A Two-phase Solution Algorithm for the Inventory Routing Problem with Time Windows

Pantelis Z. Lappas[*], Manolis N. Kritikos, George D. Ioannou

Department of Management Science and Technology, School of Business, Athens University of Economics and Business, 47a Evelpidon Str., GR-11362, Athens, Greece

Abstract

The main objective of this paper is to propose a two-phase solution algorithm for solving the Inventory Routing Problem with Time Windows (IRPTW), which has not been excessively researched in the literature. The solution approach is based on (a) a simple simulation for the planning phase (Phase I) and (b) the Variable Neighborhood Search Algorithm (VNS) for the routing phase (Phase II). Testing instances are established to investigate algorithmic performance, and the computational results are then reported. The computational study underscores the importance of integrating the inventory and vehicle routing decisions. Graphical presentation formats are provided to convey meaningful insights into the problem.

Keywords: Routing, Inventory Routing Problem with Time Windows, Simple Simulation, Variable Neighborhood Search

1. Introduction

In recent years, the Inventory Routing Problem (IRP) has received a great deal of attention from academics, consultants and practitioners. The IRP arises from the application of the Vendor Managed Inventory (VMI) concept, where the supplier (vendor) has to make inventory and routing decisions simultaneously for a given planning horizon. Analytically, the supplier monitors the inventory levels of the customers and determines (a) the delivery times (i.e., when to visit his customers), (b) the quantities (i.e., how much to deliver to each of them when they are served), so that stock-outs are avoided, and (c) the set of routes used by a fleet of vehicles to serve a given set of customers (i.e., how to integrate the customers into the vehicle routes).

The Inventory Routing Problem with Time Windows (IRPTW) is a generalization of the standard IRP involving the added complexity that every customer should be served within a given time window. The IRPTW reflects a multi-functional problem that attempts to integrate two different functions within the supply chain network, i.e., planning and routing. In particular, planning is associated with the Inventory Control Problem (ICP), while routing is related to the Vehicle Routing Problem with Time Windows (VRPTW). The integration of ICP-VRPTW problems has scarcely been studied in the literature. Liu and Lee [1] proposed a two-phase heuristic method for solving the Inventory Routing Problem with Soft Time Windows (IRPSTW). The first phase of the heuristic algorithm finds an initial solution based on a construction approach, while the second phase improves the initial solution by adopting a variable neighborhood tabu search algorithm. In addition, Zeng and Zhao [2] represented the stochastic IRPSTW as a discrete time Markov decision process model and solved it by using dynamic programming approximations. Some applications in the context of IRPTW were presented by Zhang et al. [3], Li et al. [4] and Zhang et al. [5].

Generally, IRPs can be categorized into three levels [6,7]. The first categorization is based on the structural variants presented in IRPs, namely, product, time horizon, network topology, routing, inventory policy, inventory decisions, fleet composition and fleet size. The second categorization is related to the availability of information on the demand, reflecting several types of IRPs, for example, deterministic, stochastic, and dynamic and stochastic IRPs. Furthermore, the third categorization is associated with the chosen solution approach. According to Ballou [8] the modeling of supply chain and logistics problems has traditionally relied on three primary methods, i.e., simulation, optimization (exact algorithm) and heuristics, which can be divided into two categories [9]: classic heuristics and meta-heuristics. The recent literature has shown an increased interest in so-called matheuristics, methods that combine exact and heuristic approaches [10]. Bertazzi and Speranza [11] classified IRP decisions into two classes: (a) decisions over time only, in which

---

[*] Corresponding author. Tel: (+30) 2108 203 659
E-mail addresses: pzlappas@aueb.gr (P. Lappas), kmn@aueb.gr (M. Kritikos), ioannou@aueb.gr (G. Ioannou)

the delivery times and the quantities have to be determined at the same time, while the routes are given, and (b) decisions over time and space, where delivery times, quantities and routes have to be determined simultaneously. In addition, the optimal solution of an IRP depends on the objective function that has been chosen. As a result, an objective function can be (a) the sum of transportation costs only, (b) the sum of transportation and inventory holding costs of the customers or (c) the sum of transportation and inventory holding costs of the supplier and the customers.

In this paper, the main objective is to propose an approach for solving the IRPTW with the following characteristics. A single-product type has to be delivered by a fleet of capacitated homogenous vehicles (multiple vehicles) housed at a depot over a finite and discrete planning horizon. The network topology taken into account by the IRPTW model is one-to-many; that is, one supplier serves many geographically dispersed customers. A vehicle can visit more than one customer (multiple routing), while a vehicle's trip starts and ends at the depot (supplier). As far as the replenishment policy is concerned, an Order-up-to Level (OU) policy is considered, in which any customer has defined a maximum inventory level and every time a customer is served, the delivered quantity is such that the maximum inventory level at the customer is reached. It is assumed that the depot has a sufficient supply of products that can cover all customers' demands throughout the planning horizon. Moreover, the inventory is not allowed to become negative (fixed inventory) since the lowest inventory level is either fixed or equal to zero. With respect to the availability of information on customer demand, the proposed IRPTW model is deterministic since the demand is fully available to the supplier at the beginning of the planning horizon. Regarding the solution approach, a two-phase solution algorithm that combines a simple simulation and a Variable Neighborhood Search Algorithm (VNS) (i.e., a single-point search meta-heuristic) is presented to handle the IRPTW. The simple simulation is associated with the inventory allocation decisions (planning phase), while VNS is related to the routing decisions (routing phase). In addition, IRPTW decisions are decisions over time and space, while the objective function represents the sum of transportation costs only. This case corresponds to an environment in which the transportation cost represents the major cost component (e.g., the supplier and the customers represent entities of one and the same company).

