



Production, Manufacturing and Logistics

## Two-stage vehicle routing problem with arc time windows: A mixed integer programming formulation and a heuristic approach

Cihan Çetinkaya<sup>a</sup>, Ismail Karaoglan<sup>b,\*</sup>, Hadi Gökçen<sup>c</sup><sup>a</sup> Logistics Command of Gendarmerie, Guvercinlik, Ankara, Turkey<sup>b</sup> Department of Industrial Engineering, Faculty of Engineering and Architecture, Selcuk University, 42075 Selcuklu, Konya, Turkey<sup>c</sup> Department of Industrial Engineering, Faculty of Engineering, Gazi University, 06570 Maltepe, Ankara, Turkey

## ARTICLE INFO

## Article history:

Received 26 August 2011

Accepted 2 May 2013

Available online 11 May 2013

## Keywords:

Logistics

Vehicle routing problem with arc time windows

Memetic algorithm

Mathematical formulation

## ABSTRACT

In this paper, we introduce a new variant of the Vehicle Routing Problem (VRP), namely the Two-Stage Vehicle Routing Problem with Arc Time Windows (TS\_VRP\_ATWs) which generally emerges from both military and civilian transportation. The TS\_VRP\_ATW is defined as finding the vehicle routes in such a way that each arc of the routes is available only during a predefined time interval with the objective of overall cost minimization. We propose a Mixed Integer Programming (MIP) formulation and a heuristic approach based on Memetic Algorithm (MA) to solve the TS\_VRP\_ATW. The qualities of both solution approaches are measured by using the test problems in the literature. Experimental results show that the proposed MIP formulation provides the optimal solutions for the test problems with 25 and 50 nodes, and some test problems with 100 nodes. Results also show that the proposed MA is promising quality solutions in a short computation time.

© 2013 Elsevier B.V. All rights reserved.

### 1. Introduction

The design of distribution systems is one of the key components of logistics systems as it offers a great potential in order to reduce costs and to improve service quality. Therefore, huge number of studies have been devoted to find good solutions for these systems. In this research area, one of the main topics is the vehicle routing problem (VRP) which involves the design of a set of minimum cost routes serving a set of customers with known demands. In classical VRPs, each customer is served exactly once, each vehicle originates and terminates at the main depot, and all customers must be assigned to only one vehicle such that the vehicle capacities are not exceeded. Following the introduction of the VRP by Dantzig and Ramser (1959), several mathematical models and exact solution procedures developed for small and medium-size VRPs in the literature (Achuthan et al., 2003; Araque et al., 1994; Augerat et al., 1998; Baldacci et al., 2012; Lysgaard et al., 2004; Ralphs, 2003). It is commonly known that VRP belongs to the class of NP-hard problems. Because of its intractability manner, a great deal of heuristic/meta-heuristic approaches have been successfully implemented for the problem. We refer the readers to the review of (Toth and Vigo, 2002) for VRPs and their applications, (Eksioglu et al., 2009) for taxonomic review of VRPs, (Funke et al., 2005) for

local search algorithms, and more recently (Doerner and Schmid, 2010) for meta-heuristic algorithms.

The vehicle routing problem with time windows (VRP\_TW) is a natural extension of the VRP, motivated by the availability of nodes for a specific time windows. In general form of the VRP\_TW, each node must be visited in a predetermined time interval. However, in practice, time window restrictions for node routing problems (i.e. VRPs) can also be appeared on arcs in addition to nodes. Two of them can be given as follows:

- In military logistic system of Turkey, some roads can only be used in predetermined time windows. For instance, especially in eastern cities of Turkey, the roads between city centers and districts generally pass through highlands which are potential ambush locations. Therefore, these roads can only be used in daylight to avoid terrorist attack.
- In addition to the military applications, time window restriction can also be appeared in civilian transportation systems. For example, the trucks cannot enter the city center in rush hours (i.e. between 7–10 am and between 4–7 pm), because these hours includes the beginning and ending time of shifts and the traffic density reaches at the extreme level. Therefore, the entrance of trucks to the city center in these hours is forbidden by local government to facilitate the traffic flow.

Although, there exist many studies in the node routing literature considering the node time windows, to the best of

\* Corresponding author. Tel.: +90 332 223 3751.

E-mail addresses: [cihancetinkaya1903@hotmail.com](mailto:cihancetinkaya1903@hotmail.com) (C. Çetinkaya), [ikaraoglan@selcuk.edu.tr](mailto:ikaraoglan@selcuk.edu.tr) (I. Karaoglan), [hgoekcen@gazi.edu.tr](mailto:hgoekcen@gazi.edu.tr) (H. Gökçen).

our knowledge, there is no previous study considering the arc time windows. Because of the increasing importance of these restrictions for the military and civilian transportation, this paper considers the Vehicle Routing Problem with Arc Time Windows (VRP\_ATWs). Two-stage version of VRP\_ATW (namely TS\_VRP\_ATW) is discussed as well, which is the distribution network structure of the military logistic system in Turkey. In this problem, the network is divided into three layers (facility, depots and customers) and routing operations encountered in both stages each consisting successive layers (i.e. between facility-depots, and depots-customers). We propose Mixed Integer Programming (MIP) formulations for both VRP\_ATW and TS\_VRP\_ATW. We also develop a meta-heuristic approach called Memetic Algorithm (MA) to solve the large-size problem instances. In order to evaluate the performance of the proposed MIP formulation and MA, we carry out experimental study by using various size test problems adapted from the literature. Experimental results show that the proposed MIP formulation gives the optimal solutions for the test problems with 25 and 50 nodes, and some test problems with 100 nodes. The proposed MA also achieves good quality solutions especially for the large-size test problems. The average percentage gap values of the proposed MIP formulation and MA are 1.05 and 2.26, while the average solution times are 984.81 and 11.20 s, respectively.

The main contribution of this paper is threefold. First, we define the VRP\_ATW and emphasize the relationship with other node routing problems. Second, we propose a MIP formulation for VRP\_ATW, and extend this formulation to TS\_VRP\_ATW. Although, there are several papers considering complex routing operations among stages (i.e. (Ambrosino and Graziascutella, 2005; Ambrosino et al., 2009)), to the best of our knowledge, this is the first paper dealing with multi stage VRP under time window restrictions. Finally, we propose a meta-heuristic approach (MA) to solve the large-size problem instances.

The rest of this paper is organized as follows. Related literature for the problem given in Section 2. The definition of the VRP\_ATW, the mathematical formulation and the relationship with other routing problems are shown in Section 3. The definition and the mathematical formulation for the TS\_VRP\_ATW are given in Section 4. The detailed description of the proposed algorithm is given in Section 5. Section 6 reports computational results, and conclusion and future directions are discussed in the last section.

## 2. Literature review

Hadjar and Soumis (2009) have studied the closely related problem, namely multiple depot vehicle scheduling problem with time windows, and proposed a branch-and-price algorithm for the problem. In this paper, the time windows have been transferred from nodes to the arcs for reducing the dimension of huge number of variables, and thus speed up the solution procedure. However, such a problem involves node time windows rather than arc time windows under different structure.

The concept of “time window restrictions on arcs” has been well documented in the capacitated arc routing problem (CARP) literature. In these problems, the main objective is to find a least cost traversal of all required edges in the network in such a way that service must begin and end at the same node (i.e. depot node), and the service must start and finish in a predetermined time interval for all required edges. These time window restrictions may be encountered in areas such as urban waste collection, postal deliveries, treatment of roads to remove snow or to prevent ice forming, or taking pictures of street and roads for digital maps. CARP with time windows (CARP\_TW) have been introduced by

Mullaseril (1997). In this paper, a heuristic algorithm and a transformation approach into a VRP\_TW have been developed. Wøhlk (2005) has also considered the CARP\_TW and presented two mathematical models, several heuristics, a column generation technique to obtain the lower bounds and a dynamic programming approach for the problem. Reghioui et al. (2007) have proposed a hybrid heuristic technique combining greedy randomized adaptive search procedure (GRASP) and path relinking. The authors have shown that proposed heuristic can compete with the best existing heuristic in the literature. Vansteenwegen et al. (2010) have considered the mobile mapping van problem equivalent to CARP with time soft windows and proposed a two-phase heuristic search algorithm for the problem. The authors have proven the strength of the algorithm on the test problems derived from the literature, and several real-life examples. For reference, Corbeñ and Prins (2010) can be reviewed.

