Applied Mathematical Modelling 40 (2016) 6536–6549

Contents lists available at ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

Multi-depot vehicle routing problem with time windows considering delivery and installation vehicles

Heechul Bae[a], Ilkyeong Moon[b,*]

[a] IT Convergence Technology Research Laboratory, ETRI, Daejeon 305-700, Republic of Korea
[b] Department of Industrial Engineering, Seoul National University, Seoul 151-744, Republic of Korea

ARTICLE INFO

Article history:
Received 6 August 2014
Revised 15 January 2016
Accepted 26 January 2016
Available online 8 February 2016

Keywords:
Vehicle routing problem
Multiple depots
Delivery and installation vehicles
Genetic algorithm

ABSTRACT

We extend the multi-depot vehicle routing problem with time windows (MDVRPTW), a practical and challenging problem in logistics and supply chain management, to a study of service vehicles used for delivery and installation of electronics. This study shows that MDVRPTW results can be used to minimize fixed costs of the depots and the delivery and installation vehicles as well as expenses related to travel distances and labor. Along with a mixed integer programming model, we develop a heuristic and a genetic algorithm to identify a near-optimal solution. Computational results demonstrate that the proposed algorithms can efficiently be used to solve relatively large problems.

1. Introduction

The multi-depot vehicle routing problem with time windows (MDVRPTW) is used to determine the optimal set of fleet routes for satisfying the delivery demands of a customer set under time window constraints in several depots at different locations. It is considered a practical problem in the fields of transportation, distribution, and logistics. We consider the MDVRPTW under heterogeneous service vehicles and an additional service level constraint. In the MDVRPTW applied to delivery and installation vehicles, one vehicle is assumed to be dedicated solely to the delivery of products, while another is dedicated to transporting people and equipment needed for professional installation of the products. Each customer is associated with a specific delivery demand and choice of installation options. In this case, some customers require only delivery while others also need installation services. Generally, customers want their product installed as soon as possible after they receive the delivery. In addition to time window requirements of the basic VRP, another time constraint is imposed such that the installation service is provided within a certain time interval after product delivery. The delivery and/or the installation service can be outsourced to achieve service specialization in some industries. In such a case, to achieve the synchronization of the delivery and installation vehicle for each customer, different types of service vehicles and several scattered depots should be considered simultaneously. The coordination of delivery and installation is the main focus of this paper. Firms simultaneously attempt to meet customer demand and try to secure minimum transportation and labor costs. The problem arises in various supply chain management systems, including those associated with the electronics industry.

In the literature, several variants of the VRP exist. The multi-depot vehicle routing problem (MDVRP) is the problem of allocating customers to several depots, so that the optimal set of routes is determined simultaneously to serve the delivery demands of customers within scattered depots. The MDVRP was recently studied by Cordeau and Maischberger [1],

* Corresponding author. Tel.: +82 2 880 7151.
E-mail address: ikmoon@snu.ac.kr (I. Moon).

http://dx.doi.org/10.1016/j.apm.2016.01.059
S0307-904X(16)30055-5/

Escobar et al. [2], Shimizu and Sakaguchi [3], Subramanian et al. [4], and Vidal et al. [5]. Many heuristic methods have been developed in the context of the MDVRP, including the local search method (Subramanian et al. [4]), the tabu search algorithm (Cordeau and Maischberger [1]; Escobar et al. [2]), the genetic algorithm (Vidal et al. [5]), and the hierarchical hybrid meta-heuristic (Shimizu and Sakaguchi [3]).

The location routing problem (LRP) is a generalized problem of the MDVRP because it includes decisions of the number of depots and their locations. It calls for the determination of opening and operation of depots based on routes of vehicles. For a most recent survey on the LRP, the reader is referred to Drexl and Schneider [6], Drexl and Schneider [7], and Prodhon and Prins [8]. Recently, many heuristic methods have also been developed in the context of the LRP, including the local search method (Contardo et al. [9]; Hemmelmayr et al. [10]), the Lagrangian relaxation-based search method (Özyurt and Aksen [11]), the tabu search algorithm (Escobar et al. [12]; Escobar et al. [13]), the particle swarm algorithm (Liu and Kachitvichyanukul [14]), and the ant colony algorithm (Ting and Chen [15]). Sariçiçek and Akkuş [16] recently considered models in which the locations of the hubs were decided prior to the routing of unmanned aerial vehicles which were used for monitoring border security. Mousavi et al. [17] formulated the location routing problem under uncertainty as two mixed-integer linear programming problems and solved them via fuzzy possibilistic-stochastic programming. The MDVRPTW is not a special case of the LRP because heterogeneous service vehicles are considered under time windows and additional service level constraints are imposed.

Among problems involving several variations of vehicles, the VRP as applied to a heterogeneous fleet has been studied extensively (Golden et al. [18]; Liu [19]; Naji-Azimi and Salari [20]; Salhi et al. [21]). One related variant of the heterogeneous fleet problem is the fleet size and mix vehicle routing problem (FSMVRP), which simultaneously considers vehicles with different fixed costs within a fleet. In the FSMVRP, the number of available vehicles for each type is not given or limited. This variant was first studied by Golden et al. [18], and later by Liu [19]. Recently, the FSMVRP with backhauls was proposed by Salhi et al. [21]. In the VRPB, delivery and pickup service for customers can be considered simultaneously. A combined problem with the FSMVRP and VRPB is considered representative of a realistic routing and logistics distribution problem (Salhi et al. [21]). Recently, research on the VRP has also focused on more realistic situations such as overtime and capacity overloads. Moon et al. [22] presented a mixed integer programming model and a genetic algorithm for the VRPTWOV that included driver overtime and vehicle outsourcing. Lee and Kim [23] proposed a distributed dispatching method for large-scale pickup and delivery systems. Kim and Lee [24] developed an ant colony algorithm for a delivery schedule and proposed an expert system based on the developed algorithm to improve service quality for customers.