The remainder of the paper is organized as follows. A problem description and mathematical formulation are presented in Section 2. The proposed solution approach is described and analyzed in detail in Section 3. Section 4 presents computational results, while in Section 5, conclusions and future research are given.

2. Problem Description and Mathematical Formulation

The IRPTW is a variation of the classical VRPTW formulation. Whereas the VRPTW focuses on a single period, the IRPTW considers a multi-period time horizon, typically measured in terms of days. The IRPTW can be defined on a complete directed graph $G = (N, A)$ where $N = \{0, n + 1\} \cup \{1, \dots, n\}$ is the set of nodes and $A = \{(i, j): i, j \in N, i \neq j\}$ is the set of arcs. Arcs $1, \dots, n$ correspond to the customers, whereas 0 and $n + 1$ represent the single depot (origin-depot and destination-depot). The set of arcs represents connections between the depot and the customers and among customers. No arc terminates in node 0, and no arc originates from node $n + 1$. The proposed model deals with the repeated distribution of a single product from a single supplier to a set of geographically dispersed customers $C = \{1, \dots, n\}$ over a given time horizon of length H. The set of time horizons is denoted by $T = \{1, \dots, H\}$. Each customer $i \in C$ faces a different demand d_i^t per time period $t \in T$. It is assumed that the depot has a sufficient supply of items that can cover all customers' demands throughout the planning horizon. To each arc $(i, j) \in A$, where $i \neq j$, a travel cost c_{ij} and a travel time t_{ij} are associated. The cost and travel time matrices satisfy the triangle inequality. Nodes are associated with points of the plane having the given coordinates $(x_i, y_i) \forall i \in N$, and the travel cost c_{ij} for each arc $(i, j) \in A$ is defined as the Euclidean distance between the two nodes $i, j \in N$.

A fleet of m homogenous vehicles, with capacity Q, is available for the distribution of the product. The fleet of vehicles is denoted by $K = \{1, \dots, m\}$. Each customer $i \in C$ is associated with a time interval $[e_i, l_i]$, called a time window and a service time s_i, where $e_i \leq l_i \; \forall i \in C$. The service of each customer must start within the associated time window, and the vehicle must stop at the customer location for s_i time instants, where $0 \leq s_i \leq l_i - e_i \; \forall i \in C$. Moreover, in case of early arrival at the location of customer $i \in C$, the vehicle generally is allowed to wait until time instant e_i, i.e., until the service may start. Therefore, a vehicle must arrive at the customer $i \in C$ before l_i. It can arrive before e_i but the customer $i \in C$ will not be

serviced before. The depot has also time windows $[e_0, l_0]$ and $[e_{n+1}, l_{n+1}]$ where $e_0 = e_{n+1}$ and $l_0 = l_{n+1}$. The time windows associated with the depot represent the earliest possible departure from the depot as well as the latest possible return time at the depot, respectively. As a result, vehicles may not leave the depot before e_0 and must be back before or at time l_{n+1}. In addition, $s_0 = s_{n+1} = 0$. Each customer maintains his own inventory up to capacity U_i $\forall i \in C$. At the beginning of the planning horizon each customer $i \in C$ has an initial inventory level of $I_i^0 = U_i$ of product. Furthermore, the formulation uses the following decision variables:

- w_{ik}^t: the amount of delivery to customer $i \in C$ in period $t \in T$ by vehicle $k \in K$.
- x_{ijk}^t: a binary variable that is equal to 1 if vehicle $k \in K$ drives from node i to node j $\forall$ (i, j) $\in A$ where $i \neq j, j \neq n + 1, j \neq 0$, and 0 otherwise.
- a_{ik}^t: the time vehicle $k \in K$ starts to service customer $i \in C$.
- y_{ik}^t: a binary variable that is equal to 1 if customer $i \in C$ is visited by vehicle $k \in K$ in period $t \in T$, and 0 otherwise.
- z_k^t: a binary variable that is equal to 1 if vehicle $k \in K$ is used in period $t \in T$, and 0 otherwise.
- I_i^t: a nonnegative variable indicating the inventory level at customer $i \in C$ at the end of period $t \in T$.

Moreover, stock-outs are not allowed at the customers, while the quantities delivered by each vehicle in each route cannot exceed the vehicle capacity. As far as the replenishment policy is concerned, an Order-up-to Level (OL) policy is considered, in which any customer has defined a maximum inventory level and every time a customer is served, the delivered quantity is such that the maximum inventory level at the customer is reached. After defining the necessary parameters and decision variables, the IRPTW can be formulated as shown below:

$$\min \sum_{i \in N} \sum_{j \in N} c_{ij} \sum_{k \in K} \sum_{t \in T} x_{ijk}^t \tag{1}$$