## 3. Vehicle routing problem with arc time windows

In this section, we first define the VRP\_ATW and give the relationship with other related problems. We then present a Mixed Integer Programming (MIP) formulation for the problem.

### 3.1. Problem definition

The VRP\_ATW can be defined as follows: Let  $G = (N, A)$  be a complete directed graph where  $N = \{0\} \cup N_C$  is a set of nodes in which “0” and  $N_C$  represent the depot and customers, respectively, and  $A = \{(i, j) | i \neq j, \forall i, j \in N\}$  is the set of arcs. A fleet of homogeneous vehicles with known capacity is available at the depot. Each customer has a nonnegative demand. Each arc has a nonnegative cost, a transportation time, and a time window which identifies the availability of arc. The transition for all arcs must begin within its time window, if used. The problem is to determine the vehicle routes with minimum total cost under following constraints: (i) each vehicle is used at most one route, (ii) each customer is served exactly once, (iii) each route begins and ends at the depot, (iv) the total vehicle load at any point of the route does not exceed the vehicle capacity, and (v) the transition for each arc on any route begins within its time window.

### 3.2. Mixed integer programming formulation for the VRP\_ATW

We next introduce sets, notations and decision variables then proceed by representing the MIP formulation (referred as F1).

#### 3.2.1. Sets and notations

---

$N_C$ :	set of all customers
$N$ :	set of all nodes ( $N = \{0\} \cup N_C$ )
$Q$ :	vehicle capacity
$d_i$ :	demand of customer $i \in N_C$ with $d_i \leq Q$
$c_{ij}$ :	cost of an arc $(i, j) \in A$
$t_{ij}$ :	transportation time of an arc $(i, j) \in A$
$a_{ij}$ :	earliest available time of an arc $(i, j) \in A$
$b_{ij}$ :	latest available time of an arc $(i, j) \in A$

---

#### 3.2.2. Binary decision variables

$$x_{ij} = \begin{cases} 1 & \text{if a vehicle travels directly from node } i \text{ to node } j \text{ } (\forall i, j \in N) \\ 0 & \text{otherwise} \end{cases}$$

### 3.2.3. Additional decision variables

---

$U_i$ ; Load on a vehicle just before having serviced to customer  $i$   
( $\forall i \in N_C$ )  
 $S_i$ ; Departure time of a vehicle from node  $i$  ( $\forall i \in N_C$ )

---

### 3.2.4. Model (F1)

$$\min \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ij} \quad (1)$$

Subject to

$$\sum_{i \in N} x_{ij} = 1 \quad \forall j \in N_C \quad (2)$$

$$\sum_{j \in N} x_{ji} - \sum_{j \in N} x_{ij} = 0 \quad \forall i \in N \quad (3)$$

$$U_j - U_i + Qx_{ij} + (Q - d_i - d_j)x_{ji} \leq Q - d_i \quad \forall i, j \in N_C; \quad i \neq j \quad (4)$$

$$U_i \geq d_i + \sum_{j \in N_0, j \neq i} d_j x_{ij} \quad \forall i \in N_C \quad (5)$$

$$U_i \leq Q - (Q - d_i)x_{i0} \quad \forall i \in N_C \quad (6)$$

$$S_i - S_j + M1_{ij}x_{ij} \leq M1_{ij} - t_{ij} \quad \forall i, j \in N_C; \quad i \neq j \quad (7)$$

$$S_j \geq (a_{0j} + t_{0j})x_{0j} \quad \forall j \in N_C \quad (8)$$

$$S_i \geq a_{ij}x_{ij} \quad \forall i \in N_C; \quad \forall j \in N; \quad i \neq j \quad (9)$$

$$S_i \leq b_{ij} + M2_{ij}(1 - x_{ij}) \quad \forall i \in N_C; \quad \forall j \in N; \quad i \neq j \quad (10)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in N \quad (11)$$

$$U_j, S_j \geq 0 \quad \forall j \in N_C \quad (12)$$

where  $M1_{ij}$  and  $M2_{ij}$  are sufficiently large numbers satisfying constraint sets (7) and (10) when  $x_{ij} = 0$ , respectively. Furthermore, we set  $x_{ij} = 0$  whenever  $(d_i + d_j > Q)$  or  $(a_{ij} + t_{ij} > \max_{(k \in N | k \neq i)}(b_{jk}))$ .

In (F1), objective function (1) minimizes the total transportation cost. Constraint sets (2) and (3) are known as degree constraints. While constraint set (2) ensures that each customer must be visited exactly once, constraint set (3) guarantees that the number of entering and leaving arcs to each node are equal. Constraint sets (4)–(6) are well known Miller–Tucker–Zemlin (MTZ) constraints eliminating the subtours, subject to the capacity restrictions. Constraint sets (7)–(10) define the time windows restrictions. While constraint set (7) determines the departure time of each customer depending on the previous node on a route, constraint sets (8) and (9) guarantee that the departure time of each node must be greater than the earliest available time of used arc. Similarly, constraint set (10) ensures that the departure time of each node must be smaller than the latest available time of used arc. And finally, constraint sets (11) and (12) are integrality and non-negativity constraints, respectively.

In most combinatorial optimization problems, the efficiency of any MIP formulation depends on the strength of the linear programming (LP) relaxation of a given formulation. There are several ways to strengthen the LP relaxations of the formulations. One of them is to lift the current constraints. Please note that constraint sets (4)–(6) are lifted version of the classical MTZ constraints. These constraints have been proposed by Miller et al. (1960) for the Traveling Salesman Problem (TSP). Kulkarni and Bhawe (1985) have adapted them to the VRP and Desrochers and Laporte (1991) have lifted them and introduced new bounding constraints on additional variables. Finally, Kara et al. (2004) have corrected the lifted version of these constraints. Meanwhile, exact meanings to additional variables have been given by Kara (2008).

Another way to strengthen the LP relaxations is improving the constraints including big  $M$  constants (i.e.  $M1_{ij}$  and  $M2_{ij}$  in constraint sets (7) and (10)). These numbers should be as small

as possible to obtain good lower bounds, which shorten the optimization process. Therefore, we set  $M1_{ij} = \bar{S}_i - S_j + t_{ij}$  and  $M2_{ij} = \bar{S}_i - b_{ij}$  where  $\bar{S}_i = \max_{j \in N} (b_{ij} + t_{ij})$  and  $S_j = \min(a_{0j} + t_{0j})$ .

The last way is to include the valid inequalities, one of the most practical ways to strengthen the LP relaxations of the formulations. We therefore utilize the following valid inequality for (F1).

$$\sum_{j \in N_C} x_{0j} \geq r_{VRP\_ATW}(N_C) \quad (13)$$

where,  $r_{VRP\_ATW}(N_C) = \lceil \sum_{i \in N_C} d_i / Q \rceil$  and  $\lceil \bullet \rceil$  is the smallest integer bigger than  $\bullet$ . This constraint bounds below the number of routes originating from the depot.

In this paper, we used all of these rules to strengthen the LP relaxation of our formulation.

### 3.3. Relationships between VRP\_ATW and other routing problems

The VRP\_ATW can be considered as the classical VRP when  $a_{ij} = 0$  and  $b_{ij} = \infty$ . Since the VRP belongs to the class of NP-hard problems, the VRP\_ATW is also NP-hard.

Any VRP\_TW instance can easily be converted into a VRP\_ATW by replacing the time window constraints on nodes with time window constraints on arcs, such that  $a_{ij} = e_i$  and  $b_{ij} = \min\{l_i, l_j - t_{ij}\}$  where  $e_i(l_i)$  is the earliest (latest) available time of nodes for the corresponding VRP\_TW instance. This transformation technique has also been used by Hadjar and Soumis (2009) for the multiple depot vehicle scheduling problem with time windows. Therefore, any solution procedure developed for the VRP\_ATW can also be used directly to solve the VRP\_TW after this transformation. Moreover, time window constraints can be appeared on both nodes and arcs. In this situation, the problem can be converted to the equivalent VRP\_ATW by setting  $a'_{ij} = \max\{e_i, a_{ij}\}$  and  $b'_{ij} = \min\{b_{ij}, \min\{l_i, l_j - t_{ij}\}\}$  where  $a'_{ij}(b'_{ij})$  is the earliest (latest) available time of arcs for the transformed VRP\_ATW instance. Please note that, after these transformations any arc satisfying  $a_{ij} > b_{ij}$  can be removed from the problem instance.