Interest in the VRP stems from its practical importance as well as considerable difficulty in finding optimal solutions for it. The VRP is regarded as one of the most challenging integer programming problems. Lenstra and Rinnooy Kan [28] showed that the VRP is NP-hard. The MDVRPTW is NP-hard in the strong sense, because it generalizes the VRP as further complexity is added through time windows and multiple depots. Due to the NP-hard nature of the problem, many good heuristic approaches have been developed for it. Here, we propose a heuristic method and a genetic algorithm for the MDVRPTW under installation vehicles and dual time constraints for customers. The concept of the service level for installation vehicles was first presented by Sun [25], who set it as the duration between delivery and installation services and also determined the delivery and installation vehicle routes and schedules for a fixed depot. Sun [25] synchronized the delivery and installation vehicles by using an endosymbiotic evolutionary algorithm based on a stochastic search mechanism. More recently, a vehicle scheduling problem for simultaneous delivery and installation was studied by Sim et al. [27] and Kim et al. [26]. Sim et al. [27] developed an ant colony algorithm for creating a vehicle schedule for simultaneous delivery and installation. The developed algorithm was applied to and tested in the context of a logistics company. Kim et al. [26] developed a genetic algorithm for creating a vehicle schedule.

The differences among the vehicle routing studies that include a service level and an investigation described in this paper are summarized in Table 1. We minimize transportation distances and labor time while satisfying dual time constraints of

Table 1
Characteristics of literature review.

Authors	This paper	Sun [25] and Kim et al. [26]	Sim et al. [27]
Number of depots	Multi-depot	Single-depot	Single-depot
Delivery time windows	Time windows	None	Time windows
Installation time constraint	Service level constraint	Service level constraint	Synchronization constraint
Total cost (Objective function)	Depot cost, Vehicle cost, Transportation cost, Labor cost	Transportation cost	Vehicle cost, Transportation cost
Decision	Number of depots, Number of vehicles, Routing	Routing	Number of vehicles, Routing
Methodology	MILP, Genetic algorithm	MINP[a], Endosymbiotic evolutionary algorithm, Genetic algorithm	MILP, Ant colony algorithm

[a] MINP(Mixed integer nonlinear programming).

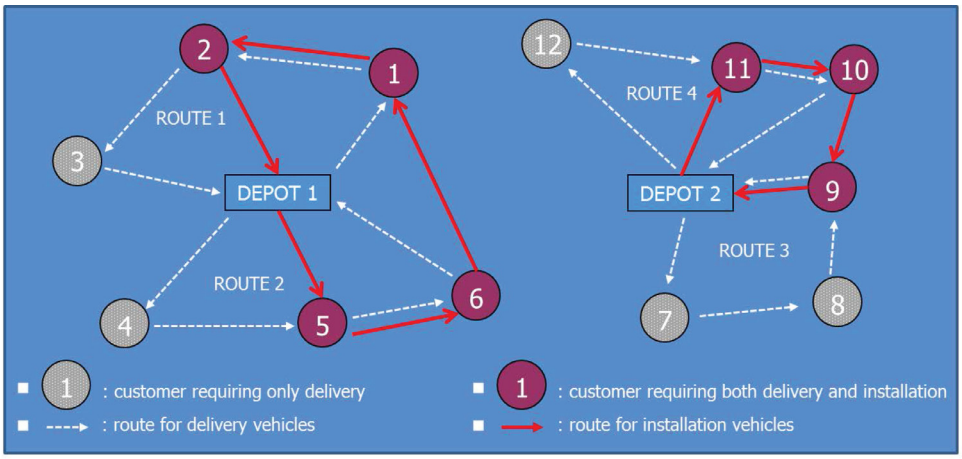

Fig. 1. MDVRPTW with delivery and installation vehicles that satisfy dual time constraints.

delivery and installation vehicles (see Fig. 1). We determine a set of depots from which routes originate that satisfy both delivery demands and installation requirements of customers such that the fixed costs of the depots, delivery and installation vehicles, as well as transportation and labor costs are minimized. The purpose of this paper is twofold: It extends the MDVRPTW by accounting for delivery and installation vehicles with an additional service requirement and by addressing the time interval between delivery and installation services, and (ii) it shows the development of both a heuristic and a genetic algorithm for the MDVRPTW.

This paper is organized as follows: In Section 2, the assumptions and notation are presented; furthermore, we develop a mixed integer programming model. A heuristic algorithm for the MDVRPTW is presented in Section 3. A genetic algorithm for the MDVRPTW is developed in Section 4. In Section 5, a numerical example and computational experiments illustrate the solution procedure, and the results are compared with heuristic solutions. The paper ends with conclusions in Section 6.

2. Problem formulation

The following assumptions are used in the multi-depot vehicle routing problem with time windows (MDVRPTW):

(1) Each vehicle must leave from and return to the same depot.
(2) Each customer is served exactly once by one vehicle.
(3) The customer demand along the route does not exceed the vehicle capacity.
(4) All of the customers with known demands are assigned to vehicles.
(5) The total time of a route does not exceed the maximum vehicle route time. The maximum vehicle route times are equivalent to the time window of the depot.
(6) The time window constraints of each customer delivery must be satisfied.
(7) The difference between the arrival times of delivery and installation vehicles at the same customer must not exceed the service level (maximum allowable time interval).