Subject to:

$$I_i^0 = U_i, \forall i \in C \tag{2}$$

$$I_i^{t-1} - I_i^t + \sum_{k \in K} w_{ik}^t = d_i^t, \forall i \in C, \forall t \in T \tag{3}$$

$$I_i^t \leq U_i, \forall i \in C, \forall t \in T \tag{4}$$

$$\sum_{i \in C} w_{ik}^t \leq Q z_k^t, \forall k \in K, \forall t \in T \tag{5}$$

$$\sum_{k \in K} y_{ik}^t \leq 1, \forall i \in C, \forall t \in T \tag{6}$$

$$\sum_{k \in K} y_{0k}^t = m, \forall t \in T \tag{7}$$

$$\sum_{k \in K} y_{n+1,k}^t = m, \forall t \in T \tag{8}$$

$$\sum_{j \in N} x_{jik}^t = y_{ik}^t, \forall i \in N \setminus \{0\}, \forall k \in K, \forall t \in T \tag{9}$$

$$\sum_{j \in N} x_{ijk}^t = y_{ik}^t, \forall i \in N \setminus \{n + 1\}, \forall k \in K, \forall t \in T \tag{10}$$

$$a_{ik}^t + s_i + t_{ij} \leq a_{jk}^t + M\left(1 - x_{ijk}^t\right), \forall i, j \in N, \forall k \in K, \forall t \in T \tag{11}$$

$$a_{ik}^t \geq e_i y_{ik}^t, \forall i \in N, \forall k \in K, \forall t \in T \tag{12}$$

$$a_{ik}^t \leq l_i y_{ik}^t, \forall i \in N, \forall k \in K, \forall t \in T \tag{13}$$

$$x_{ijk}^t \in \{0,1\}, \forall i, j \in N, \forall k \in K, \forall t \in T \tag{14}$$

$$y_{ik}^t \in \{0,1\}, \forall i \in N, \forall k \in K, \forall t \in T \tag{15}$$

$$I_i^t \geq 0, \forall i \in C, \forall t \in T \tag{16}$$

$$w_{ik}^t \geq 0, \forall i \in C, \forall k \in K, \forall t \in T \tag{17}$$

$$a_{ik}^t \geq 0, \forall i \in C, \forall k \in K, \forall t \in T \tag{18}$$

$$z_k^t \leq \sum_{i \in C} y_{ik}^t \quad \forall k \in K, \forall t \in T \tag{19}$$

$$z_k^t n \geq \sum_{i \in C} y_{ik}^t \quad \forall k \in K, \forall t \in T \tag{20}$$

The total cost includes only the transportation costs as depicted in the objective function (1). Constraints (2) indicate that each customer $i \in C$ has an initial inventory level equal to his maximum inventory level. Constraints (3) are the inventory balance equations for the customers. Constraints (4) limit the total amount of inventory to $U_i, \forall i \in C$. Constraints (5), (19) and (20) ensure that the vehicle capacities are not exceeded on any day $t \in T$ during the planning horizon. Constraints (6)-(10) impose that each customer is visited exactly once, m vehicles leave the depot, and the same vehicle enters and leaves a given customer. Constraints (11) ensure feasibility in terms of the time necessary when traveling from node i to node $j \forall i, j \in N$. In addition, ensure simultaneously the elimination of subtours where M is a large constant. Constraints (12) and (13) impose that service may only start within the given interval $[e_i, l_i] \forall i \in N$. Constraints (14)-(18) are the domain constraints.

3. Solution Approach for the IRPTW

Due to the NP-hard nature of the IRPTW, a two-phase solution algorithm based on (a) a simple simulation and (b) the Variable Neighborhood Search Algorithm (VNS) is proposed to handle the problem. The first phase (Phase I) is related to the planning phase of the IRPTW, in which delivery times and quantities are determined by implementing the well-known inventory policy (s,S) for inventory management using a simple simulation. In the second phase (Phase II), the VNS is applied to combine the customers into the vehicle routes by solving a VRPTW for a specific time period during the planning horizon.

3.1. Phase I: Simple Simulation (planning phase)

In particular, (s,S) inventory policy reflects the OU policy, where s and S correspond to a minimum and a maximum inventory level, respectively. An order for S-s units is placed immediately when the inventory level is reduced to s. Since stock-outs are not allowed, inventory policy (s_i, S_i) is applied to each customer $i \in C$ setting $s_i = 0 \forall i \in C$. In addition, each customer has an initial inventory level equal to his maximum inventory capacity $U_i \forall i \in C$. At the end of the planning horizon, each customer should have an inventory level equal to his initial inventory level.

Since the demands are fully available to the supplier at the beginning of the planning horizon, by applying an (s,S) inventory policy to each customer, Phase I of the algorithm enables the supplier to run an inventory simulation to determine delivery times and quantities, so that stock-outs are avoided. A sample problem of

a distribution system that comprises a single supplier and three customers can be considered to explain the inventory simulation algorithm (Fig. 1). In addition, Fig. 2 illustrates for each customer the relative inventory levels during the planning horizon and the delivery times.
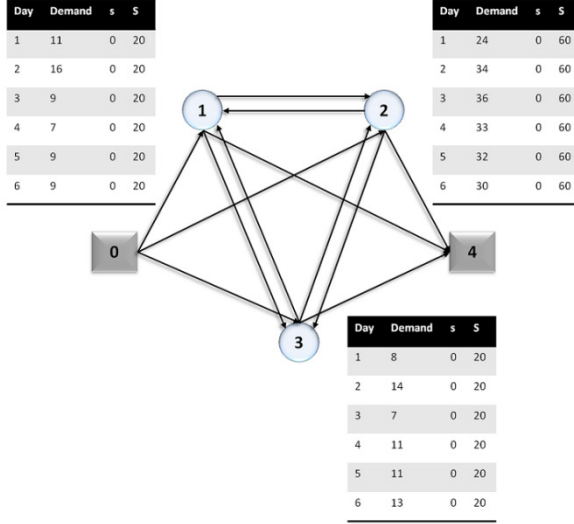

