$

Likewise, TC_{trp} can also be viewed as a function of $(f_1, f_2, \ldots, f_n)$ as follows:

$$TC_{\text{trp}}(f_1, f_2, \ldots, f_n) = \frac{1}{m} \sum_{\forall i \in I_0} \sum_{\forall j \in I_0} \sum_{\forall t \in T} \sum_{\forall k \in K} c_{ij} x_{ijtk}, \quad f_i \in F \tag{7}$$

The formulation of TC then can be described as follows:

$$TC(f_1, f_2, \ldots, f_n) = \text{Min}\left(\sum_{\forall i \in I} \frac{hD_i}{2f_i} + \frac{1}{m} \sum_{\forall i \in I_0} \sum_{\forall j \in I_0} \sum_{\forall t \in T} \sum_{\forall k \in K} c_{ij} x_{ijtk} \right), \quad f_i \in F \tag{8}$$

2.4.2. Stationary-interval property constraint

Constraints (9) guarantee that each retailer must be served as many as the assigned visit frequency. Constraints (10) ensure that each retailer is visited only on the days corresponding to the assigned visit-day combination and visit frequency. Eqs. (9) and (10) are required to fulfill the stationary-interval property.

$$\sum_{\forall t \in T} y_{it} = f_i, \quad \forall i \in I; f_i \in F \tag{9}$$

$$\sum_{r=t+1}^{t+\frac{m}{f_i}} y_{ir} = 1, t = 0, \ldots, \left(m - \frac{m}{f_i}\right), \quad \forall i \in I; f_i \in F \tag{10}$$

2.4.3. Vehicle capacity

Constraints (11) guarantee schedule feasibility with respect to the vehicle capacity aspect. The load of the vehicle must not exceed the vehicle capacity during a tour.

$$\sum_{\forall i \in I_0} \sum_{\forall j \in I_0} q_i x_{ijtk} \leqslant C, \quad \forall t \in T; \forall k \in K \tag{11}$$

2.4.4. Tour duration

Constraints (12) ensure schedule feasibility with respect to the tour duration aspect. The duration of a tour must not exceed the maximum tour duration R.

$$\sum_{\forall i \in I_0} \sum_{\forall j \in I_0} (t_{ij} + d_i) x_{ijtk} \leqslant R, \quad \forall t \in T; \forall k \in K \tag{12}$$

2.4.5. Time-windows

Constraints (13)–(15) guarantee schedule feasibility with respect to time window constraint. Note that M_{ij} are large constants and can be replaced by $\max\{b_i + d_i + t_{ij} - e_j, 0\}$, $i,j \in I_0$.

$$s_{itk} + d_i + t_{ij} - s_{jtk} \leqslant (1 - x_{ij}) M_{ij}, \quad \forall i,j \in I_0; \forall t \in T; \forall k \in K \tag{13}$$

$$b_i \leqslant s_{itk} \leqslant e_i, \quad \forall i \in I_0; \forall t \in T; \forall k \in K \tag{14}$$

$$S_{0tk} = d_0 = 0, \quad \forall t \in T; \forall k \in K \tag{15}$$

2.4.6. Standard constraints

The following standard constraints of the PVRP were adapted from Cordeau et al. (1997). Eq. (16) guarantees that each vehicle, which visits a retailer on a given day, also leaves that retailer on the same day. Eq. (17) guarantees that each vehicle can only be used once at any day. Eq. (18) is standard sub-tour elimination constraints. Eqs. (19) and (20) ensure binary values for the solution.

$$\sum_{\forall i \in I_0} x_{irtk} - \sum_{\forall j \in I_0} x_{rjtk} = 0, \quad \forall r \in I_0; \ \forall t \in T; \ \forall k \in K \tag{16}$$

$$\sum_{\forall j \in I} x_{rjtk} \leqslant 1, \quad r = 0; \ \forall t \in T; \ \forall k \in K \tag{17}$$

$$\sum_{i \in B} \sum_{j \in B} x_{ijtk} \leqslant |B| - 1, \quad \forall t \in T; \ \forall k \in K; \quad B \subseteq I; \ |B| \geqslant 2 \tag{18}$$

$$x_{ijtk} \in \{0,1\}, \quad \forall i,j \in I_0; \ \forall t \in T; \ \forall k \in K \tag{19}$$

$$y_{it} \in \{0,1\}, \quad \forall i \in I_0; \ \forall t \in T \tag{20}$$

3. Algorithms

3.1. Overview

Since the IPVRPTW generalizes the PVRPTW, which is known a NP-hard problem (Cordeau et al., 2001), therefore we aim to develop heuristic algorithms for solving the IPVRPTW. In the following sections, the two phases used by the heuristics are explained: the initialization phase (I) and the improvement phase (II).

Before describing the phases, let first define a "node" as a visit at retailer $i \in I$ performed by a vehicle on a day. Thus, each retailer $i \in I$ has $f_i \in F$ node(s) and the total number of nodes N can be calculated as in Eq. (21).

$$N = \sum_{\forall i \in I} f_i. \tag{21}$$

Let also define n_x as the number of retailers, $i \in I$ which has the visit frequency $f_i = x$; $x \in F$. Accordingly, the following relationship holds:

$$\sum_{\forall x \in F} n_x = N \tag{22}$$

3.2. Phase I: initialization

The objective of the initialization phase is to determine the "best" visit frequency which can be feasibly assigned to each retailer $i \in I$. This phase can be divided into four steps. The following is a general procedure of this phase.