However, as far as we search the literature, the reverse relationship (i.e. transformation from VRP\_ATW to VRP\_TW) does not exist, since there could be different time windows of all outgoing arcs for all nodes, and aggregating them to a single time window for origin node is not possible. Instead, we can transform any VRP\_ATW instance to the Generalized Vehicle Routing Problem with Time Windows (GVRP\_TW) which provides a useful modeling framework for a wide variety of applications (Baldacci et al., 2010; Moccia et al., 2011). The detailed definition of the GVRP\_TW and the transformation steps are given in Appendix A.

### 4. Two-stage vehicle routing problem with arc time windows

Our interest in considering two-stage version of VRP\_ATW actually stems from the logistic system of the General Commandership of Gendarmerie of Turkey. In this system, ammunitions (spare parts) stored (produced) in a main depot (factory) which is located in Ankara are firstly dispatched to the provincial gendarmerie units which are located in city centers all around country by using large trucks. Then these parts routed from the provincial gendarmerie units to the local/border units by using small pickup trucks. However, due to the “safe day” implementation which is applied for the distribution to the eastern cities in order to avoid the terrorist attack, the roads are available only during a predetermined time windows. The problem under investigation can be described as Two-Stage VRP\_ATW (TS\_VRP\_ATW) where the network is divided into three layers (i.e. facility, depots and customers) and routing operations is encountered in both stages each consisting successive layers (i.e. between facility-depots, and depots-customers).

TS\_VRP\_ATW can be defined as follows: Let  $G = (N, A)$  be a tripartite complete directed graph where  $N = \{0\} \cup J \cup K$  is a set of nodes in which “0”,  $J$  and  $K$  represent facility, depots and customers, respectively, and  $A = \{(i, j) | i \neq j, \forall i, j \in N\}$  is the set of arcs. Each arc has a nonnegative cost, a transportation time, and a time window which identifies the availability of arc. Each customer is served from a predefined depot. Facility and depots have enough capacity to meet their demands. Two types of vehicle fleet with known capacities are available to serve between facility–depots, and between depots–customers, respectively. Each customer has a non-negative demand. The problem is to determine the vehicle routes with minimum total cost under following constraints: (i) commodities are transported from facility to depots on a route, and then from depots to the customers. The transportation from facility to customers is not permitted, (ii) each vehicle is used at most one route in only one stage, (iii) each customer/depot is served exactly once, (iv) each route begins and ends at the node where they started (i.e. facility or depot), (v) the total vehicle load at any point of the route does not exceed the vehicle capacity, and (vi) the transition for each arc on any route begins within its time window,

The TS\_VRP\_ATW can be considered as the VRP\_ATW when the depots intermediate between facilities and customers are removed. Since the VRP\_ATW belongs to the class of NP-hard problems, the TS\_VRP\_ATW is also NP-hard.

According to these definitions, the sets, notations, decision variables and MIP formulation (referred as F2) are given as follows ( $c_{ij}$ ,  $t_{ij}$ ,  $a_{ij}$  and  $b_{ij}$  are as given in Section 3.2):

#### 4.1. Sets and notations

$J$	Set of all depots
$K$	Set of all customers
$N$	Set of all nodes ( $N = \{0\} \cup J \cup K$ )
$K_j$	Set of customers assigned to depot $j \in J$
$Q_{fd}$	Capacity of a vehicle available to serve between facility and depots
$Q_{dc}$	Capacity of a vehicle available to serve between depots and customers
$d_k$	Demand of customer $k \in K$ with $d_k \leq Q_{dc}$ and $\sum_{k \in K_j} d_k \leq Q_{fd} : \forall j \in J, \forall k \in K_j$

#### 4.2. Binary decision variables

$$x_{ij} = \begin{cases} 1 & \text{if a vehicle travels directly from node } i \text{ to node } j \\ & \text{in the first stage } (\forall i, j \in (0 \cup J)) \\ 0 & \text{otherwise} \end{cases}$$

$$y_{jkl} = \begin{cases} 1 & \text{if a vehicle, originating from depot } j, \text{ travels directly} \\ & \text{from node } k \text{ to node } l \text{ in the second stage } (\forall j \in J; \forall k, l \in (j \cup K_j)) \\ 0 & \text{otherwise} \end{cases}$$

#### 4.3. Additional decision variables

$U_j$	Load on a vehicle just before having serviced to depot $j$ in the first stage ( $\forall j \in J$ )
$V_k$	Load on a vehicle just before having serviced to customer $k$ in the second stage ( $\forall k \in K$ )
$S_k$	Departure time of a vehicle from node $k$ ( $\forall k \in (J \cup K)$ )

#### 4.4. Model (F2)

$$\min \sum_{i \in (0 \cup J)} \sum_{j \in (0 \cup J)} c_{ij} x_{ij} + \sum_{j \in J} \sum_{k \in (j \cup K_j)} \sum_{l \in (j \cup K_j)} c_{kl} y_{jkl} \quad (14)$$

Subject to

$$\sum_{i \in (0 \cup J)} x_{ij} = 1 \quad \forall j \in J \quad (15)$$

$$\sum_{j \in (0 \cup J)} x_{ji} - \sum_{j \in (0 \cup J)} x_{ij} = 0 \quad \forall i \in (0 \cup J) \quad (16)$$

$$U_j - U_i + Q_{fd} x_{ij} + (Q_{fd} - D_i - D_j) x_{ji} \leq Q_{fd} - D_i \quad \forall i, j \in J; i \neq j \quad (17)$$

$$U_i \geq D_i + \sum_{j \in J, j \neq i} D_j x_{ij} \quad \forall i \in J \quad (18)$$

$$U_i \leq Q_{fd} - (Q_{fd} - D_i) x_{i0} \quad \forall i \in J \quad (19)$$

$$S_j - S_m + M1_{jm} x_{jm} \leq M1_{jm} - t_{jm} \quad \forall j, m \in J; j \neq m \quad (20)$$

$$S_j \geq (a_{0j} + t_{0j}) x_{0j} \quad \forall j \in J \quad (21)$$

$$S_j \geq a_{jm} x_{jm} \quad \forall j \in J; \forall m \in (0 \cup J); j \neq m \quad (22)$$

$$S_j \leq b_{jm} + M2_{jm} (1 - x_{jm}) \quad \forall j \in J; \forall m \in (0 \cup J); j \neq m \quad (23)$$

$$\sum_{k \in (j \cup K_j)} y_{jkl} = 1 \quad \forall j \in J, \forall l \in K_j \quad (24)$$

$$\sum_{k \in (j \cup K_j)} y_{jkl} - \sum_{k \in (j \cup K_j)} y_{jlk} = 0 \quad \forall j \in J, \forall l \in (j \cup K_j) \quad (25)$$

$$V_l - V_k + Q_{dc} y_{jkl} + (Q_{dc} - d_k - d_l) y_{jlk} \leq Q_{dc} - d_k \quad \forall j \in J, \forall k, l \in K_j; k \neq l \quad (26)$$

$$V_k \geq d_k + \sum_{l \in K_j, k \neq l} d_l y_{jkl} \quad \forall j \in J, \forall k \in K_j \quad (27)$$

$$V_k \leq Q_{dc} - (Q_{dc} - d_k) y_{jkl} \quad \forall j \in J, \forall k \in K_j \quad (28)$$

$$S_k - S_l + M1_{kl} y_{jkl} \leq M1_{kl} - t_{kl} \quad \forall j \in J, \forall k, l \in K_j; k \neq l \quad (29)$$

$$S_k \geq S_j + (a_{jk} + t_{jk}) y_{jkl} \quad \forall j \in J, \forall k \in K_j \quad (30)$$

$$S_k \geq a_{kl} y_{jkl} \quad \forall j \in J, \forall k \in K_j; \forall l \in (j \cup K_j); k \neq l \quad (31)$$