The following notation is used in the model:

Sets
I : set of delivery nodes
A : set of installation nodes, $A \subset I$
J : set of depot nodes
N : set of all nodes, $N = I \cup J$
L : set of installation and depot nodes, $L = A \cup J$
K : set of delivery vehicles
S : set of installation vehicles

Parameters
F_j : fixed cost for depot j
CF_k : fixed cost for vehicle k

CF_s : fixed cost for vehicle s
CT_k : transportation cost for vehicle k per unit time
CT_s : transportation cost for vehicle s per unit time
CR_k : labor cost of the delivery person for vehicle k per unit time
CR_s : labor cost of the installation engineer for vehicle s per unit time
t_{ij} : transportation time between nodes i and j
st_i : service time of the installation vehicle at customer i
e_i : earliest time at customer i
l_i : latest time at customer i
r_k : maximum route time allowed for vehicle k
d_i : demand at customer i
q_k : capacity of vehicle k
SL : service level for installation vehicles
M : a sufficiently large number

Decision variables
a_i : arrival time of the delivery vehicle at customer i
b_i : arrival time of the installation vehicle at customer i
a_{jk} : arrival time of delivery vehicle k at depot j
b_{js} : arrival time of installation vehicle s at depot j
wd_i : waiting time of the delivery vehicle at customer i
wt_i : waiting time of the installation vehicle at customer i

$$x_{ijk} = \begin{cases} 1, & \text{if delivery vehicle } k \text{ travels directly from } i \text{ to } j \\ 0, & \text{otherwise} \end{cases}$$

$\forall i, j \in N, k \in K$

$$x_{ijs} = \begin{cases} 1, & \text{if installation vehicle } s \text{ travels directly from } i \text{ to } j \\ 0, & \text{otherwise} \end{cases}$$

$\forall i, j \in L, s \in S$

$$Z_j = \begin{cases} 1, & \text{if depot } j \text{ is used} \\ 0, & \text{otherwise} \end{cases}$$

$\forall j \in J$

The formulation of the MDVRPTW can be stated as follows:

$$\text{Min} \sum_{j\in J} F_j Z_j + \sum_{i\in N}\sum_{j\in N}\sum_{k\in K} CT_k t_{ij} x_{ijk} + \sum_{i\in L}\sum_{j\in L}\sum_{s\in S} CT_s t_{ij} x_{ijs} + \sum_{k\in K}\sum_{i\in J}\sum_{j\in I} CF_k x_{ijk} + \sum_{s\in S}\sum_{i\in J}\sum_{j\in A} CF_s x_{ijs} + \sum_{j\in J}\sum_{k\in K} CR_k a_{jk} + \sum_{j\in J}\sum_{s\in S} CR_s b_{js}, \tag{1}$$

$$\text{s.t.} \sum_{i\in N} x_{ijk} = \sum_{i\in N} x_{jik}, \quad \forall j \in N, k \in K, \tag{2}$$

$$\sum_{i\in L} x_{ijs} = \sum_{i\in L} x_{jis}, \quad \forall j \in L, s \in S, \tag{3}$$

$$\sum_{k\in K}\sum_{i\in N} x_{ijk} = 1, \quad \forall j \in I, \tag{4}$$

$$\sum_{s\in S}\sum_{i\in L} x_{ijs} = 1, \quad \forall j \in A, \tag{5}$$

$$\sum_{i\in J}\sum_{j\in I} x_{ijk} \le 1, \quad \forall k \in K, \tag{6}$$

$$\sum_{i\in J}\sum_{j\in A} x_{ijs} \le 1, \quad \forall s \in S, \tag{7}$$

$$\sum_{i\in N} x_{ijk} \le Z_j, \quad \forall j \in J, k \in K, \tag{8}$$

$$\sum_{i\in L} x_{ijs} \le Z_j, \quad \forall j \in J, s \in S, \tag{9}$$

$$\sum_{i\in I}\sum_{j\in N} d_i x_{ijk} \le q_k, \quad \forall k \in K, \tag{10}$$

$$a_j \ge a_i + wd_i + t_{ij} + M\left(\sum_{k\in K} x_{ijk} - 1\right), \quad \forall i \in N, j \in I, \tag{11}$$

$$b_j \ge b_i + st_i + wt_i + t_{ij} + M\left(\sum_{s\in S} x_{ijs} - 1\right), \quad \forall i \in L, j \in A, \tag{12}$$

$$r_k \ge a_{jk} \ge a_i + wd_i + t_{ij} + M(x_{ijk} - 1), \quad \forall i \in I, j \in J, k \in K, \tag{13}$$

$$r_s \ge b_{js} \ge b_i + st_i + wt_i + t_{ij} + M(x_{ijs} - 1), \quad \forall i \in A, j \in J, s \in S, \tag{14}$$

$$e_i \le a_i + wd_i \le l_i, \quad \forall i \in I, \tag{15}$$

$$a_i + wd_i \le b_i + wt_i \le SL + a_i + wd_i, \quad \forall i \in A, \tag{16}$$

$$a_j = b_j = st_j = wd_j = wt_j = 0, \quad \forall j \in J, \tag{17}$$

$$x_{ijk} = \{0, 1\}, \quad \forall i, j \in N, k \in K, \tag{18}$$

$$x_{ijs} = \{0, 1\}, \quad \forall i, j \in L, s \in S, \tag{19}$$

$$Z_j = \{0, 1\}, \quad \forall j \in J. \tag{20}$$

The objective function (1) sums all of the costs, including the fixed costs of the depots and the delivery and installation vehicles as well as those related to transportation and labor. The fixed costs are amortized so that they can be directly compared with operating costs. Constraints (2) and (3) ensure that the vehicle that leaves the customer is the same vehicle as the one that visits the customer. Constraints (4) and (5) stipulate that every customer is assigned to one delivery vehicle and that each customer needing installation is assigned to exactly one installation vehicle, respectively. Constraints (6) and (7) define whether or not each vehicle should be assigned to a depot. Constraints (8) and (9) guarantee that vehicles can be assigned to a depot only if that depot is open. Constraint (10) is the capacity constraint of the delivery vehicles. Constraints (11)–(14) ensure compatible arrival times for service, waiting, and transportation between nodes i and j. The service time is considered only for an installation vehicle. Constraints (15) and (16) specify the time windows at each customer, and the service level is the time interval between delivery and installation services. Constraint (17) defines the depot times. Finally, Constraints (18), (19), and (20) are the binary requirements for the decision variables regarding the delivery vehicle route, the installation vehicle route, and the depot location, respectively.