The algorithm starts from a situation where each retailer $i \in I$ is set to have the lowest allowable visit-frequency ($f_i = 1$). Thus, the delivery size of each retailer $i \in I$ equals to its total demand over the m-day period, $q_i = D_i$. The data set is then sorted in non-increasing fashion of D_i in order to avoid difficulties in the future.

In the second step, we choose a set of mK seed nodes to initialize each tour H_{tk}, $\forall t \in T$, $\forall k \in K$. The seed nodes are selected in such a way so that they will be geographically well scattered. A number of mK simple tours then are built by linking each seed node to the depot.

After that, we apply the cheapest insertion heuristic method to assign the unrouted nodes to the tours. In any iteration, for each unrouted node, we compute the feasible insertion cost for all adjacent nodes of the current tours. The unrouted node having the minimum feasible insertion cost is then inserted between the corresponding adjacent nodes of the tour. The iteration is repeated until all nodes have been inserted into the tours.

In the last step, we perform the *Visit Frequency Optimization Procedure*. The procedure is carried out by gradually increasing the visit frequency of each retailer $i \in I$ until the highest allowable visit frequency, $f_i = m$, so that the current objective function can be reduced. This procedure is performed 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, there are 3 stages in the 6-day period and the target visit frequency of stage 1 is 2, while those of stages 2 and 3 are 3 and 6, respectively.

On one hand, the increase of the visit frequency of retailer $i \in I$ will TC_{inv}. Let $\Delta TC_{\text{inv}(i)}$ be defined as the decrease in the inventory holding cost because of the increase of the visit frequency of retailer $i \in I$ from $f_i = \lambda$ (the current visit frequency) to $f_i = \mu$ (the target visit frequency). Since $\lambda < \mu$, the value of $\Delta TC_{\text{inv}(i)}$ is always positive. The formulation of $\Delta TC_{\text{inv}(i)}$ can be expressed as follows:

$$\Delta TC_{\text{inv}(i)} = \frac{hD_i}{2}\left(\frac{1}{\lambda} - \frac{1}{\mu}\right), \quad i \in I : \lambda, \mu \in F \tag{23}$$

On the other hand, the increase of the visit frequency of retailer $i \in I$ from $f_i = \lambda$ to $f_i = \mu$ will also increase TC_{trp}. Let $\Delta TC_{\text{trp}(i)}$ be defined as the increase in the traveling cost because of this increase of visit frequency. The value of $\Delta TC_{\text{trp}(i)}$ depends on the move used for increasing the number of nodes in the graph.

The move symbolized by $M_{1(i)}$ can be described as follows. First, we delete all λ nodes of retailer i from their current corresponding tours. We then generate a number of $(\mu - \lambda)$ new nodes and insert all of these nodes simultaneously to the tours associated with the allowable visit-day combinations for $f_j = \mu$ at the minimum feasible insertion cost. Fig. 5 shows move $M_{1(i)}$ to increase the visit frequency from $f_i = 2$ to $f_i = 3$.

Let $VC_{1(i)}$ denote the insertion cost of move $M_1(i)$ which can be defined as the insertion cost of the μ nodes in the new corresponding tours minus the savings obtained by removing λ nodes from the old corresponding tours in order to obtain the shortest tour. Then, $\Delta TC_{\text{trp}(i)}$ can be formulated as follows:

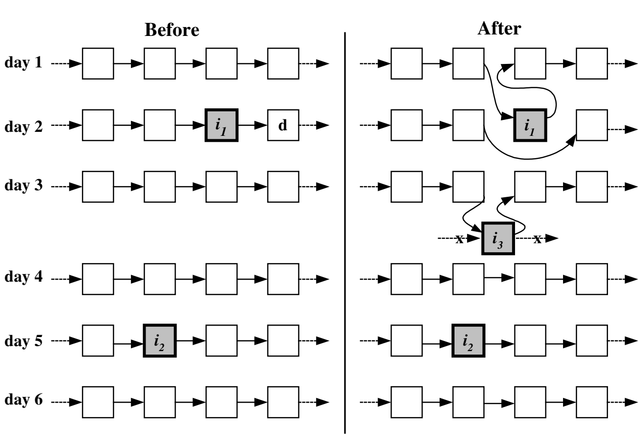

Fig. 5. Increasing the visit frequency from $f_i = 2$ to 3.

$$\Delta TC_{\text{trp}(i)} = \frac{VC_{1(i)}}{m}, \quad i \in I \tag{24}$$

Note that since the algorithms are designed to find the shortest tour, therefore, it does not mean the increase of f_i will automatically increase TC_{trp}.

The changes in the inventory holding and traveling costs will alter the system-wide costs. Let define $\Delta TC_{(i)}$ as the difference between the decrease in the inventory holding cost and the increase in the traveling cost. The formulation of $\Delta TC_{(i)}$ is expressed as follows:

$$\Delta TC_{(i)} = \omega_{\text{inv}} \Delta TC_{\text{inv}(i)} - \omega_{\text{trp}} \Delta TC_{\text{trp}(i)}, \quad i \in I \tag{25}$$

where ω_{inv} and ω_{trp}; $\omega_{\text{inv}} + \omega_{\text{trp}} = 1$ are two subjective weight parameters for the inventory holding cost and the traveling cost, respectively. The larger the ω_{inv}, the more emphasis is put on the inventory holding cost, and vice versa.