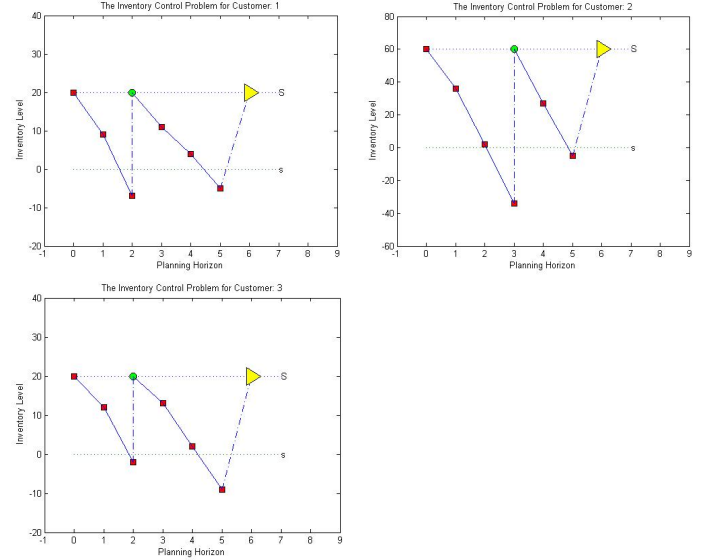**Fig. 1.** Illustrative Example   **Fig. 2.** Inventory Simulation

Below, an algorithm (Algorithm 1) is presented that applies the (s,S) policy to customers. Initially, based on a specific test problem (IRPTWdata), the number of customers (NC) as well as the length of the planning horizon (H) are defined. Then, for each customer i, his (s_i, S_i) inventory policy and demands during the planning horizon (d) are taken into account to determine the delivery quantities and times (deliveries) as well as the inventory levels (inventories). It is worth noting that the time starts from zero, where customer demand is equal to zero and an initial inventory level exists for each customer. To define the delivery quantities the (s,S) policy is applied to each customer. Analytically, for each time period of the planning horizon, if the inventory level (IL) is less than s_i, a delivered quantity (OQ) is defined such that the maximum inventory level at the customer is reached. To define the inventory levels, the inventory balance equation is applied. Namely, the amount of inventory in the next time period must be equal to the current inventory plus the amount of delivered quantity minus the demand in the next time period.

Algorithm 1. *Simple simulation (Phase I)*

Inputs: $IRPTWdata$
$NC \leftarrow getNoCustomers(IRPTWdata), H \leftarrow getPlanningHorizon(IRPTWdata),$
$Inventories \leftarrow [], Deliveries \leftarrow []$
for $i = 1:NC$ **do**
    $s_i \leftarrow gets_i(IRPTWdata), S_i \leftarrow getS_i(IRPTWdata), d \leftarrow [0, d_i^1, d_i^2, \dots, d_i^H], IL(1) \leftarrow S_i,$
    $j \leftarrow 1$
    while $j \leq H$ **do**
        $j \leftarrow j + 1$
        if $IL(j - 1) < s_i$ **then**
            $OQ(j - 1) \leftarrow S_i - IL(j - 1)$
        else
            $OQ(j - 1) \leftarrow 0$
        end − **if**
        $IL(j) \leftarrow IL(j - 1) + OQ(j - 1) - d(j)$

$$end - while$$
$$n \leftarrow j, OQ(n) \leftarrow S_i - IL(n), IL(n) \leftarrow S_i, Deliveries \leftarrow [Deliveries; OQ],$$
$$Inventories \leftarrow [Inventories; IL]$$
$$end - for$$
Output: $Deliveries, Inventories$

3.2. Phase II: Variable Neighborhood Search Algorithm (routing phase)

Since the simple simulation focuses only on the planning phase by determining the delivery times and quantities, the vehicle routes should be constructed. The routing phase (Phase II) is related to the usage of a VNS algorithm for solving a VRPTW for each time period of the planning horizon where delivery quantities have been scheduled. The VNS is a single-point search meta-heuristic introduced by Mladenović and Hansen [12]. A set of neighborhood structures N_k where $k = 1, .., n$ are defined. The algorithm successively explores the set of pre-defined neighborhoods to provide a better solution. Each iteration of the algorithm includes three steps: shaking, local search and move. At each iteration, an initial solution is shaked from the current neighborhood N_k. For example, a solution x' is generated randomly in the current neighborhood $N_k(x)$. Path representation is the most natural way of representing the routes of a VRPTW. Since a VRPTW consists of one or more routes, the length of each path is variable. On account of this, a dynamic variable, x, can be used to represent the solution of the VRPTW. For instance, $x = \{x.R_1, x.R_2\}$ where $x.R_1 = [0,2,0]$ is the first route and $x.R_2 = [0,1,3,0]$ is the second route. The zero value in each row vector represents the supplier, while the other numbers represent the customers. A local search procedure is applied to the solution x' to generate the solution x''. The objective of the VNS is to minimize the cost associated with all proposed routes of a specific time period of the planning horizon. If a solution x consists of h routes, the cost function is equal to $f(x) = \sum_{j=1}^{j=h} Cost(x.R_j)$. Therefore, the current solution is replaced by the new local optima x'' if and only if a better solution has been found (i.e., $f(x'') < f(x)$). The same search procedure is thus restarted from the solution x'' in the first neighborhood N_k. If no better solution is found, the algorithm moves to the next neighborhood N_{k+1}, randomly generates a new solution in this neighborhood, and attempts to improve it.