3. Heuristic algorithm

In this section, we develop a heuristic algorithm for the MDVRPTW considering delivery and installation vehicles. The problem is decomposed into two distinct problems: The first involves solving the MDVRPTW considering only delivery vehicles, and the second involves solving the MDVRPTW considering only installation vehicles with respect to requirements of the service level. Based on the results of the first problem, the second problem is generalized to the case of the VRP with time windows because the delivery vehicle's arrival time to the installation customers is calculated and the service level is given.

For each decomposed problem, the heuristic first assigns each customer to its nearest depot and constructs routes for the customer set of each depot. A routing construction heuristic is similar to the nearest neighbor heuristics (NNH) (Reinelt [29]).

procedure nearest neighbor (Reinelt [29])

(1) Select an arbitrary node j, set $l = j$ and $T = \{1, 2, \ldots, n\}\backslash\{j\}$.
(2) As long as $T \neq \emptyset$ do the following.
  (2.1) Let $j \in T$ such that $c_{lj} = \min\{c_{li} \mid i \in T\}$.
  (2.2) Connect l to j and set $T = T\backslash\{j\}$ and $l = j$.
(3) Connect l to the first node (selected in Step (1)) to form a tour. end of nearest neighbor.

Within each customer set, vehicle routes are determined to minimize the total transportation distances, and they can be made by connecting a pair of consecutive customers served from a depot. In our study, the NNH was used for constructing routes: First, two nodes with the minimum distances were identified and then the next node as close as possible to the

previous one was selected until a route is established with all customers and one depot. Specifically, the following heuristic procedure is repeated until all of the customers within each group are assigned to a route:

Step 1. Calculate the transportation distances for every pair of nodes including customers and a depot.
Step 2. Select a pair of nodes with the minimum transportation distances.
Step 3. Connect the nodes from a start node to an end node by a visiting sequence in ascending order of their earliest time.
Step 4. Connect the next node as close as possible to the previously connected end node in order to make a route until all of the nodes are assigned to the route.
Step 5. Arrange the route so that a vehicle leaves from a depot and returns to the same depot.
Step 6. Assign vehicles to the route sequentially until the demand, time windows, and service level of all of the customers are satisfied.

One of the main characteristics of this algorithm is that infeasible solutions are not allowed throughout the search. Feasible solutions are achieved through the simple assignment of vehicles. Feasible routes are made by a sequential assignment method of a vehicle that meets the time windows, maximum route time, and capacity constraints at each customer node. In other words, we assign vehicle k iteratively to the constructed route as long as its capacity and the maximum route time is not exceeded and each customer's time window is not violated. This simple method allowed us to make feasible routes easily.

4. Genetic algorithm

In this section, we develop a genetic algorithm for the vehicle scheduling problem for delivery and installation services. We use a genetic scheme to simultaneously find a good assignment and vehicle routes. Genetic algorithms, which have been widely used in various areas, are stochastic search algorithms based on the mechanisms of natural selection (Gen and Cheng [30]). In contrast to the use of conventional search techniques, one starts a genetic algorithm with an initial set of randomly generated solutions called a *population*. Each individual in the population is called a *chromosome* and represents a solution to the problem. A chromosome evolves through successive iterations called *generations*. During each generation, genetic operators, such as crossover, mutation, and selection, create chromosomes with new qualities that more or less allow the individual to adapt to an environment. Each chromosome is then evaluated by specific measures of fitness as a solution. Any genetic algorithm used to solve a problem must possess certain basic components that depend on the problem under investigation. We explain our overall strategies in the following subsection, including the genetic representation, penalty and fitness functions, and the genetic operators. To undertake the overall procedure, let $P(t)$ and $C(t)$ be the parents and offspring in the current generation t, respectively. The pseudo-code for solving the vehicle scheduling problem for delivery and installation is outlined in Fig. 2.

4.1. Representation and initialization

The proper representation of a solution plays a key role in the development of a genetic algorithm. The assignment and sequence have to be decided simultaneously for the problem.

We use a random number representation for the assignment chromosome. Each assignment chromosome in the population is used to assign customers to depots by a simple decoding process. $\text{Priority}_i = \lfloor (J-1) \times Gene(i) \rfloor + 1$, $\lfloor G \rfloor$ is the function which finds the integer number less than G. Priority_i represents a priority of depot selection and J is the total number of depot nodes. Each customer has priorities of depot selection by distances from a customer to depots. Priority 1 means that a customer is assigned to its nearest depot and priority J is assigned to its depot of the longest distance. This decoding process easily converts a random number chromosome between 0 and 1 into integer representation of depot selection by distance priority.

As in most genetic algorithms for the traveling salesman problem, a chromosome is simply a permutation of n customer nodes (Gen and Cheng [30]). The output can be interpreted as the order in which a vehicle must visit all the assigned customers if the same vehicle is to make successive trips to each customer. To represent the vehicle scheduling problem for delivery and installation, we used permutation-based encoding and decoding methods in our investigation. The length of the permutation chromosome reflects the total number of customers. A gene in a chromosome is characterized by the locus (position) and the allele (value) (Gen et al. [31]). In our model, each allele represents the customers on the route, and each locus represents the position of the customer. In addition, $v(i)$ represents the priority of customer i. Fig. 3 shows an example of the genes sequenced in a chromosome of seven customers. The vehicle visitation sequence for customers is 1-3-4-2-6-5-7.