At an iteration of a stage, we evaluate all retailers that may feasibly increase their current visit frequency to the target visit frequency. We then select a retailer, say retailer $j \in I$, where $\Delta TC_{(j)} = \max_{i \in I}\{\Delta TC_{(i)}\}$, $\Delta TC_{(j)} > 0$, to increase its visit frequency at this iteration. The remaining retailers will keep their current visit frequency. The delivery size of retailer j then will be adjusted to the new delivery size q'_j which is:

$$q'_j = \frac{\lambda}{\mu} q_j, \quad j \in I; \lambda < \mu \in F \tag{26}$$

Owing to the increase of the visit frequency of retailer j, the number of retailers serviced λ times n_λ decreases by one, while the number of retailers serviced μ times n_μ increases by one. Finally, the number of retailer nodes N increases by $(\mu - \lambda)$. The iteration is repeated until there is no more retailer that can feasibly increase its current visit frequency to the target visit frequency in this stage. Otherwise, we go to the next stage to do the same procedure.

The step-by-step algorithm of this phase is presented as follows.

Step 1.1. *Dataset definition*
Define the dataset where each retailer $i \in I$ assumes having $f_i = 1$, and sort the dataset in non-increasing fashion of the total demand D_i.
Step 1.2. *Seed retailer determination*

a. Let define L be the set of seed retailers, with initially $L = \emptyset$.
b. For each tour H_{tk}, $\forall t \in T$, $\forall k \in K$:
   - For each retailer i, $\forall i \in I$:
     - Do $L = L \cup r^*$,
     - where $\begin{cases} r^* = \arg\max\limits_{i \in I,\ i \notin L}(c_{i0}), \ \forall i \in I \text{ if } L = \emptyset \\ r^* = \arg\max\limits_{i \in I,\ i \notin L}(c_{i0}\prod_{r \in L} c_{ir}) \text{ if } L > 1 \end{cases}$

- Set y^* be the seed retailer of tour H_{tk}. $t \in T$, $k \in K$.
- Build an initial tour consisting of seed retailer y^* and the depot such that $H_{tk} = \{(0, y^*), (y^*, 0)\}$, $t \in T$, $k \in K$

Step 1.3. *Initial allocation*

a. Find an unrouted node, call it z.
b. Find an edge (i,j) in H_{tk}. $t \in T$, $k \in K$ such that $c_{iz} + c_{zj} - c_{ij}$ is minimal.
c. Rebuild H_{tk}, $t \in T$, $k \in K$ by replacing (i,j) with $\{i, z\}$ and $\{z, j\}$.

Repeat steps 1.3a–c until there are no more unrouted retailers.

Step 1.4. *Visit frequency optimization*

The following is the algorithm to increase the visit frequency of a retailer in a stage (note λ represents the current visit frequency while μ is the target visit frequency, $\lambda < \mu \in F$).

a. Determine ω_{inv} and ω_{trp}
b. For each retailer $i \in I$:
   - Calculate $\Delta TC_{\text{inv}(i)}$
   - Choose the best feasible move $M_{1(i)}$
   - Calculate $\Delta TC_{\text{trp}(i)}$
   - Calculate $\Delta TC_{(i)}$
c. Select retailer $j \in I$ whose $\Delta TC_{(j)} = \max_{i \in I} \{\Delta TC_{(i)}\}$ and $\Delta TC_{(j)} > 0$.
d. Increase the visit frequency of retailer j from $f_j = \lambda$ to $f_j = \mu$, and do move $M_{1(j)}$.
e. Update $q_j = \dfrac{\lambda}{\mu} q_j$.
f. Let $n_\mu = n_\mu + 1$, $n_\lambda = n_\lambda - 1$ and $N = N + (\mu - \lambda)$
g. Repeat steps 1.4a–e until there are no more positive values of $\Delta TC_{(i)}$ produced, otherwise perform local optimization separately for the changed tours and go to next stage (phase).

3.3. Phase II: improvement

In this phase, we employ several interchange heuristics that can improve the current solution. First, we try to alter the visit-day combination of each retailer $i \in I$ in order to reduce traveling cost. To give further improvement, we then perform local optimization procedure.

3.3.1. Visit-day combination interchanges

The Tabu Search (TS) algorithm (see Glover, 1986) is used in the visit-day combination interchanges. Starting from the initial solution obtained from the first phase, in any iteration the algorithm moves from the current solution to the best solution in a neighborhood of the current solution. The neighborhood is defined as a blend of the most favorable local search moves that transforms one solution to another (Tarantilis & Kiranoudis, 2002).

To avoid being caught in local optimum and having cycling, solutions possessing some attributes of recently visited solutions are considered as tabu, for a number of iterations. This number of iterations is called *tabu length* and determined based on the experiences. In order to do this, the algorithm exploits a short-term memory called *tabu list*. The tabu list denoted by Γ, keeps track of recently visited solutions or their attributes. A move to a neighboring solution is permitted if the neighboring solution is neither contained in the tabu list nor possesses an identical attribute to a solution in that list. Every move that has been successfully performed, it will be inserted into the end of the tabu list, while at the same time the oldest element of this list is deleted.