$$S_k \leq b_{kl} + M2_{kl} (1 - y_{jkl}) \quad \forall j \in J, \forall k \in K_j; \forall l \in (j \cup K_j); k \neq l \quad (32)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in (0 \cup J) \quad (33)$$

$$y_{jkl} \in \{0, 1\} \quad \forall j \in J; \forall k, l \in (j \cup K_j) \quad (34)$$

$$U_j \geq 0 \quad \forall j \in J \quad (35)$$

$$V_k \geq 0 \quad \forall k \in K \quad (36)$$

$$S_j \geq 0 \quad \forall j \in (J \cup K) \quad (37)$$

where,  $M1_{ij}$  and  $M2_{ij}$  are defined as previously, and  $D_j = \sum_{k \in K_j} d_k : \forall j \in J$ . Moreover, we set  $x_{ij} = 0$  whenever  $(D_i + D_j > Q_{fd})$  or  $(a_{ij} + t_{ij} > \max_{k \in (0 \cup J) | k \neq i} (b_{jk}))$ , and  $y_{jkl} = 0$  whenever  $(d_k + d_l > Q_{dc})$  or  $(a_{kl} + t_{kl} > \max_{m \in (0 \cup J) | m \neq k} (b_{lm}))$ .

It is easy to see that (F2) is an extended version of (F1). In (F2), objective function (14) minimizes the total transportation cost. The objective function includes two parts where the former related to the first stage and the latter related to the second stage transportation operations, respectively. Constraint sets (15) and (16) are known as degree constraints. Constraint sets (17)–(19) are MTZ capacity/subtour elimination constraints for the first stage. Constraint sets (20)–(23) define the time windows restrictions. While constraint set (20) determines the departure time of each depot depending on the previous node on a route, constraint sets (22) and (21) guarantee that the departure time of each node must be greater than the earliest available time of used arc. Similarly, constraint set (23) ensures that the departure time of each node must be smaller than the latest available time of used arc. Constraint sets (15)–(23) are used for modeling the routing operation in the first stage. Similarly, constraint set (24)–(32) are used for the second stage. And finally, constraint sets (33) and (34) are integrality constraints, and (35)–(37) are non-negativity constraints.

It is worthy to note that, in the absence of the time window restrictions, the second stage routes are not affected from the arrival time to the depot of the first stage routes (please see constraint set (30)). Under this assumption, the TS\_VRP\_ATW reduces to  $(J + 1)$  independent VRPs where the one is related to the routing operation between factory and depots ( $j \in J$ ), and the

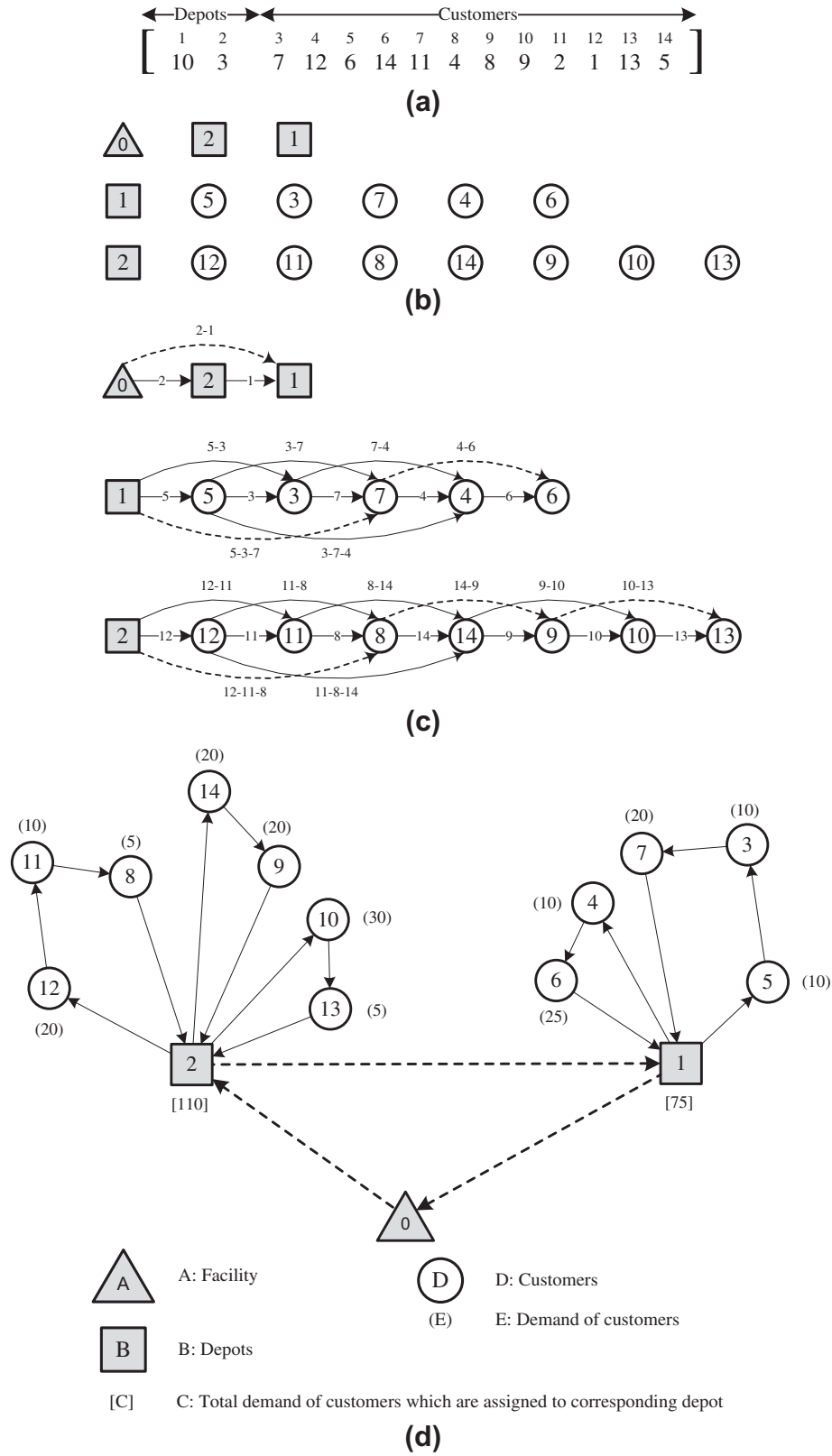


Fig. 1. An illustrative example for chromosome.

others are related to each depot ( $j \in J$ ) and corresponding customers ( $k \in K_j$ ).

As explained in previous section, we utilize the following valid inequalities to strengthen the LP relaxation of model (F2).

$$\sum_{j \in J} x_{0j} \geq r_{TS\_VRP\_ATW}(J) \quad (38)$$

$$\sum_{k \in K_j} y_{ijk} \geq r_{TS\_VRP\_ATW}(K_j) \quad \forall j \in J \quad (39)$$



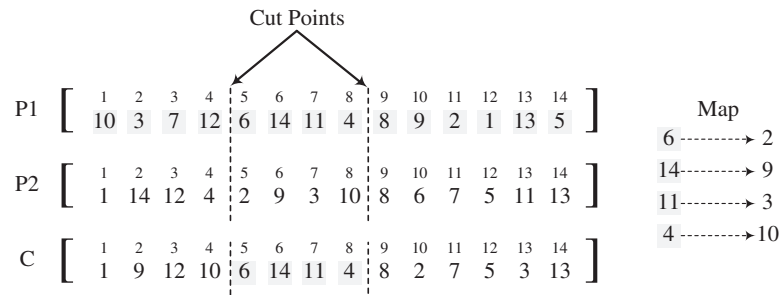


Fig. 2. An illustrative example for PMX operator.