The generation of the initial solution is based on the Push Forward Insertion Heuristic (PFIH) [13]. The method tries to insert the customer between all the arcs in the current route. It selects the arc that has the lowest additional insertion cost. In addition, the feasibility check tests all the constraints related to time windows and vehicle capacity. When the current route is full of feasible insertions, PFIH will start a new route and repeat the procedure until all the customers are routed. As far as the first step of the VNS (shaking) is concerned, the 2-interchange neighborhood operator of Osman [14] as well as the CROSS-exchange neighborhood operator of Taillard et al. [15] are used randomly (rand2interchange and randCrossExchange). Regarding the second step of the VNS (local search), nested neighborhoods are used based on the 2-interchange and CROSS-exchange mechanisms. These mechanisms (twoInterchange and crossExchange) are used systematically (not randomly). In general terms, the 2-interchange mechanism is based on customer interchange between sets of vehicles routes. The 2 means that maximum two customer nodes may be interchanged between routes. The CROSS-exchange mechanism swaps sequences of consecutive customers between two routes. The detail information about PFIH, 2-interchange and CROSS-exchange can be obtained from papers of Solomon [13], Osman [14] and Taillard et al. [15], respectively. Algorithm 2 presents the template of the proposed VNS algorithm.

Algorithm 2. *Variable neighborhood search algorithm (Phase II)*

Input: $deliveries, IRPTW\ data$
$[H, vehicleCapacity] \leftarrow getPlanningHorizonAndVehCapacity(IRPTWdata), i \leftarrow 1$
while $i \leq H$ **do**
$\quad VRPTW \leftarrow createVRPTWproblem(deliveries, i),$
$\quad x \leftarrow PFIH(VRPTW, vehicleCapacity)$
$\quad$ **Repeat**
$\quad\quad k \leftarrow 1$

$while \ k \leq 2 \ \textbf{do}$
  $\textbf{if} \ k = 1 \ \textbf{then},$
  $x^{'} \leftarrow rand2interchange(x, vehicleCapacity), \textbf{end} - \textbf{if}$
  $\textbf{if} \ k = 2 \ \textbf{then},$
  $x^{'} \leftarrow randCrossExchange(x, vehicleCapacity), \textbf{end} - \textbf{if}$
  $l \leftarrow 1, improvement \leftarrow false$
  $while \ l \leq 2 \ \textbf{do}$
    $\textbf{if} \ l = 1 \ \textbf{then},$
    $x^{''} \leftarrow twoInterchange(x^{'}, vehicleCapacity), \textbf{end} - \textbf{if}$
    $\textbf{if} \ l = 2 \ \textbf{then},$
    $x^{''} \leftarrow crossExchange(x^{'}, vehicleCapacity), \textbf{end} - \textbf{if}$
    $\textbf{if} \ f(x^{''}) < f(x^{'}) \ \textbf{then}$
      $x^{'} \leftarrow x^{''}, l \leftarrow 1, improvement \leftarrow true$
    $else$
      $l \leftarrow l + 1$
    $\textbf{end} - \textbf{if}$
  $\textbf{end} - \textbf{while}$
  $\textbf{if} \ improvement = true \ \textbf{then}$
    $x \leftarrow x^{''}, k \leftarrow 1$
  $else$
    $k \leftarrow k + 1$
  $\textbf{end} - \textbf{if}$
$\textbf{end} - \textbf{while}$
$\textbf{Until} \ stoping \ criteria$
$i \leftarrow i + 1$
$\textbf{end} - \textbf{while}$
Output: $best \ found \ solution$

In addition, based on a sample problem of one supplier and twenty-five customers, Fig.3 illustrates the IRPTW solution.
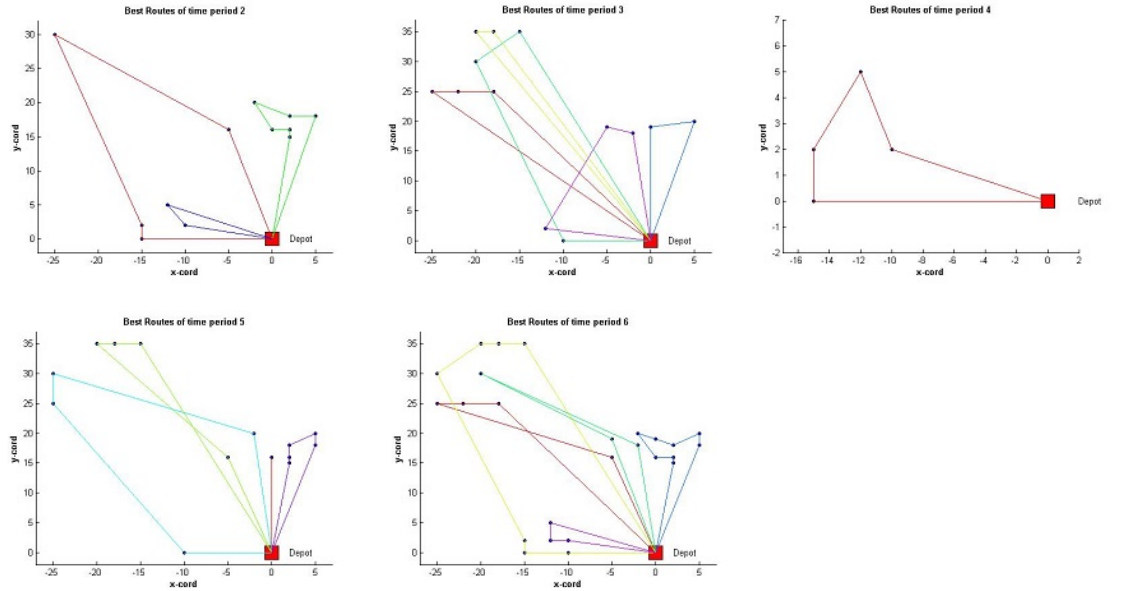