At any iteration, the algorithm explores the entire neighborhood to select the allowable non-tabu moves providing the highest saving in traveling cost. The stage is repeated until there are no more visit-day interchanges that can further reduce the traveling cost. Otherwise, we go the tour interchanges procedure.

Here, we consider two types of move called combination-insertion move and combination-exchange move. The combination-insertion move symbolized by $M_2(i)$ is an operation that simultaneously deletes a number of f_i nodes of retailer $i \in I$ from their corresponding current tours, and then reinserts all of them into the other corresponding tours of the new visit-day combination. Let $VC_2(i)$ be defined as the saving in traveling cost that can be obtained by performing $M_2(i)$. The combination-exchange move symbolized by $M_3(i)$ is an operation that two retailers i and $j \in I$ with the same visit frequency, $f_i = f_j$ and different visit-day combination, simultaneously insert all of their nodes into each other's corresponding former tours, but not necessarily into the former locations of each other. Let also define $VC_3(i)$ as the saving in traveling cost that can be obtained by performing $M_3(i)$. Figs. 6 and 7 show the examples of the combination-insertion and combination-exchange moves, respectively. Fig. 6 visualizes that the visit-day combination of retailer j has changed from 101010 to 010101. Fig. 7 presents that the visit-day combination

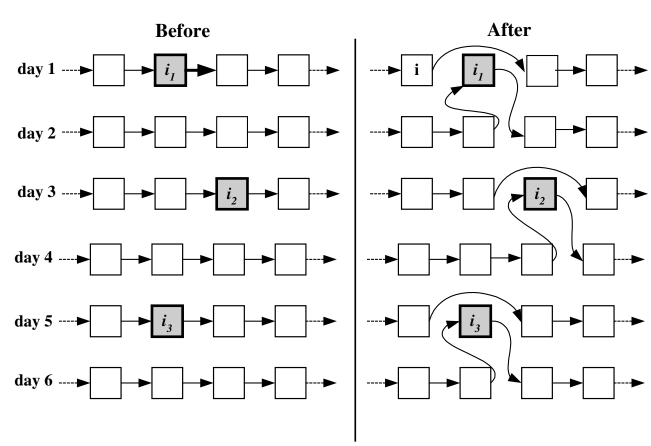

Fig. 6. Example of the combination-insertion move.

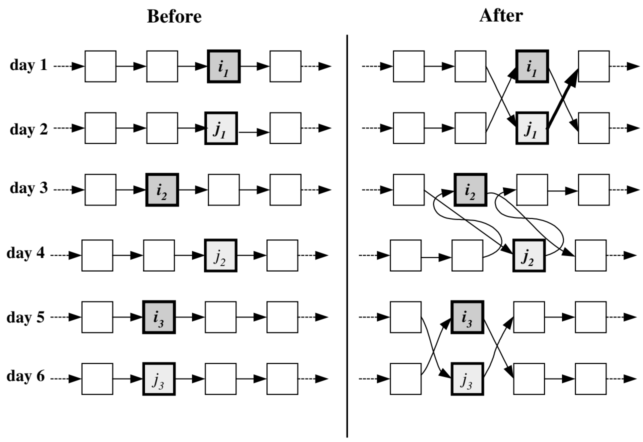

Fig. 7. Example of the combination-exchange move.

of retailer i has changed from 101010 to 010101 while the visit-day combination of retailer j has changed from 010101 to 101010.

It should be noted that we do not consider any retailer $i \in I$ with $f_i = m$ in the visit-day combination interchanges, since they have only one option of visit-day combination. Moreover, to reduce computational efforts, any retailer $i \in I$ with $f_i = 1$ are not considered in this step because such moves will be addressed in the tour interchanges procedure.

3.3.2. Tour interchanges

The TS heuristic is again employed in this step. However, we consider only single-node moves which can be defined as follows:

- $M_{4(i)}$ is an operation that transfers a single node of retailer $i \in I$ from its position in one tour to another position in the same tour at the minimum feasible insertion cost.
- $M_{5(i)}$ is an operation that transfers a node of retailer $i \in I$ from its position in one tour to another position in a different tour but in the same day at the minimum feasible insertion cost.
- $M_{6(i)}$ is an operation that employs 2-opt exchange move (Lin, 1965). This move is to exchange two arcs in a tour, say wx and yz with wy and xz at the minimum feasible insertion cost, This move also reverses the direction of the tour between nodes x and y to eliminate criss-cross in a tour.

Let also define $VC_{4(i)}$, $VC_{5(i)}$, $VC6_{(i)}$ as the savings in traveling cost resulted from $M_{4(i)}$, $M_{5(i)}$ and $M_{6(i)}$, respectively.

The following is the step-by-step algorithm for the improvement stage.