<b>Procedure:</b>	MA for TS_VRP_ATW
<b>Input:</b>	Problem input for TS_VRP_ATW, parameters of MA
<b>Output:</b>	Best Solution
<b>Step 0:</b>	$t \leftarrow 0$ , Initialize the population $P(t)$
<b>Step 1:</b>	Evaluate $P(t)$
<b>Step 2:</b>	Improve $P(t)$ using local search algorithm
<b>Step 3:</b>	Choose $P1$ and $P2$ by using tournament selection mechanism
<b>Step 4:</b>	Obtain $C$ from $P1$ and $P2$ by using $PMX$ operator.
<b>Step 5:</b>	Improve $C$ using local search algorithm
<b>Step 6:</b>	Insert $C$ into the population and update $P(t)$ .
<b>Step 7:</b>	$t \leftarrow t + 1$ .
<b>Step 8:</b>	If the termination criterion is met then go to <b>Step 9</b> else go to <b>Step 3</b>
<b>Step 9:</b>	Report the best solution.

Fig. 3. Steps of the proposed memetic algorithm.

**Table 1**  
Computational results for C-type test problems with tight time windows.

K	J	Problem name	LB	MIP model		Memetic algorithm	
				Gap	CPU	Gap	CPU
20	5	C101_25*	234.79	0.00	0.37	0.00	1.82
		C102_25*	234.77	0.00	0.42	0.00	1.48
		C103_25*	234.76	0.00	0.55	0.00	1.47
		C104_25*	234.74	0.00	0.47	0.00	2.30
		C105_25*	234.79	0.00	0.44	0.00	1.51
		C106_25*	234.79	0.00	0.45	0.00	1.49
		C107_25*	234.78	0.00	0.55	0.00	1.49
		C108_25*	234.78	0.00	0.43	0.00	1.51
		C109_25*	234.77	0.00	0.50	0.00	2.01
		Average	234.77	0.00	0.46	0.00	1.68
40	10	C101_50*	426.41	0.00	10.09	0.00	5.91
		C102_50*	423.26	0.00	3.39	0.00	5.89
		C103_50*	423.22	0.00	2.26	0.00	6.29
		C104_50*	423.19	0.00	2.65	0.00	6.40
		C105_50*	426.40	0.00	11.14	0.00	6.27
		C106_50*	426.40	0.00	17.14	0.00	5.99
		C107_50*	426.32	0.00	10.67	0.00	6.46
		C108_50*	423.85	0.00	4.07	0.00	7.17
		C109_50*	423.25	0.00	2.92	0.00	6.18
		Average	424.70	0.00	7.15	0.04	6.28
90	10	C101_100	871.12	8.40	3600.00	10.70	24.49
		C102_100	869.26	8.50	3600.00	10.70	22.68
		C103_100	865.58	8.70	3600.00	11.00	22.98
		C104_100	873.29	7.70	3600.00	10.00	22.53
		C105_100	860.73	9.70	3600.00	12.00	23.49
		C106_100	881.41	7.00	3600.00	9.30	22.73
		C107_100	864.48	8.80	3600.00	11.20	21.36
		C108_100	882.54	6.60	3600.00	8.90	23.22
		C109_100	871.52	8.00	3600.00	10.20	24.74
		Average	871.10	8.16	3600.00	10.44	23.14
Overall average			510.19	2.72	1202.54	3.48	10.37

**Table 2**

Computational results for R type test problems with tight time windows.

K	J	Problem name	LB	MIP model		Memetic algorithm	
				Gap	CPU	Gap	CPU
20	5	R101_25*	485.57	0.00	0.31	0.00	1.50
		R102_25*	485.57	0.00	0.19	0.00	1.24
		R103_25*	485.56	0.00	0.20	0.00	1.23
		R104_25*	485.56	0.00	0.12	0.00	1.48
		R105_25*	485.57	0.00	0.20	0.00	1.45
		R106_25*	485.57	0.00	0.22	0.00	1.28
		R107_25*	485.56	0.00	0.11	0.00	1.75
		R108_25*	485.56	0.00	0.14	0.00	1.43
		R109_25*	485.57	0.00	0.33	0.00	1.79
		R110_25*	485.56	0.00	0.36	0.00	1.27
		R111_25*	485.56	0.00	0.31	0.00	1.49
		R112_25*	485.56	0.00	0.30	0.00	1.23
Average			485.56	0.00	0.23	0.00	1.43
40	10	R101_50*	830.15	0.00	5.13	0.00	5.30
		R102_50*	825.04	0.00	2.61	0.00	5.29
		R103_50*	825.04	0.00	2.40	0.00	5.02
		R104_50*	825.04	0.00	2.92	0.00	5.26
		R105_50*	825.05	0.00	2.34	0.00	5.34
		R106_50*	825.04	0.00	2.37	0.00	4.94
		R107_50*	825.04	0.00	2.68	0.00	5.33
		R108_50*	825.04	0.00	2.32	0.00	5.50
		R109_50*	825.04	0.00	3.04	0.00	6.16
		R110_50*	825.04	0.00	2.67	0.00	6.58
		R111_50*	825.04	0.00	3.14	0.00	5.63
		R112_50*	825.04	0.00	3.07	0.00	6.74
Average			825.47	0.00	2.89	0.00	5.59
90	10	R101_100	1021.96	1.60	3600.00	6.70	23.20
		R102_100	1021.68	0.70	3600.00	4.30	22.88
		R103_100	1021.09	0.70	3600.00	4.20	24.19
		R104_100	1021.08	0.70	3600.00	4.20	23.52
		R105_100	1021.01	0.90	3600.00	5.30	21.63
		R106_100	1020.60	0.80	3600.00	4.60	26.42
		R107_100	1019.14	0.90	3600.00	4.40	26.76
		R108_100	1021.41	0.70	3600.00	4.20	24.14
		R109_100	1019.73	0.90	3600.00	4.30	27.50
		R110_100	1021.85	0.60	3600.00	4.10	28.26
		R111_100	1021.07	0.70	3600.00	4.30	28.21
		R112_100	1021.11	0.80	3600.00	4.20	25.32
Average			1020.98	0.83	3600.00	4.57	25.17
Overall average			777.34	0.28	1201.04	1.52	10.73

where  $r_{TS\_VRP\_ATW}(J) = \lceil \sum_{j \in J} D_j / Q_{dc} \rceil$ ,  $r_{TS\_VRP\_ATW}(K_j) = \lceil \sum_{k \in K_j} d_k / Q_{dc} \rceil$  and  $\lceil \bullet \rceil$  is the smallest integer bigger than  $\bullet$ . While, constraint (38) bounds below the number of routes originating from the facility, constraint set (39) bounds below the number of routes originating from each depot, respectively.

## 5. Proposed heuristic approach for the TS\_VRP\_ATW

As mentioned above, TS\_VRP\_ATW belongs to the class of NP-hard problems. Therefore, the proposed formulation is not directly applicable to find optimal solutions for the large-size problems. Hence, heuristic solution procedures are necessary to obtain good quality solutions quickly for these problems.

Motivated by the successful applications of Memetic Algorithms (MAs) in solving various vehicle routing problems (Boudia et al., 2007; Labadi et al., 2008; Liu and Jiang, 2012; Nguveu et al., 2010; Prins, 2009), we propose a MA to solve realistic size of the TS\_VRP\_ATW.

MA is an evolutionary search procedure that use local search within a classical genetic algorithm framework. MA offer significant advantages over conventional methods by using simultaneously diversification (i.e. recombination) and intensification

mechanisms (i.e. local search). This section presents the detailed description of components of the proposed MA for TS\_VRP\_ATW.

### 5.1. Representation

The performance of MA depends heavily on the chromosome structure that is used to represent solutions, and recombination operators that are used for this chromosome structure. Therefore, an appropriate chromosome structure should be developed for problem. When the related literature is examined, it is seen that good results were obtained by permutation encoding scheme originally proposed by Prins (2004) for the VRPs. For this reason, we developed an encoding scheme for our problem based on permutation encoding. In our structure, the length of the chromosome is equal to the sum of number of depots and customers (i.e.,  $|J| + |K|$ ). Each gene gives information about the relative sequence of corresponding node with respect to other nodes. The decoding of a chromosome includes two main steps. Firstly, all nodes are grouped into their assigned depot. Please note that, for the first stage, the depots are considered as the customers of the facility. Then, the optimal routes for each group are obtained by using following splitting procedure which is modified version of route partitioning

**Table 3**

Computational results for RC type test problems with tight time windows.