The decoding procedure sequentially cuts a permutation chromosome into routes that satisfy the time windows, maximum route time, and capacity constraints, and leads to solution of a vehicle scheduling problem that matches up with the sequence. One of the main characteristics of the decoding procedures is that infeasible solutions are not allowed throughout the search. Feasible solutions are achieved through the simple and sequential assignment of vehicles. The following decoding

Genetic algorithm for the vehicle scheduling problem for delivery and installation:

Begin

$t \leftarrow 0$;
initialize $P(t)$ by the permutation encoding routine;
evaluate $P(t)$ by the permutation decoding routine;
while (not termination condition) do

create $C'(t)$ from $P(t)$ by the order crossover routine;
create $C(t)$ from $C'(t)$ by the swap mutation routine;
evaluate $C(t)$ by the permutation decoding routine;
select $P(t+1)$ from $P(t)$ and $C(t)$ by the ranking selection routine;
$t \leftarrow t+1$;

end

End

Fig. 2. The pseudo-code for the overall procedure for solving the MDVRPTW.

Customer, i	1	2	3	4	5	6	7
Sequence, $v(i)$	1	4	2	3	6	5	7

Fig. 3. Example of a sequence chromosome for seven customers.

procedure for the permutation chromosome, where the binary route variable x_{ijk} is equal to one if vehicle k travels directly from node i to node j, reflects a delivery route:

Procedure. Decoding of the permutation chromosome for delivery routes

Input: $v(i)$
Output: x_{ijk}, a_i
Step 1. Put $x_{ijk} = 0, \forall ij \in I, k \in K$.
Set the vehicle number to $k = 1$, the arrival time to $a_m^* = 0$ and the sum of demand at customer to $dsum_k^* = 0$.
$q = \text{argmax}\{v(i), i \in I\}$; set the total number of customers.
Step 2. $m = \text{argmim}\{v(i), i \in I\}$; select the first priority node.
If $v(m) = q$, then go to Step 5.
$n = \text{argmim}\{v(i), i \in I \backslash m\}$; select the second priority node.
$a_m^* = \max\{a_m^*, e_m\}$ and $dsum_k^* = dsum_k^* + d_m$.
Step 3. Check the feasibility of the constraints.
Calculate $a_n^* = a_m^* + t_{mn}$ and $dsum_k^* = dsum_k^* + d_n$.
If $a_n^* \le l_n$, $a_n^* \le r_k$ and $dsum_k^* \le q_k$, then $x_{mnk} = 1$.
Otherwise, $k = k+1$, $a_m^* = 0$ and $dsum_k^* = 0$; go to Step 3.
Step 4. If $x_{mnk} = 1$, then $v(m) = q$; update the priority of the gene.
Step 5. If $v(i) = q, \forall i \in I$, then calculate the total cost and stop.
Otherwise, go to Step 2.

The decoding procedure for the permutation chromosome for the installation routes can be stated as follows where the binary route variable x_{ijs} is equal to one if vehicle s travels directly from node i to node j:

Procedure. Decoding of the permutation chromosome for the installation routes

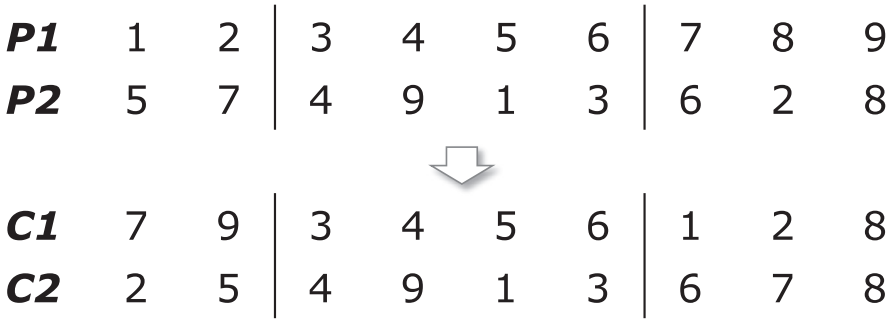

Fig. 4. Example of OX for the sequence chromosomes.

Input: $v(i)$
Output: x_{ijs}, b_i
Step 1. Put $x_{ijs} = 0, \forall ij \in I, s \in S$.
Set the vehicle number to $s = 1$ and the arrival time to $b_m^* = 0$.
$q = \text{argmax}\{v(i), i \in I\}$; set the total number of customers.
Step 2. $m = \text{argmim}\{v(i), i \in I\}$; select the first priority node.
If $v(m) = q$, then go to Step 5.
$n = \text{argmim}\{v(i), i \in I \backslash m\}$; select the second priority node.
$b_m^* = \max\{b_m^*, b_m + wt_m + st_m\}$.
Step 3. Check the feasibility of the constraints.
Calculate $b_n^* = b_m^* + t_{mn}$.
If $b_n^* - a_n + wd_n \le SL$ and $b_n^* \le r_s$, then $x_{mns} = 1$.
Otherwise, $s = s + 1$ and $b_m^* = 0$; go to Step 3.
Step 4. If $x_{mns} = 1$, then $v(m) = q$; update the priority of the gene.
Step 5. If $v(i) = q, \forall i \in I$, then calculate the total cost and stop.
Otherwise, go to Step 2.

The initial population is typically created either randomly or using modifications of well-known construction heuristics (Bräysy and Gendreau [32]). We apply the heuristic solution described earlier to the initial population instead of generating chromosomes randomly. An initial population with a chromosome of heuristic solution is more likely to increase the speed of convergence in large size problems. Because an initial population with the heuristic solution has a possible drawback of early convergence to local optimal solutions in small size problems, an initial population is generated properly according to problem sizes. A population of size 50 was used for computational experiments.

4.2. Genetic operators

A simple genetic algorithm that yields good results for many practical problems features three operators: crossover, mutation, and selection. It is the combination of these genetic operators that affects the performance of the genetic algorithm and the scope of the search (Gen and Cheng [30]). Therefore, it is important to select proper genetic operators for each problem. Accumulated information is usually exploited by the selection mechanisms and is partially exploited by the crossover mechanism, while new regions of the search space are explored by the mutation operator.