Step-by-step Algorithms: Repeat the following steps until no more reduction of the objective function can be performed:

Step 2.1. *Visit-day combination interchanges*

a. Set $\Gamma = \emptyset$
b. For each retailer $i \in I$, generate all possible moves $M_{2(i)}$ in the neighborhood, and find the maximum $VC_{2(i)}$ excluding the moves listed on the tabu list Γ
c. For each retailer $i \in I$, generate all possible moves $M_{3(i)}$ in the neighborhood, and find the maximum $VC_{3(i)}$ excluding the moves listed on the tabu list Γ
d. Select the retailer $s \in I$ so that $VC_{2(s)} = \max_{i \in I}\{VC_{2(i)}\}$
e. Select the retailer $r \in I$ so that $VC_{3(r)} = \max_{i \in I}\{VC_{3(i)}\}$
f. If $VC_{2(s)} > VC_{3(r)}$, $s,r \in I$ then do $M_{2(s)}$ otherwise do $M_{3(r)}$
g. Update the tabu list Γ
h. Repeat steps 2.1b–g until there is no more feasible $M_{2(i)}$ or $M_{3(i)}$, $\forall i \in I$

Step 2.2. *Tour interchanges*

Do the following for each move type-x consecutively starting from $x = 4$, 5, and 6:

a. Set $\Gamma = \emptyset$
b. For each retailer $i \in I$, generate all possible moves $M_{x(i)}$, and find the maximum $VC_{x(i)}$ excluding the moves listed on the tabu list Γ
c. Select the retailer $s \in I$ so that $VC_{x(s)} = \max_{i \in I}\{VC_{x(i)}\}$, and do $M_{x(s)}$.
d. Update the tabu list Γ
e. Repeat steps 2.2b–d until there is no more feasible $M_{x(i)}\,\forall i \in I$

3.4. Variants of heuristics

Based on the stages of the visit frequency optimization procedure, we build two basic variants of heuristics whose called INC-1 and INC-2. Heuristic INC-1 considers only the retailers whose current visit frequency equals to the immediate-lower possible number of the target visit frequency of the stage. In the opposite, heuristic INC-2 is designed in a slightly different way. In heuristic INC-2, the current visit frequency is not necessarily the immediate possible lower number of the target visit frequency of the stage. Fig. 8 show the transition diagrams of both heuristics. We may observe here that heuristic INC-2 has wider searching spaces than heuristic INC-1.

Moreover, we develop two other variants of heuristics enhancing the basic heuristics. The enhanced heuristics are named as heuristics *INC-1-plus* and INC-2-plus,

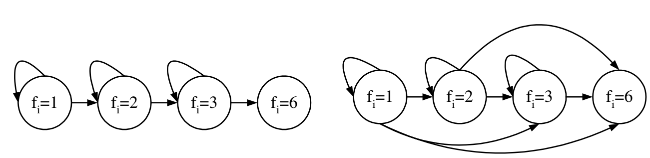

Fig. 8. Transition diagrams of heuristics INC-1 and INC-2.

respectively. Different from the previous ones, in INC-1-plus and INC-2-plus, we also consider visit frequencies changes in the improvement phase by adding the visit frequency optimization procedure (step 1.4) before step 2.2.

4. Numerical experiments

To evaluate the performance of our heuristics, we conducted several numerical experiments with the data instances of the PVRPTW generated by Cordeau et al. (2001). The algorithms were coded in C-language and run on an Intel Pentium 4, 2800 MHz personal computer. We have selected 8 data instances of the 6-day period case whose visit frequencies and visit-day combinations fulfill the stationary property.

These instances can be divided into two groups. The first group consists of instances Pr07a, Pr08a, Pr09a, and Pr10a that have narrow time windows, while the second group consists of instances Pr07b, Pr08b, Pr09b, and Pr10b that have larger time windows. The characteristics of these instances are shown in Table 2. Note that $U[x, y]$ represents a uniform random number in the interval $[x, y]$.

To use these instances, we calculated the total demand D_i for each retailer $i \in I$, which is the multiplication of the delivery size and the visit frequency in the original instances. Other parameters such as retailers' geographical coordinates, time windows, and service times, and the number of vehicles were not changed. We assume that inventory holding cost per unit per day h equals to 1, and any fraction numbers of the delivery size, q_i, $\forall i \in I$, is allowed.

First, we conducted several experiments to select the most appropriate values of parameters ω_{inv} and ω_{trv} through sensitivity analyses for all heuristics considered (INC-1, INC-2, INC-1 plus, INC-2 plus). We used five

Table 2
Characteristics of data instances

Problem	n	n_x				N	K	C	R	b_i	e_i
		1	2	3	6						
Pr07a	72	18	18	18	18	216	5	500	200	$U[60, 480]$	$U[b_i + 90, b_i + 180]$
Pr08a	144	36	36	36	36	432	10	475	190	$U[60, 480]$	$U[b_i + 90, b_i + 180]$
Pr09a	216	54	54	54	54	648	15	450	180	$U[60, 480]$	$U[b_i + 90, b_i + 180]$
Pr10a	288	72	72	72	72	864	20	425	170	$U[60, 480]$	$U[b_i + 90, b_i + 180]$
Pr07b	72	18	18	18	18	216	4	500	200	$U[60, 300]$	$U[b_i + 180, b_i + 360]$
Pr08b	144	36	36	36	36	432	8	475	190	$U[60, 300]$	$U[b_i + 180, b_i + 360]$
Pr09b	216	54	54	54	54	648	12	450	180	$U[60, 300]$	$U[b_i + 180, b_i + 360]$
Pr10b	288	72	72	72	72	864	16	425	170	$U[60, 300]$	$U[b_i + 180, b_i + 360]$