K	J	Problem name	LB	MIP model		Memetic algorithm	
				Gap	CPU	Gap	CPU
20	5	RC101_25*	367.41	0.00	0.31	0.00	2.03
		RC102_25*	367.41	0.00	0.33	0.00	1.59
		RC103_25*	367.40	0.00	0.36	0.00	1.58
		RC104_25*	367.40	0.00	0.20	0.00	1.80
		RC105_25*	367.41	0.00	0.22	0.00	1.68
		RC106_25*	367.41	0.00	0.30	0.00	1.54
		RC107_25*	367.40	0.00	0.27	0.00	1.66
		RC108_25*	367.40	0.00	0.37	0.00	1.94
Average			367.41	0.00	0.30	0.00	1.73
40	10	RC101_50*	661.29	0.00	13.06	0.00	7.80
		RC102_50*	661.29	0.00	12.03	0.00	8.00
		RC103_50*	661.28	0.00	7.02	0.00	7.63
		RC104_50*	661.28	0.00	8.83	0.00	6.80
		RC105_50*	661.29	0.00	15.33	0.00	7.81
		RC106_50*	661.29	0.00	14.32	0.00	6.47
		RC107_50*	661.28	0.00	12.62	0.00	6.58
		RC108_50*	661.28	0.00	10.05	0.00	6.12
Average			661.29	0.00	11.66	0.00	7.15
90	10	RC101_100	1133.52	5.20	3600.00	10.20	25.24
		RC102_100	1130.40	5.80	3600.00	10.50	27.53
		RC103_100	1124.06	6.00	3600.00	11.10	29.46
		RC104_100	1128.76	6.00	3600.00	10.60	28.62
		RC105_100	1130.99	5.80	3600.00	10.40	29.27
		RC106_100	1132.18	5.70	3600.00	10.70	25.93
		RC107_100	1122.93	6.50	3600.00	11.60	28.24
		RC108_100	1131.38	5.40	3600.00	10.80	24.82
Average			1129.28	5.80	3600.00	10.74	27.39
Overall average			719.32	1.93	1203.98	3.58	12.09

**Table 4**

Computational results for C type test problems with wide time windows.

K	J	Problem name	LB	MIP model		Memetic algorithm	
				Gap	CPU	Gap	CPU
20	5	C201_25*	292.17	0.00	0.25	0.00	1.26
		C202_25*	292.09	0.00	0.34	0.00	1.23
		C203_25*	291.99	0.00	0.24	0.00	1.22
		C204_25*	291.91	0.00	0.33	0.00	1.43
		C205_25*	292.16	0.00	0.28	0.00	1.27
		C206_25*	292.15	0.00	0.25	0.00	1.27
		C207_25*	292.11	0.00	0.28	0.00	1.31
		C208_25*	292.13	0.00	0.36	0.00	1.51
Average			292.09	0.00	0.29	0.00	1.31
40	10	C201_50*	511.85	0.00	2.79	0.00	6.66
		C202_50*	511.63	0.00	10.16	0.00	5.53
		C203_50*	511.51	0.00	3.63	0.00	5.56
		C204_50*	511.41	0.00	3.52	0.00	5.53
		C205_50*	511.82	0.00	2.85	0.00	6.62
		C206_50*	511.79	0.00	1.97	0.00	5.62
		C207_50*	511.73	0.00	2.12	0.00	5.34
		C208_50*	511.76	0.00	2.06	0.00	5.66
Average			511.69	0.00	3.64	0.00	5.81
90	10	C201_100	912.35	2.10	3600.00	1.90	23.69
		C202_100	909.05	1.80	3600.00	1.90	25.34
		C203_100	911.26	1.50	3600.00	1.70	22.83
		C204_100	907.82	1.80	3600.00	2.00	22.67
		C205_100	910.95	2.20	3600.00	2.20	23.96
		C206_100	901.53	3.10	3600.00	2.90	23.66
		C207_100	896.25	3.90	3600.00	3.80	22.22
		C208_100	909.62	2.10	3600.00	2.00	23.75
Average			907.35	2.31	3600.00	2.30	23.52
Overall average			570.38	0.77	1201.31	0.77	10.21



**Table 5**

Computational results for R type test problems with wide time windows.

K	J	Problem name	LB	MIP model		Memetic algorithm	
				Gap	CPU	Gap	CPU
20	5	R201_25*	476.67	0.00	0.19	0.00	1.30
		R202_25*	476.66	0.00	0.19	0.00	1.60
		R203_25*	476.61	0.00	0.19	0.00	1.99
		R204_25*	476.60	0.00	0.14	0.00	1.56
		R205_25*	476.66	0.00	0.17	0.00	1.36
		R206_25*	476.65	0.00	0.17	0.00	1.37
		R207_25*	476.61	0.00	0.19	0.00	1.80
		R208_25*	476.60	0.00	0.14	0.00	1.24
		R209_25*	476.64	0.00	0.17	0.00	1.80
		R210_25*	476.65	0.00	0.16	0.00	1.25
		R211_25*	476.64	0.00	0.16	0.00	2.18
Average		476.64	0.00	0.17	0.00	1.59	
40	10	R201_50*	728.82	0.00	1.50	1.40	6.18
		R202_50*	728.78	0.00	0.61	1.50	6.51
		R203_50*	728.76	0.00	0.75	1.50	5.23
		R204_50*	728.76	0.00	1.37	0.00	6.14
		R205_50*	728.80	0.00	1.04	0.00	6.37
		R206_50*	728.77	0.00	1.39	0.00	5.76
		R207_50*	728.76	0.00	1.03	0.00	5.23
		R208_50*	728.75	0.00	1.00	0.00	6.39
		R209_50*	728.79	0.00	1.14	0.00	6.49
		R210_50*	728.79	0.00	1.15	0.00	5.87
		R211_50*	728.78	0.00	1.23	0.00	6.14
Average		728.78	0.00	1.11	0.40	6.03	
90	10	R201_100*	967.03	0.00	715.64	0.60	28.75
		R202_100*	966.29	0.00	301.91	0.00	29.55
		R203_100*	966.26	0.00	177.62	0.10	26.94
		R204_100*	966.21	0.00	241.22	0.10	26.18
		R205_100*	966.35	0.00	273.37	0.50	27.23
		R206_100*	966.25	0.00	202.41	0.10	28.20
		R207_100*	966.23	0.00	182.72	0.10	27.28
		R208_100*	966.19	0.00	170.52	0.10	26.64
		R209_100*	966.28	0.00	365.95	0.00	29.29
		R210_100*	966.30	0.00	180.07	0.10	27.50
		R211_100*	966.28	0.00	330.75	0.10	27.11
Average		966.33	0.00	285.65	0.16	27.70	
Overall average		723.92	0.00	95.64	0.19	11.77	

procedure proposed by Prins (2004) for the VRP. Let  $\Omega_k = \{c_1, c_2, \dots, c_s\}$  be a node set of corresponding group where  $s$  is the number of customers assigned to depot  $k$ , and  $G(N_k, A_k)$  be a directed acyclic auxiliary graph where  $N_k = \{k \cup \Omega_k\}$  is the set of vertices and  $A_k = \{(c_i, c_j) | c_i, c_j \in N_k \text{ and } i < j\}$  is the set of directed arcs. Each arc  $(c_i, c_j) \in A_k$  represents a feasible route where the vehicle departs from depot (node  $k$ ) and visits nodes  $c_{i+1}, c_{i+2}, \dots, c_{j-1}$  and  $c_j$ , consecutively, and turns back to the depot, such that load and time windows restrictions on any arc are satisfied. After the construction of auxiliary graph, a minimum cost-path on  $G$  from node  $k$  to last node ( $c_s$ ) defines the optimal partition of the sequence which leads to a feasible solution originating from depot  $k$ .