For a permutation-based sequence representation, several crossover operators have been proposed, such as the partial-mapped crossover (PMX), order crossover (OX), cycle crossover (CX), position-based crossover (PX), and heuristic crossover (Gen et al. [31]). PMX and OX are conventional two-cut point crossovers and PX is a variation of a uniform crossover operator. A uniform crossover operator such as PX enables us to exchange bits rather than segments, whereas PMX and OX are to choose two cut points uniformly at random and, based on these cut points, to exchange segments. In this problem, we use OX in order to preserve and generate a good segments schema from a pair of parents. Fig. 4 depicts an example of OX for the sequence chromosomes (Gen and Cheng [30]). For the assignment chromosomes, we use a uniform crossover operator which enables us to exchange bits rather than segments. Uniform crossover first generates a random crossover mask and then exchanges relative genes between parents according to the mask (Gen and Cheng [30]). Fig. 5 shows an example of a uniform crossover for the assignment chromosomes.

Several mutation operators have also been proposed for permutation representation, such as swap, inversion, and insertion mutations. We used the swap mutations for the sequence chromosomes. Two digits are randomly selected and their positions are exchanged in the swap mutation. Fig. 6 depicts an example of swap mutation for the sequence chromosomes. For the assignment chromosomes, a digit is randomly selected with probability and a new random number between zero and one is generated.

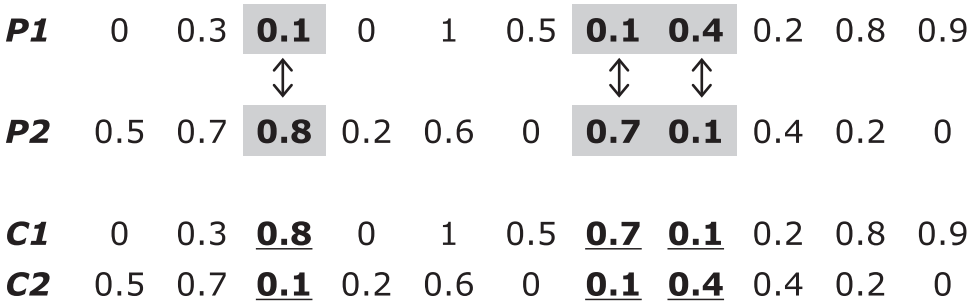

Fig. 5. Example of uniform crossover for the assignment chromosomes.

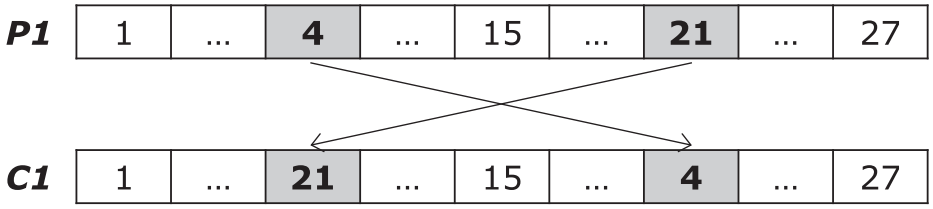

Fig. 6. Example of swap mutation for the sequence chromosomes.

Table 2
Computational results for parameter selection.

P_m / P_c	Mean error (%)		
	0.4	0.6	0.8
0.2	0.045	0.038	0.020
0.3	0.039	0.046	0.049
0.4	0.053	0.045	0.046

Selection is a process in which individual strings create a new population for the next generation based on their fitness function values. In this paper, parents are chosen with a rank-based mechanism. A ranking approach offers a smoother selection probability curve. This prevents good strings from totally dominating the search at an early stage.

4.3. Termination condition and parameter selection

In this study, we performed computational experiments with the following values of the GA parameters. In order to select the appropriate values of the GA parameters, we conducted a pilot experiment using 10 randomly generated problems. Three different crossover and mutation rates are considered. Computational results for parameter selection are shown in Table 2.

- Population size : 50
- Crossover rate (P_c) : 0.8
- Mutation rate (P_m) : 0.2

We considered as many as 6000 generations, even for small problems, but we used tighter termination conditions to generate a solution with little computational effort for large problems. The genetic algorithm makes a stop once no improvement is made in 100 generations or the best individual is not allowed to improve more than 0.01% over 500 generations.

5. Computational experiment

In this section, we present computational results for the MDVPRTW. Our computational experiments were performed on three instances: small problem, medium problem, and the modified Cordeau et al.'s [33] problem. Specifically, we compare the performances of the mixed integer program and the genetic algorithm. The mixed integer program was implemented and solved with LINGO 10.0. All computational tests presented here were performed on an Intel Core(TM) i7 with 2.67 GHz and 4GB RAM.

Example I (Small problem and sensitivity analysis)**.** First, we solved a small problem. The parameters for customers in a small problem are shown in Tables 3 and 4. The other parameters, for comparison, are as follows:

- Number of depots: 2
- Number of delivery nodes: 5

Table 3
Data on customer demands (in units) and time constraints (in minutes).

Customer	1	2	3	4	5
Demand	10	7	13	19	26
Service time	–	–	30	40	55
Earliest time	24	79	56	38	47
Latest time	225	229	170	278	113

Table 4
Distance matrix of the vehicle operation for each customer and depot (in minutes).

		Customer					Depot	
		1	2	3	4	5	6	7
Customer	1	0.0	24.8	13.6	34.9	18.3	17.5	11.6
	2	24.8	0.0	35.4	25.1	8.4	13.8	18.1
	3	13.6	35.4	0.0	36.6	30.5	23.9	17.8
	4	34.9	25.1	36.6	0.0	30.3	18.3	23.3
	5	18.3	8.4	30.5	30.3	0.0	14.3	15.4
Depot	6	17.5	13.8	23.9	18.3	14.3	0.0	
	7	11.6	18.1	17.8	23.3	15.4		0.0

Table 5
Vehicle routing times for the small problem.