Table 3
Selected parameters ω_{inv} and ω_{trp}

Heuristic	$\omega_{\text{inv}}/\omega_{\text{trp}}$	Problem							
		Pr07a	Pr08a	Pr09a	Pr10a	Pr07b	Pr08b	Pr09b	Pr10b
INC-1	0.7/0.3				√				√
	0.6/0.4	√	√	√			√	√	
	0.5/0.5					√			
	0.4/0.6								
	0.3/0.7								
INC-1-plus	0.7/0.3	√			√				√
	0.6/0.4		√	√			√	√	
	0.5/0.5					√			
	0.4/0.6								
	0.3/0.7								
INC-2	0.7/0.3			√					√
	0.6/0.4	√	√		√	√	√	√	
	0.5/0.5								
	0.4/0.6								
	0.3/0.7								
INC-2-plus	0.7/0.3			√	√				√
	0.6/04	√	√			√	√	√	
	0.5/0.5								
	0.4/0.6								
	0.3/0.7								

different combinations of ω_{inv} and ω_{trv}, which are (i) $\omega_{\text{inv}} = 0.7 - \omega_{\text{trp}} = 0.3$, (ii) $\omega_{\text{inv}} = 0.6 - \omega_{\text{trp}} = 0.4$, (iii) $\omega_{\text{inv}} = 0.5 - \omega_{\text{trp}} = 0.5$, (iv) $\omega_{\text{inv}} = 0.4 - \omega_{\text{trp}} = 0.6$, and (v) $\omega_{\text{inv}} = 0.3 - \omega_{\text{trp}} = 0.7$. Among these combinations we choose the combination that produces the best solution for each instance. The results of these experiments are summarized in Table 3. For example, the best solutions of instances Pr07a, Pr08a and Pr09a for INC-1 were produced by the parameters $\omega_{\text{inv}} = 0.6$ and $\omega_{\text{trp}} = 0.4$, while the best solution of instance Pr08a for heuristic INC-1-plus were produced by the parameters $\omega_{\text{inv}} = 0.5$ and $\omega_{\text{trp}} = 0.5$.

The complete results of the four heuristics are displayed in Table 4. The results show that heuristic INC-1 outperforms the other heuristics in one instance (Pr10b). Heuristic INC-1-plus yields the best results in four instances (Pr07a, Pr08a, Pr09a, Pr10a) while heuristic INC-2-plus provides the best results in three instances (Pr10a, Pr08b, Pr09b). No best solution can be obtained from heuristic INC-2.

Table 5 shows the relative order of the four heuristics, from best to worse in terms of mean ranks is INC-1-plus, INC-2-plus, INC-1, INC-2. However, there is no consistent pattern in the ranking of the four heuristics. Our statistical analyses support the findings. We conducted the Friedman test (see Golden & Stewart, 1985)

Table 5
Mean ranks

Problem	INC-1(%)	INC-1-plus (%)	INC-2 (%)	INC-2-plus (%)
Pr07a	2	1	3.5	3.5
Pr08a	3	1	4	2
Pr09a	2	1	4	3
Pr10a	2	1	3.5	3.5
Pr07b	4	3	2	1
Pr08b	4	2	3	1
Pr09b	4	3	2	1
Pr10b	1	2	3	4
Mean ranks	2.75	1.75	3.13	2.38

Table 4
Results of the experiments

Problem	Heuristic	ω_{inv}	ω_{trp}	n_x				N	TC_{inv}	TC_{trp}	TC	CPU time (seconds)
				1	2	3	6					
Pr07a	INC-1	0.6	0.4	15	56	1	0	130	742.5	643.8	1386.30	4.67
	INC-1-plus	0.7	0.3	7	31	31	3	180	578.5	797.78	1376.28	8.34
	INC-2	0.6	0.4	14	56	2	0	132	741.83	644.77	1386.61	5.36
	INC-2-plus	0.6	0.4	14	56	2	0	132	741.83	644.77	1386.61	9.31
Pr08a	INC-1	0.6	0.4	15	84	29	16	366	1220.08	1282.17	2502.25	125.27
	INC-1-plus	0.6	0.4	13	77	36	18	383	1175	1308.96	2483.96	140.98
	INC-2	0.6	0.4	15	77	28	24	397	1154.17	1351.46	2505.63	133.91
	INC-2-plus	0.6	0.4	12	77	29	26	409	1135.33	1364.64	2499.97	196.72
Pr09a	INC-1	0.6	0.4	26	106	56	28	574	1669.67	1839.36	3509.02	540.03
	INC-1-plus	0.6	0.4	19	110	56	31	593	1626.00	1864.47	3490.47	638.31
	INC-2	0.7	0.3	10	101	61	44	659	1420.33	2098.92	3519.25	1369.23
	INC-2-plus	0.7	0.3	8	90	67	51	695	1376.67	2133.89	3510.56	1424.91
Pr10a	INC-1	0.7	0.3	18	126	107	37	813	2066.42	2650.73	4717.15	2410
	INC-1-plus	0.7	0.3	16	123	91	58	883	1927.25	2788.61	4715.86	2816.26
	INC-2	0.6	0.4	30	151	51	56	821	2189.25	2545.85	4735.10	2926.02
	INC-2-plus	0.7	0.3	14	112	108	54	886	1945.75	2789.35	4735.10	3227.45
Pr07b	INC-1	0.5	0.5	11	47	8	6	165	630.50	636.24	1266.74	5.05
	INC-1-plus	0.5	0.5	9	49	8	6	167	622.50	632.61	1255.11	6.27
	INC-2	0.6	0.4	7	39	23	3	172	596.33	647.43	1243.76	8.7
	INC-2-plus	0.6	0.4	7	38	24	3	173	594.33	648.19	1242.52	9.88
Pr08b	INC-1	0.6	0.4	16	77	35	16	371	1208.83	1224.47	2433.31	128.14
	INC-1-plus	0.6	0.4	13	76	35	20	390	1170	1248.76	2418.76	143.36
	INC-2	0.6	0.4	11	75	37	21	398	1131.58	1287.96	2419.54	232.28
	INC-2-plus	0.6	0.4	8	63	48	25	428	1045.67	1349.79	2395.46	259.74
Pr09b	INC-1	0.6	0.4	8	104	77	27	609	1613.67	1676.03	3289.7	578.03
	INC-1-plus	0.6	0.4	8	102	77	29	617	1610.5	1674.88	3285.38	886.94
	INC-2	0.6	0.4	7	93	87	29	628	1523.42	1700.58	3224	1226.76
	INC-2-plus	0.6	0.4	6	91	87	32	641	1510.58	1712.35	3222.94	1403.00
Pr10b	INC-1	0.7	0.3	16	136	85	51	849	2188.25	2324.88	4513.13	2791.48
	INC-1-plus	0.7	0.3	16	130	68	74	924	2101.25	2419.28	4520.53	3191.95
	INC-2	0.7	0.3	13	129	109	37	820	2256.92	2370.14	4627.06	3631.28
	INC-2-plus	0.7	0.3	12	124	110	42	842	2243.33	2384.82	4628.15	3809.23