Fig. 3. IRPTW solution for the (1-25) IRPTW model

Table 1 presents the routes that take place in each time period of the planning horizon.

Table 1
Cost information and routes for the IRPTW sample problem

IRPTW Solution		
Routes of Period 1	**Routes of Period 2**	**Routes of Period 3**
	Route 1: 0-24-25-19-10-0	Route 1: 0-13-17-18-0
	Route 2: 0-5-3-7-9-4-1-0	Route 2: 0-16-14-0
	Route 3: 0-23-21-0	Route 3: 0-20-15-12-0
		Route 4: 0-6-2-0
		Route 5: 0-8-11-22-0
VRPTW Cost = 0	VRPTW Cost = 163.4705	VRPTW Cost = 338.1404
Routes of Period 4	**Routes of Period 5**	**Routes of Period 6**
Route 1: 0-24-25-23-21-0	Route 1: 0-7-0	Route 1: 0-13-17-18-10-0
	Route 2: 0-10-16-14-12-0	Route 2: 0-20-24-25-19-16-14-12-0
	Route 3: 0-20-18-19-9-0	Route 3: 0-8-15-11-0
	Route 4: 0-5-3-4-2-1-0	Route 4: 0-5-3-7-9-6-4-2-1-0
		Route 5: 0-23-22-21-0
VRPTW Cost = 35.0462	VRPTW Cost = 247.8036	VRPTW Cost = 330.9369
Total VRPTW Cost		
1115.3975		

4. Computational Experiments and Results

This section presents the computational results of the proposed two-phase solution algorithm. The algorithm was developed in the MATLAB programming language and executed on a DELL personal computer with an Intel® Core™ i3-2120, clocked at 3.30 GHz, a microprocessor with 4 GB of RAM memory under the operating system Microsoft Windows 7 Professional. Since new benchmark instances were designed, the efficiency and the effectiveness of the proposed algorithm cannot be compared to other published IRPTW studies using benchmark instances previously introduced. This is due to the differentiated manner in which the proposed algorithm operates based on the assumptions presented in Section 2. However, this section validates the two-phase solution algorithm and then evaluates its performance by comparing the algorithm's solutions with solutions obtained by solving a VRPTW for each time period of the planning horizon based on the known demands (the planning phase is ignored). The algorithm has been tested on a newly introduced set of 18 IRPTW benchmark instances described in the following. All benchmark instances and their computational results are available at http://www.msl.aueb.gr/files/SimVnsIRPTW.zip.

The new datasets have been developed by generalizing the well-known datasets C101, C201, R101, R201, RC101 and RC201 of Solomon [13]. As a result, these datasets are divided into six classes. The datasets are named in the form of "IRPTW_Z_nX_pY" strings, where "Z" stands for the class related to a specific dataset of Solomon [13], i.e., C101, C201, R101, R201, RC101 and RC201, "X" stands for the number of customers and "Y" stands for the number of time periods. For instance, the problem IRPTW_C101_n25_p6 represents a test problem of the first class (i.e., dataset that was generated by the dataset C101 of Solomon [13]) with 25 customers and a planning horizon of 6 days. Different problem sizes, based on the total number of customers, were designed, in each class. Specifically, each class contains problems with 25, 50 and 100 customers. Nodes coordinates are modified in such a way that the depot is located at the origin (i.e., coordinates (0,0)). The distance matrix is obtained by calculating the Euclidean distances. Time windows related to customers as well as the maximum operation time for each vehicle are kept the same as in the Solomon's datasets [13].

Demand exists for each customer at each time period of the planning horizon. Customer demand at each time period was generated according to the Poisson distribution, Poisson(λ), where λ is the rate parameter. For each customer, the rate parameter is equal to his demand in the single-period VRPTW of Solomon [13]. In addition, for each customer $i \in C$, his maximum inventory capacity is defined as $U_i = 2\lambda_i$. As it usually happens in real life, customers with higher expected demands will have higher inventory capacities. Therefore, for each customer $i \in C$, inventory policy (s_i, S_i) is equal to $(0, U_i)$. An unlimited fleet of

identical vehicles with capacity Q is available for the distribution of the product. The vehicle capacity is kept the same as in the Solomon's datasets. At the beginning of the planning horizon, each customer $i \in C$ has an initial inventory level up to his maximum inventory capacity, i.e., U_i. Finally, the supplier has a sufficient supply of products that can cover customers' demands throughout the planning horizon.