Customer (Depot)		6	4	3	1	5	2	6
Delivery vehicle	Arrival time	–	18.3	74.6	88.2	106.5	114.8	128.7
	Waiting time	–	19.7	0.0	0.0	0.0	0.0	–
	Delivery time	–	38.0	74.6	88.2	106.5	114.8	–
	Earliest time	–	38.0	56.0	24.0	47.0	79.0	–
	Latest time	–	278.0	170.0	225.0	113.0	229.0	–
Installation vehicle	Arrival time	–	18.3	114.6	–	175.2	–	244.4
	Waiting time	–	19.7	0.0	–	0.0	–	–
	Installation time	–	38.0	114.6	–	175.2	–	–
	Service time	–	40.0	30.0	–	55.0	–	–

- Number of installation nodes: 3
- Number of vehicles: 4 (2 delivery vehicles and 2 installation vehicles)
- Service level: 70 min
- Vehicle capacity: 100 units
- Maximum transportation time: 300 min
- Fixed cost for depots: $100/depot
- Fixed cost for delivery and installation vehicles: $10/vehicle
- Transportation cost for delivery and installation vehicles: $1/minute
- Labor cost: $1/minute

The solution is comprised of $Z_6 = 1, x_{642} = 1, x_{644} = 1,$ one delivery and installation route, and a total cost of $701.8. Each vehicle's routing times for the small problem are summarized in Table 5. The table shows the delivery vehicle attempts to minimize the total transportation time within the time windows and the installation vehicle attempts to minimize the arrival time at the depot and meet the customer's service level. The resulting transportation routes are as follows:

- Delivery Vehicle: 6-4-3-1-5-2-6
- Installation Vehicle: 6-4-3-5-6.

Table 6 shows the sensitivity analysis with respect to the service level. This analysis reveals that if the service level is relaxed (i.e., larger time intervals are allowed between delivery and installation), the solutions will be better due to decreased labor costs. Table 7 also shows the results of sensitivity analysis taking into account labor costs and the service level. The sensitivity analyses include a general problem with vehicle routing service by one vehicle (i.e., delivery and installation are handled simultaneously such that the delivery person and installation engineer travel together). Table 7 includes the results of sensitivity analysis serviced by separate vehicles with varying service levels and labor costs. The GAP in parentheses shows the percentage gap between the cost of service by one vehicle and that of service by separate vehicles. This analysis clearly shows that the solution achieved by separate vehicles is significantly better than that procured by one vehicle. In

Table 6
Sensitivity analysis on the value of the service level.

Service level		Selected depot	Routes	Transportation cost($)	Labor cost($)	Total cost($)
Serviced by one vehicle		6	6-5-2-1-3-4-6	115.9	547.3	773.2
Serviced by separate vehicles	0	6	6-4-3-1-5-2-6 6-4-5-6 6-3-6	219.6	436.5	786.0
	20	6	6-4-3-1-5-2-6 6-4-5-6 6-3-6	219.6	434.7	784.2
	40	7	7-4-2-5-1-3-7 7-4-5-3-7	208.5	413.2	741.7
	60	7	7-4-2-5-1-3-7 7-4-5-3-7	208.5	393.3	721.7
	70	6	6-4-3-1-5-2-6 6-4-3-5-6	208.7	373.1	701.8

Table 7
Sensitivity analysis on the value of labor costs and the service level.

Service level		Cost factor			
		$CR_k = 1, CR_s = 1$	$CR_k = 1, CR_s = 2$	$CR_k = 2, CR_s = 2$	$CR_k = 2, CR_s = 4$
Serviced by one vehicle		$773.2	$1046.8	$1320.5	$1867.7
Serviced by separate vehicles	0	$786.0 (−1.7%)	$1067.9 (−2.0%)	$1222.6 (7.4%)	$1800.2 (3.6%)
	20	$784.2 (−1.4%)	$1047.9 (−0.1%)	$1218.9 (7.7%)	$1760.2 (5.8%)
	40	$741.7 (4.1%)	$983.4 (6.1%)	$1155.0 (12.5%)	$1638.3 (12.3%)
	60	$721.7 (6.7%)	$963.4 (8.0%)	$1115.0 (15.6%)	$1598.3 (14.4%)
	70	$701.8 (9.2%)	$946.2 (9.6%)	$1074.9 (18.6%)	$1563.7 (16.3%)

Table 8
Comparison of the results from the mixed integer program and the genetic algorithm.

J	I	A	K	S	q	Mixed integer program			Genetic algorithm	
						Average evaluation time	Objective value($)	Remark	Objective value($)	Remark
2	5	3	2	2	100	107 s	701.8	*	701.8	*
4	20	5	2	3	200	20 h	3175.8	*	3175.8	*
5	20	10	2	3	200	27 h	4401.4	*	4401.4	*

particular, as labor costs and service levels increase, the total cost of service by separate vehicles is significantly lower than those accrued through use of one vehicle.

Example II (Medium problem)**.** Table 8 shows a comparison of the results from the mixed integer program and the genetic algorithm. In the table, the asterisk indicates an optimal solution. Each objective function has the same value. From this result, the applicability of our model is demonstrated. Because of the characteristics of the NP-hard problem, the computational time increases exponentially as the number of customers is increased. The genetic algorithm obtains optimal solutions within two minutes for the examples in Table 8.