Table 6
Performance comparisons based on the best-known PVRPTW solutions

Problem	TC_{inv}	TC_{trp}	TC	INC-1 (%)	INC-1-plus (%)	INC-2 (%)	INC-2-plus (%)
	Cordeau et al. (2001)						
Pr07a	474	1165.26	1639.26	−15.43	−16.04	−15.41	−15.41
Pr08a	1003	1674.18	2677.18	−6.53	−7.22	−6.41	−6.62
Pr09a	1368	2382.5	3750.5	−6.44	−6.93	−6.17	−6.40
Pr10a	1925	3101.62	5026.62	−6.16	−6.18	−5.80	−5.80
Pr07b	474	934.35	1408.35	−10.06	−10.88	−11.69	−11.77
Pr08b	1003	1331.27	2334.27	4.24	3.62	3.65	2.62
Pr09b	1368	1848.32	3216.32	2.28	2.15	0.24	0.21
Pr10b	1925	2367.94	4292.94	5.13	5.30	7.78	7.81
Average				−4.12%	−4.52%	−4.22%	−4.42%

to evaluate the rankings of these four heuristics. The null hypothesis was that the performance of all four heuristics are equal at $\alpha = 0.05$. The test statistic obtained was 1.88, which is lower than $F_{0.05,3,21} = 3.07$. Thus, we cannot reject the null hypothesis. We then can conclude that there is no difference in the performances of all heuristics. It also means that we have no reason to perform a further analysis to identify "the best" of the four heuristics.

In terms of computational time as shown in Table 4, the two basic heuristics (INC-1 and INC-2) are always better than the enhanced heuristics (INC-1-plus and INC-2-plus). The results show that the computational time has increased significantly when we consider visit frequencies changes in the improvement phase. Moreover, the computational time required by heuristic INC-2 is always higher than that of heuristic INC-1. It can be easily understood since the space of searching of heuristic INC-2 is wider than that of heuristic INC-1. Similar situation is also found in the comparison of the computational time required by INC-1-plus and INC-2-plus. Heuristic INC-1-plus is always better than heuristic INC-2-plus.

Finally, in order to obtain a meaningful evaluation, we compare among the four heuristics based on the best-known solutions of PVRPTW provided by Cordeau et al. (2001). Although the objectives of the IPVRPTW and the PVRPTW are different, such a comparison was made since the PVRPTW solutions can also be viewed as feasible solutions of the IPVRPTW. The comparison is similar to that of Federgruen and Zipkin (1984). They compared their IRP solutions to the VRP solutions in the literature. It should be noted here that our aim is not to compare the performance of our heuristics and that of the literature.

Table 6 displays this comparison where column "%" represents the percentage deviations between TC in our solutions and in Cordeau et al. (2001). The value of TC_{inv} in this literature was calculated using Eq. (5), and the value of TC_{trp} was obtained from their PVRPTW solutions divided by the period length ($m = 6$).

From the comparison showed in this table, we may observe that the four heuristics can produce substantial savings for all instances with narrow time windows (Pr07a, Pr08a, Pr09a, Pr10a) and the smallest-size instances with larger time-windows (Pr07b). Heuristic INC-1 can reduce TC ranging from 6.16% to 15.43% with the average of 8.92% while heuristics INC-1-plus can reduce TC ranging from 6.18% to 16.04% with the average of 9.45%. Moreover, both heuristic INC-2 and INC-2-plus can produce reductions of TC ranging from 5.80% to15.41%. In the average, heuristic INC-2-plus (9.20%) is better than heuristic INC-1 (9.10%). Unfortunately, the four heuristics were failed to yield better solutions compared to the literature for the other three size instances with larger time-windows (Pr08b, Pr09b, Pr10b).

Finally, the relative order of the four heuristics from best to worse compared to the best-known solutions of PVRPTW is slightly different from the previous comparison, which are INC-1, INC-2-plus, INC-2 and INC-1.