Since the algorithm cannot be compared to other published IRPTW studies, the best solution obtained from the proposed algorithm (IRPTW) is compared to the best solution obtained if the planning phase is ignored (p − VRPTW). In the aftermath of ignoring the planning phase, a VRPTW needs to be solved for each day of the planning horizon according to daily demand. The proposed VNS for the routing phase is then used to solve a daily VRPTW through the planning horizon. To compare the results, the following gap percentage formula is used: $\text{Gap (\%)} = (\text{Sol}_{IRPTW} - \text{Sol}_{p-VRPTW}) \times \frac{1}{\text{Sol}_{p-VRPTW}} \times 100$. The $\text{Sol}_{p-VRPTW}$ corresponds to the solution obtained by solving the daily VRPTWs according to the known daily demands, while the Sol_{IRPTW} determines the solution obtained by applying the proposed two-phase solution algorithm. Since the Sol_{IRPTW} is compared with the $\text{Sol}_{p-VRPTW}$, a positive gap means that the $\text{Sol}_{p-VRPTW}$ is outperformed. The computational results obtained are summarized in Table 2. For the p − VRPTW and IRPTW problems, the respective total vehicle routing cost is presented. In addition, for each IRPTW the computation time (in seconds) needed to obtain a solution is presented, while the last column of the table shows the gap between the two problems reflecting the respective relative error.

Table 2
Experimental results

Instance	p-VRPTW	IRPTW		p-VRPTW – IRPTW
	Vehicle Routing Cost	Vehicle Routing Cost	Computation Time (seconds)	Gap (%)
IRPTW_C101_n25_p6	1150.8817	**1115.3975**	142.5091	-3.0832
IRPTW_C101_n50_p6	2559.8361	**2077.3767**	1.0827e+03	-18.8473
IRPTW_C101_n100_p6	5685.3165	**5067.128**	7.6578e+03	-10.8734
IRPTW_C201_n25_p6	1293.2554	**883.8036**	165.4300	-31.6606
IRPTW_C201_n50_p6	2753.3707	**1614.115**	2.1193e+03	-41.3768
IRPTW_C201_n100_p6	3883.0779	**2812.4137**	3.9252e+04	-27.5726
IRPTW_R101_n25_p6	3804.3657	**1880.0941**	68.9572	-50.5806
IRPTW_R101_n50_p6	6728.73	**3462.5182**	461.9215	-48.5413
IRPTW_R101_n100_p6	10388.3709	**5566.806**	4.6026e+03	-46.4131
IRPTW_R201_n25_p6	2795.4622	**1629.6064**	135.5557	-41.7053
IRPTW_R201_n50_p6	5106.1462	**2626.2472**	1.5377e+03	-48.5669
IRPTW_R201_n100_p6	7567.238	**3837.7514**	1.5402e+04	-49.2846
IRPTW_RC101_n25_p6	3257.2105	**1892.2402**	82.0641	-41.9061
IRPTW_RC101_n50_p6	5969.5287	**3907.7457**	556.1790	-34.5385
IRPTW_RC101_n100_p6	10783.6372	**6145.4136**	5.2140e+03	-43.0117
IRPTW_RC201_n25_p6	2549.0663	**1521.4596**	213.5986	-40.3131
IRPTW_RC201_n50_p6	4548.0065	**2878.2674**	2.2410e+03	-36.7136
IRPTW_RC201_n100_p6	8191.1021	**5053.3035**	1.2644e+04	-38.3074

Based on Table 2, it can be concluded that better solutions are obtained when the planning phase is considered. The ability of each customer to have storage enables a significant decrease in the VRPTW cost, reducing the total number of routes during the planning horizon. As it can be observed, in all cases, the two-phase solution algorithm provides better solutions than the p − VRPTW, with gaps in the interval of −3.0832 percent to −50.5806 percent. The results indicate that if the inventory capacity of each customer is taken into account during the planning phase, better solutions can be obtained, significantly reducing the total transportation cost and designating the importance of integrating supply chain activities. To illustrate in more detail the behavior of the proposed algorithm, more information is presented about the vehicles (number of routes) used in each time period of the planning horizon in Table 3.

Table 3
Number of vehicles used during the planning horizon

Instance	p-VRPTW							IRPTW						
	P1	P2	P3	P4	P5	P6	No. of Routes	P1	P2	P3	P4	P5	P6	No. of Routes
IRPTW_C101_n25_p6	3	3	3	3	3	3	**18**	0	3	5	1	4	5	**18**
IRPTW_C101_n50_p6	7	6	5	6	6	6	36	0	7	6	4	6	9	**32**
IRPTW_C101_n100_p6	11	12	12	12	12	12	71	0	11	17	7	13	20	**68**
IRPTW_C201_n25_p6	2	2	2	2	2	2	12	0	1	2	1	1	2	**7**
IRPTW_C201_n50_p6	4	4	4	3	3	4	22	0	3	4	2	3	5	**17**
IRPTW_C201_n100_p6	5	5	4	5	4	6	29	0	5	4	3	5	7	**24**
IRPTW_R101_n25_p6	9	9	9	9	9	9	54	0	4	6	3	4	10	**27**
IRPTW_R101_n50_p6	14	14	13	14	14	14	83	1	8	7	5	9	13	**43**
IRPTW_R101_n100_p6	23	23	23	23	23	23	138	0	12	13	10	15	24	**74**
IRPTW_R201_n25_p6	4	4	4	4	4	4	24	0	3	2	2	4	4	**15**
IRPTW_R201_n50_p6	6	6	6	6	6	6	36	0	3	5	3	4	6	**21**
IRPTW_R201_n100_p6	9	9	8	10	9	9	54	1	6	6	3	4	9	**29**
IRPTW_RC101_n25_p6	6	6	6	6	6	6	36	0	3	6	1	6	6	**22**
IRPTW_RC101_n50_p6	10	10	10	10	10	10	60	0	7	10	5	8	12	**42**
IRPTW_RC101_n100_p6	20	18	19	20	17	19	113	0	11	16	5	14	20	**66**
IRPTW_RC201_n25_p6	4	4	4	4	4	4	24	0	3	2	2	2	4	**13**
IRPTW_RC201_n50_p6	6	6	5	5	5	5	32	0	4	2	4	4	5	**19**
IRPTW_RC201_n100_p6	10	10	11	10	11	10	62	0	7	7	4	6	9	**33**