An illustrative example for a chromosome is given in Fig. 1. In this example, TS\_VRP\_ATW consist of a facility (numbered as 0), 2 depots (numbered as 1 and 2) and 12 customers (numbered between 3 and 14). Customers with number between 3–7 and 8–14 are served from depot 1 and 2, respectively. Besides, the capacity of vehicle which is available between the facility and the depots is 200 units, and the one which is available between the depots and the customers is 40 units. Numbers in each gene determine the priorities of nodes relative to each other (Fig. 1a). Smaller numbers indicate that it will be routed earlier than the nodes having greater numbers. By considering relative priority values, initially the depots are assigned to the facility, and then, the customers to the depots (Fig. 1b). After this separation process, an auxiliary

graph is constructed for each group (Fig. 1c). In this network, the first node is the facility/depot, from which the route will begin, and the others are the customers that will be routed. Each arc in the graph represents a possible route and the cost of each arc gives the total cost of corresponding route. For example, arc (5–3) represents the feasible route departed from depot 1 and serving only customer 3 (i.e. 1–3–1), and arc (3–4) represents the feasible route departed from depot 1 and serving customers 7 and 4 (i.e. 1–7–4–1). Similarly, for each possible route, these arcs are constructed. It is important to note that infeasible arcs should not be included to the graph, due to capacity or time window restrictions. For example, arc (8–10) which represents the route 2–14–9–10–2 is not included to the graph since the total demand of customer (70 units) is greater than the vehicle capacity (40 units). After construction of auxiliary graph, the shortest path from first node to last node is found and optimal solution of corresponding sequence is obtained by the routes that are represented by arcs used in that path (Fig. 1d).

## 5.2. Crossover operator

The crossover operator is used to explore new solutions in the search space. In general, crossover operator is applied to obtain new solutions by exchanging some information between selected parents. Since our encoding scheme uses permutation encoding,

**Table 6**

Computational results for RC type test problems with wide time windows.

K	J	Problem name	LB	MIP model		Memetic algorithm	
				Gap	CPU	Gap	CPU
20	5	RC201_25*	303.91	0.00	0.31	0.00	2.34
		RC202_25*	303.87	0.00	0.28	0.00	1.65
		RC203_25*	303.85	0.00	0.30	0.00	1.51
		RC204_25*	303.85	0.00	0.36	0.00	1.70
		RC205_25*	303.89	0.00	0.30	0.00	1.36
		RC206_25*	303.89	0.00	0.42	0.00	1.34
		RC207_25*	303.88	0.00	0.55	0.00	1.34
		RC208_25*	303.87	0.00	0.30	0.00	2.27
Average		303.88	0.00	0.35	0.00	1.69	
40	10	RC201_50*	511.12	0.00	9.86	0.00	7.76
		RC202_50*	511.09	0.00	5.40	0.00	6.61
		RC203_50*	511.08	0.00	7.60	0.00	6.19
		RC204_50*	511.07	0.00	5.05	2.70	6.77
		RC205_50*	511.11	0.00	8.00	0.00	5.55
		RC206_50*	511.10	0.00	7.53	0.00	6.14
		RC207_50*	511.09	0.00	5.87	0.00	6.12
		RC208_50*	511.08	0.00	6.72	2.70	5.29
Average		511.09	0.00	7.00	0.68	6.30	
90	10	RC201_100	1020.50	3.60	3600.00	3.40	30.56
		RC202_100	1009.95	4.50	3600.00	4.50	27.50
		RC203_100	1023.05	3.10	3600.00	3.10	28.35
		RC204_100	1023.53	3.20	3600.00	3.10	27.62
		RC205_100	1018.68	3.60	3600.00	3.60	28.98
		RC206_100	1026.46	2.80	3600.00	2.80	29.79
		RC207_100	1022.77	3.20	3600.00	3.10	27.20
		RC208_100	1017.66	3.70	3600.00	3.70	27.29
Average		1020.33	3.46	3600.00	3.41	28.41	
Overall average		611.76	1.15	1202.45	1.36	12.13	

several crossover operators (i.e. Partially Mapped Crossover (PMX), Order Crossover (OX), Cycle Crossover (CX)) can be used directly. After some preliminary experiments, we utilize PMX operator as illustrated in Fig. 2. In this operator two cut points is selected randomly and the substring between these points is directly inherited from first parent ( $P_1$ ) to the same positions of the child. The substring is also used to determine the mapping relation. Then, each gene between cut points in the second parent ( $P_2$ ) are swapped using mapping relation, and copied to the empty positions of the child.

### 5.3. Local search

In this paper, for the purpose of improvement of newly produced individuals, local search algorithm is developed. In this algorithm, an individual that were obtained by crossover is converted to the real solution as explained in Section 5.1 and successive swap operations are applied to this solution. In swap operation, locations of two nodes which were randomly selected in the same route are exchanged. In each iteration of local search, the best solution is selected amongst  $|N|$  new neighbor obtained from current solution by swapping randomly selected nodes. If new solution is better than current solution, new solution is accepted as current solution. These steps are repeated until no improvement is obtained.

### 5.4. Algorithm outline

In our implementation, we use tournament selection mechanism. In this mechanism, two individual selected randomly from the population and the best individual is accepted as first parent ( $P_1$ ). The same steps are also used for the selection of second

parent ( $P_2$ ). We implement steady-state replacement strategy in which only one new solution is obtained by PMX operator, improved by local search and inserted to the current population, and then the worst chromosome is removed from population. It should be noted that local search algorithm is also implemented to the initial population to improve the solution quality.

According to these definitions, the overall pseudo-code of MA for solving TS\_VRP\_ATW is outlined in Fig. 3.

## 6. Computational results

This section presents our computational results to investigate the performance of the proposed MIP formulation and MA. The state-of-the-art LP/MIP solver CPLEX (version 10.2) is used to solve the MIP formulation. We used the default CPLEX parameters to solve the formulation, and the total computation time for CPLEX has been limited to one hour (3600 s). The proposed MA has been coded using C++ programming language. After preliminary experiments, population size is accepted as  $|N|$  and total number of generation as  $50 * |N|$ . All experiments were performed on Intel Xeon 3.16 GHz equipped (with 1 GB RAM).

Since there are no benchmark instances in the literature for TS\_VRP\_ATW, we have derived our test problems considering the VRP\_TW from test set generated by Solomon (1987). The detailed description of the test problems and the conversion steps are given in Appendix B.

In comparison, following performance measures are used: (i) Average solution time (CPU). (ii) Percentage Gap (Gap): calculated as  $100 * [(UB - LB) / LB]$  where  $LB$  is the best lower bound provided by CPLEX (MIP bound) in one hour computation time and  $UB$  is the upper bound found by CPLEX (best integer feasible solution of MIP) or MA.

Tables 1–6 present computational results for the proposed MIP formulation and MA on each TS\_VRP\_ATW test set, respectively. In these tables, while the first two columns represent the number of depots and customers, respectively, the third column gives the problem name. The names are indicated with “\*” for the test problem that optimal solution is found by CPLEX within one hour computation time. Next column shows the best lower bound (*LB*) provided by CPLEX. Finally, last two columns gives the percentage gap value (*Gap*) and computation time (*CPU*) of the proposed MIP formulation and MA, respectively. In these columns, better results are shown in bold.

As can be seen from Tables 1–3, optimal solutions are obtained by not only the proposed MIP model but also the proposed MA for all test problems with 25 and 50 nodes in a very short computing time.

For the test problems with 100 nodes, good quality solutions have been obtained by the proposed MIP formulation within one hour CPU time. The gap values are obtained as 8.16, 0.83 and 5.80, for C, R and RC-type of problems, respectively. According to results, it can be concluded that the test problems where the customers are clustered are more difficult to solve than other test problems. Meanwhile, good quality solutions are also obtained by the proposed MA. For C, R and RC-type of problems, 10.44, 4.57 and 10.74 gap values are obtained, respectively. Although better feasible solutions are obtained with the proposed MIP formulation, the solution times of the proposed MA are very short (23.14, 25.17 and 27.39 s, respectively). It is well known that the solution time of any procedure should be as small as possible for the operational problems such as node routing problems. Therefore, the proposed MA is preferable to the proposed MIP formulation especially for the large-size problems.

Similar results are also obtained for the test problems with wide time windows. Tables 4–6 gives the computational results for C, R and RC-type of test problems with wide time windows, respectively. As can be seen from these tables, the best solutions are obtained by both solution methods for all test problems with 25 nodes in a very short computation time.