5. Conclusions

We have developed an inventory routing problem (IRP) model for the delivery of products in the vending machine supply-chains operated under vendor-managed inventory (VMI). The new model was built based on the existing PVRPTW model and called the Integrated Inventory and Periodic Vehicle Routing Problem with Time-windows (IPVRPTW). Two major enhancements are proposed. First, the objective has been changed to the minimization of the sum of the average inventory holding and traveling costs during the given m-day period. Second, the visit frequency is treated as a decision variable to obtain trade-offs between these costs. We built a mathematical formulation and four heuristics to solve it. The performance of the four heuristics were evaluated using a set of data instances. We observed that the performance of them were statistically indifferent. We also compared these results based on the best-known solutions of PVRPTW. The results show that substantial savings in system-wide costs can be achieved for most instances through incorporating inventory and vehicle routing decisions into a single model.

References

Anily, S., & Federgruen, A. (1990). One warehouse multiple retailer systems with vehicle routing costs. *Management Science, 36*(1), 92–114.

Annual report of the Japan Vending Machine Manufacturers Association, (2003) http://www.jvma.or.jp/pdf/fukyu.pdf(in Japanese).

Baptista, S. R. C., Oliveira & Zúquete, E. (2002). A period vehicle routing case study. *European Journal of Operational Research, 139*, 220–229.

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

Beltrami, E., & Bodin, L. (1974). Networks and vehicle routing for municipal waste collection. *Networks, 4*, 65–94.

Bramel, J., & Simchi-Levi, D. (1995). A location based heuristic for general routing problems. *Operations Research, 43*, 649–660.

Bramel, J., & Simchi-levi, D. (1997). *The logic of logistics: theory, algorithms and applications for logistics management*. New York: Springer.

Bräysy, O. (2003). A reactive variable neighborhood search algorithm for the vehicle routing problem with time window. *INFORMS Journal on Computing, 15*(4), 347–368.

Chao, I. M., Golden, B., & Wasil, E. (1995). An improved heuristic for the period vehicle routing problem. *Networks, 26*, 25–44.

Campbell, A., Clarke, L. W., & Savelsbergh, M. (2002). Inventory routing in practice. In P. Toth & D. Vigo (Eds.), *The vehicle routing problem* (pp. 22). SIAM Monographs on Discrete Mathematics and Applications.

Chien, T. W., Balakrishnan, A., & Wong, R. T. (1989). An integrated inventory allocation and vehicle routing problem. *Transportation Science, 23*(2), 67–76.

Christofides, N., & Beasley, J. E. (1984). The period routing problem. *Networks, 14*, 237–256.

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

Cordeau, J. F., Laporte, G., & Mercier, A. (2001). A unified tabu search heuristic for vehicle routing with time windows. *Journal of the Operational Research Society, 52*.

Dror, M., Ball, M., & Golden, B. (1986). A computational comparison of algorithms for the inventory routing problem. *Annals of Operations Research, 4*, 3–23.

Federgruen, A., & Zipkin, P. (1984). A combined vehicle routing and inventory allocation problem. *Operations Research, 32*, 1019–1037.

Gaudioso, M., & Paletta, G. (1992). A heuristic for the period vehicle routing problem. *Transportation Science, 26*, 86–92.

Gaur, V. & Fisher, M. L. (in press). A periodic inventory routing problem at a supermarket chain. *Operations Research*.

Ghiani, G., Laporte, G., & Musmanno, R. (2004). *Introduction to logistic system planning and control*. Halsted Press.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research, 1*, 53349.

Golden, B. L., Assad, A. A., & Dahl, R. (1984). Analysis of a large scale vehicle routing problem with an inventory component. *Large Scale Systems, 7*, 181–190.

Golden, B. L., & Stewart, W. R. (1985). Empirical analysis of heuristics. In E. L. Lawler, J. K. Lenstra, A. H. G. RinnoyKan, & D. B. Shmoys (Eds.), *The traveling salesman problem*. John Wiley and Sons.

Golden, B. L., & Wasil, E. A. (1987). Computerized vehicle routing in the soft drink industry. *Operations Research, 35*, 6–17.

Kleywegt, A. J., Nori, V. S., & Savelsbergh, M. W. P. (2004). Dynamic programming approximations for a stochastic inventory routing problem. *Transportation Science, 38*, 42–70.

Lin, S. (1965). Computer solutions of the traveling salesman problem. *Bell System Technical Journal, 65*, 2245–2269.

Russell, R., & Gribbin, D. A. (1991). Multiphase approach to the period routing problem. *Networks, 21*, 747–765.

Russell, R., & Igo, W. (1979). An assignment routing problem. *Networks, 9*, 1–17.

Silver, E. A., Pike, D. F., & Peterson, R. (1998). *Inventory management and production planning and scheduling* (Third ed.). John Wiley and Sons.

Tan, C. C. R, & Beasley, J. E. (1984). A heuristic algorithm for the period vehicle routing problem. *Omega, 12*, 497–504.

Tarantilis, C. D., & Kiranoudis, C. T. (2002). BoneRoute: an effective memory-based method for effective flee management. *Annals of Operations Research, 115*(1), 227–241.

Waller, M., Johnson, M. E., & Davis, T. (1999). Vendor managed inventory in the retail chain. *Journal of Business Logistics, 20*(1), 2002.

Zipkin, P. H. (2000). *Foundations of inventory management*. Boston: Irwin McGraw-Hill.