Due to the fact that each customer has an initial inventory level equal to his maximum inventory capacity, in most cases no routes occur in period 1. However, for test problems "IRPTW_R101_n50_p6" and "IRPTW_R201_n100_p6" a single route takes place to satisfy the daily demand of specific customers for whom their daily demands are greater than their maximum inventory capacity. Since stock-outs are not allowed, a route takes place to satisfy their demands. In addition, the number of routes is increased at the end of the planning horizon since the (s,S) inventory policy is applied for each customer. According to this policy, for each customer, the inventory level at the end of the planning horizon should be equal to the initial inventory level. On the other hand, in the context of the p-VRPTW, the number of vehicles is nearly the same, as a specific VRPTW problem should be solved on a daily basis.

5. Conclusions and Future Work

In this paper, a two-phase solution algorithm was introduced to handle the IRPTW, which has not been excessively researched in the literature. The paper gives more emphasis to how a simple simulation can be used in hybrid synthesis with a VNS (a single-point search meta-heuristic) for the solution of the IRPTW. Particularly, the simple simulation is related to the planning phase of the IRPTW to determine the delivery times and quantities, while the VNS is associated with the routing phase to determine the routes. The algorithm has been tested on a newly introduced set of 18 IRPTW benchmark instances by comparing the algorithm's solutions with the solutions obtained by solving a VRPTW for each time period of the planning horizon based on known demand (the planning phase is ignored). The computational results show that the proposed algorithm is outperformed simultaneously verifying the benefits obtained by the integration of the inventory and the vehicle routing decisions. Due to the myopic nature of the proposed algorithm, it is worth noting that the two-phase solution algorithm should be even further improved. To begin with, both simulation and VNS should be dealt with in an iterative way to define a re-optimization phase. In this case, (s,S) inventory policy can be initialized randomly and recalculated at each iteration of the solution algorithm. This can be obtained by applying a Discrete Event Monte Carlo Simulation for the planning phase of the problem. In terms of future research, the goals are (a) to extend and improve the proposed algorithm, (b) to explore the algorithm behavior in other problems (instances), (c) to take into account inventory holding costs of customers in the objective function and (d) to focus on the development of other meta-heuristic approaches for the solution of the IRPTW.

References

[1] Liu, S. C., & Lee, W. T. (2011). A Heuristic method for the inventory routing problem with time windows. Expert Systems with Applications 38(10), 13223-13231.

[2] Zeng, W., & Zhao, Q. (2010). Study of stochastic demand inventory routing problem with soft time windows based on MDP. In Z. Zeng, & J. Wang (Eds.), Advances in neural network research and applications (pp. 193-200). Shanghai: Springer.

[3] Zhang, C., Nemhauser, G., Sokol, J., Cheon, M. S., & Keha, A. (2013). Flexible solutions to maritime inventory routing problems with delivery time windows. Georgia Institute of Technology, Technical Report.

[4] Li, Z., Jiang, C., & Jiang, L. (2015). An inventory routing problem with soft time windows. 12th International Symposium on Operations Research and its Application in Engineering, Technology and Management, Luoyang, China.

[5] Zhang, C., Nemhauser, G., Sokol, J., Cheon, M. S., & Papageorgiou, D. (2015). Robust inventory routing with flexible time window allocation. Georgia Institute of Technology, Technical Report.

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

[7] Coelho, L. C., Cordeau, J. F., & Laporte, G. (2013). Thirty years of inventory routing. Transportation Science 48(1), 1-19.

[8] Ballou, R. H. (1989). Heuristics: rules of thumb for logistics decision making. Journal of Business Logistics 10(1), 122-132.

[9] Griffis, S. E., Bell, J. E., & Closs, D. J. (2012). Metaheuristics in logistics and supply chain management. Journal of Business Logistics 33(2), 90-106.

[10] Maniezzo, V., Stützle, T., & Voβ, S. (2009). Matheuristics: hybridizing metaheuristics and mathematical programming. New York: Springer.

[11] Bertazzi, L., & Speranza, M. G. (2012). Inventory routing problems: an introduction. EURO Journal on Transportation and Logistics 1(4), 307-326.

[12] Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. Computers & Operations Research 24(11): 1097-1100.

[13] Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations Research 35(2), 254-265.

[14] Osman, I. H. (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problems. Annals of Operations Research 41(4), 421-451.

[15] Taillard, E., Badeau, P., Gendreau, M., Guertin, F., & Potvin, J. Y. (1997). A tabu search heuristic for the vehicle routing problem with time windows. Transportation Science 31(2), 170-186.