As for the test problems with 50 nodes, the best solutions are obtained by the proposed MIP formulation. Meanwhile, good quality solutions are also obtained by the proposed MA in a short computation time. Only 5 out of 27 test problems (namely R201-50, R202-50, R203-50, RC204-50 and RC208-50) cannot optimally be solved with the proposed MA. For C, R and RC-type of problems, 0.00, 0.40 and 0.68 gap values are obtained within 5.81, 6.03 and 6.30 s, respectively.

With regard to the test problems with 100 nodes, while the optimal solutions are obtained for R-type test problems within 285.65 s on average, no optimal solutions are obtained for C and RC-type test problems (the gap values are obtained as 2.31 and 3.46) within predetermined maximum solution time. Meantime, good solutions results are obtained with the proposed MA in a short computation times. For C, R and RC-type of problems, the gap values are obtained as 2.30, 0.16 and 3.41 within 23.52, 27.70 and 28.41 s, respectively. Moreover, better results are obtained with the proposed MA for 4 of 8 in C-type and 3 of 8 in RC-type test problems, respectively.

## 7. Conclusion and future directions

In this study, a variant of the vehicle routing problem, called Two-Stage Vehicle Routing Problem with Arc Time Windows (TS\_VRP\_ATW) has been dealt with. We have proposed a Mixed Integer Programming (MIP) formulation and a Memetic Algorithm (MA) for the problem. The performance of the proposed solution methods have been investigated on a wide range of test problems

derived from the literature. Experimental results indicate that the proposed MIP formulation gives the optimal solutions for small- and medium-size test problems, and some large-size test problems. Moreover, the proposed MA achieves good quality solutions for the test problems in a short computation time. According to these results, we conclude that the problem with up to 50 nodes should be solved using proposed MIP formulation. Otherwise, we recommend that it should be solved using proposed MA.

Further research can be performed on exact solution methods (i.e. Branch and Cut) for the large-size test problems to obtain optimal solutions in a reasonable computation time. Moreover, the proposed MIP formulation and MA can be adapted to different type of this problem such as VRP with multiple arc time windows, multi-period VRP\_ATW, stochastic VRP\_ATW, and VRP\_ATW with heterogeneous vehicles. Complex distribution systems including arc time windows restrictions such as location routing problem with arc time windows, multistage VRP in which vehicles can be used in each stage, can also be investigated. Finally, the performance of the proposed MA can be increased by using new neighborhood mechanisms (i.e. *insert* and/or *swap-range* operators in addition to *swap*).

## Acknowledgements

We thank to the Associate Editor and anonymous referees for their helpful comments and suggestions, which greatly contributed to the improvement of the paper.

## Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at <http://dx.doi.org/10.1016/j.ejor.2013.05.001>.

## References

- Achuthan, N.R., Caccetta, L., Hill, S.P., 2003. An improved branch-and-cut algorithm for the capacitated vehicle routing problem. *Transportation Science* 37, 153–169.
- Ambrosino, D., Graziascutella, M., 2005. Distribution network design: new problems and related models. *European Journal of Operational Research* 165, 610–624.
- Ambrosino, D., Sciomachen, A., Scutella, M., 2009. A heuristic based on multi-exchange techniques for a regional fleet assignment location-routing problem? *Computers & Operations Research* 36, 442–460.
- Araque, J., Kudva, G., Morin, T.L., Pekny, J.F., 1994. A branch-and-cut algorithm for vehicle routing problems. *Annals of Operations Research* 50, 37–59.
- Augerat, P., Belenguer, J.M., Benavent, E., Corberan, A., Naddef, D., 1998. Separating capacity constraints in the CVRP using tabu search. *European Journal of Operational Research* 106, 546–557.
- Baldacci, R., Bartolini, E., Laporte, G., 2010. Some applications of the generalized vehicle routing problem. *Journal of the Operational Research Society* 61, 1072–1077.
- Baldacci, R., Mingozzi, A., Roberti, R., 2012. Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operational Research* 218, 1–6.
- Boudia, M., Prins, C., Reghioui, M., 2007. An Effective Memetic Algorithm with Population Management for the Split Delivery Vehicle Routing Problem, Hybrid Metaheuristics. In: Bartz-Beielstein, T., Blesa Aguilera, M., Blum, C., Naujoks, B., Roli, A., Rudolph, G., Sampels, M. (Eds.), vol. 4771. Springer Berlin/Heidelberg, pp. 16–30.
- Corberán, A., Prins, C., 2010. Recent results on arc routing problems: an annotated bibliography. *Networks* 56, 50–69.
- Dantzig, G.B., Ramser, J., 1959. The truck dispatching problem. *Management Science* 6, 80–88.
- Desrochers, M., Laporte, G., 1991. Improvements and extensions to the Miller–Tucker–Zemlin subtour elimination constraints. *Operations Research Letters* 10, 27–36.
- Doerner, K., Schmid, V., 2010. Survey: matheuristics for rich vehicle routing problems. In: Blesa, M., Blum, C., Raidl, G., Roli, A., Sampels, M. (Eds.), Hybrid Metaheuristics, vol. 6373. Springer, Berlin/Heidelberg, pp. 206–221.
- Eksioglu, B., Vural, A.V., Reisman, A., 2009. The vehicle routing problem: a taxonomic review. *Computers & Industrial Engineering* 57, 1472–1483.
- Funke, B., Grünert, T., Irnich, S., 2005. Local search for vehicle routing and scheduling problems: review and conceptual integration. *Journal of Heuristics* 11, 267–306.

- Hadjar, A., Soumis, F., 2009. Dynamic window reduction for the multiple depot vehicle scheduling problem with time windows. *Computers & Operations Research* 36, 2160–2172.
- Kara, I., 2008. Two Indexed Polynomial Size Formulations for Vehicle Routing Problems. Başkent University, Ankara/Turkey.
- Kara, I., Laporte, G., Bektas, T., 2004. A note on the lifted Miller–Tucker–Zemlin subtour elimination constraints for the capacitated vehicle routing problem. *European Journal of Operational Research* 158, 793–795.
- Kulkarni, R.V., Bhawe, P.R., 1985. Integer programming formulations of vehicle routing problems. *European Journal of Operational Research* 20, 58–67.
- Labadi, N., Prins, C., Reghioui, M., 2008. A memetic algorithm for the vehicle routing problem with time windows. *RAIRO – Operations Research* 42, 415–431.
- Liu, R., Jiang, Z., 2012. The close–open mixed vehicle routing problem. *European Journal of Operational Research* 220, 349–360.
- Lysgaard, J., Letchford, A.N., Eglese, R.W., 2004. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming* 100, 423–445.
- Miller, C., Tucker, A., Zemlin, R., 1960. Integer programming formulations and traveling salesman problems. *Journal of the ACM* 7, 326–329.
- Moccia, L., Cordeau, J.F., Laporte, G., 2011. An incremental tabu search heuristic for the generalized vehicle routing problem with time windows. *Journal of the Operational Research Society*.
- Mullaseril, P.A., 1997. Capacitated Rural Postman Problem with Time Windows and Split Delivery. Ph.D. thesis, University of Arizona, Tucson, Arizona.
- Ngueveu, S.U., Prins, C., Wolfler Calvo, R., 2010. An effective memetic algorithm for the cumulative capacitated vehicle routing problem. *Computers & Operations Research* 37, 1877–1885.
- Prins, C., 2004. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers and Operations Research* 31, 1985–2002.
- Prins, C., 2009. Two memetic algorithms for heterogeneous fleet vehicle routing problems. *Engineering Applications of Artificial Intelligence* 22, 916–928.
- Ralphs, T.K., 2003. Parallel branch and cut for capacitated vehicle routing. *Parallel Computing* 29, 607–629.
- Reghioui, M., Prins, C., Labadi, N., 2007. GRASP with path relinking for the capacitated arc routing problem with time windows. In: vol. 4448. LNCS, pp. 722–731.
- Solomon, M.M., 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research* 35, 254–265.
- Toth, P., Vigo, D., 2002. *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics. SIAM, Philadelphia.
- Vansteenwegen, P., Souffriau, W., Sörensen, K., 2010. Solving the mobile mapping van problem: a hybrid metaheuristic for capacitated arc routing with soft time windows. *Computers & Operations Research* 37, 1870–1876.
- Wøhlk, S., 2005. Contributions to Arc Routing. Ph.D. thesis, University of Southern Denmark.