Example III (Modified Cordeau et al.'s instances)**.** In addition, we conducted computational experiments to evaluate the performance of the heuristic and the genetic algorithm. Cordeau et al. [33]'s instances were considered for comparison purposes. A values that represent the number of installation nodes were added to our data based on Cordeau et al. [33]. The number of installation nodes, A, equals one-half the number of customers. The difference between groups (a) and (b) lies only in the length of the time windows. In group (a), narrow time windows are generated by choosing a uniform random number e_i in the interval [60, 480] and then choosing a uniform random number l_i in the interval $[e_i + 90, e_i + 180]$. In group (b), larger time windows are created by choosing the random numbers e_i and l_i in the interval [60, 300] and $[e_i + 180, e_i + 360]$, respectively (Cordeau et al. [33]). The characteristics of Cordeau et al. [33]'s instances are shown in Table 9. We first show the comparison with a general case of vehicle scheduling problem using the Cordeau's instances. Fig. 7 illustrates a typical improvement of total costs according to the service level. The service level of zero means that a delivery person and an installation engineer travel separately, but the installation engineer should arrive at an installation customer earlier or the same time with the delivery person.

We improve the initial solutions based on the heuristic solution in each instance with different service levels by the genetic algorithm, and the CPU time of the genetic algorithm is set to 3 min for the instances. As shown in this figure,

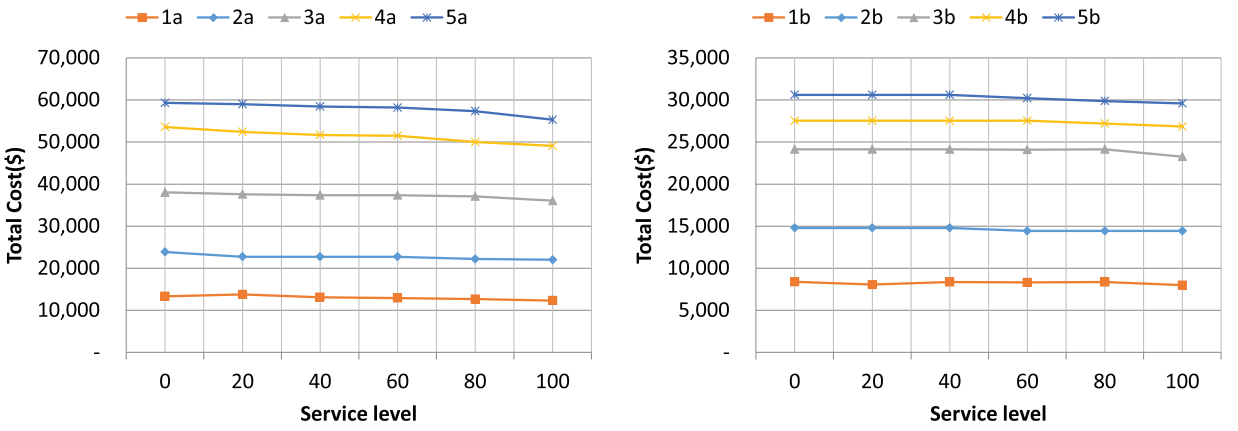

Fig. 7. Total costs graph according to the different service level.

Table 9
Characteristics of the Cordeau's instances of the MDVRPTW.

No.	I	A	J	q	No.	I	A	J	q
1a	48	23	4	200	1b	48	23	4	200
2a	96	47	4	195	2b	96	47	4	195
3a	144	71	4	190	3b	144	71	4	190
4a	192	95	4	185	4b	192	95	4	185
5a	240	119	4	180	5b	240	119	4	180
6a	288	143	4	175	6b	288	143	4	175
7a	72	35	6	200	7b	72	35	6	200
8a	144	71	6	190	8b	144	71	6	190
9a	216	107	6	180	9b	216	107	6	180
10a	288	143	6	170	10b	288	143	6	170

Table 10
Comparison of the results of the heuristic method and the genetic algorithm.

No.	Heuristic			Genetic algorithm			GAP(%)
	CPU time (seconds)	Distance	Objective value ($)	CPU time (seconds)	Distance	Objective value ($)	
1a	13	2225.4	14273.3	180	2691.0	12325.6	13.6
2a	25	3759.6	22722.1	180	3885.0	22021.2	3.1
3a	37	6186.5	37149.1	180	6600.6	36047.5	3.0
4a	49	8265.5	50406.0	180	9821.4	49093.1	2.6
5a	60	8401.4	56487.8	180	8996.5	55297.1	2.1
6a	71	8898.3	62685.8	180	9654.4	61952.2	1.2
7a	20	3156.1	20220.7	180	3248.3	18730.5	7.4
8a	38	6234.3	37927.2	180	6651.1	37556.9	1.0
9a	55	7225.0	54024.3	180	8305.7	53427.4	1.1
10a	72	10659.7	73337.0	180	11232.9	71553.7	2.4
1b	14	1854.0	8422.5	180	1904.6	7987.5	5.2
2b	25	3061.9	14436.0	180	3061.9	14436.0	0.0
3b	38	5222.0	24140.0	180	5195.1	23268.8	3.6
4b	50	6236.3	26839.2	180	6236.3	26839.2	0.0
5b	61	5989.4	29589.3	180	5989.4	29589.3	0.0
6b	71	7729.7	37715.2	180	7729.7	37715.2	0.0
7b	20	2638.6	12973.2	180	3097.6	12473.5	3.9
8b	38	4635.6	21077.5	180	4635.6	21077.5	0.0
9b	55	5511.0	25473.5	180	6149.2	24772.2	2.8
10b	73	7559.3	35946.9	180	7559.3	35946.9	0.0

a separate scheduling of delivery and installation service is more effective when time windows are more tight and large service levels are allowed between delivery and installation. Because the length of time windows in group (b) is quite large, the improvement in group (b) according to the service levels is less than that in group (a). Moreover, because no exact solutions are available for benchmarking the problem, we compared the results from our heuristic with those from the genetic algorithm. Table 10 displays the computational results of the algorithms.

We computed the total objective function value using Eq. (1) and the same cost parameters as used for the small problem. We set the service level at 100 min for the Cordeau et al. [33]'s instance. We also used the maximum transportation time of 700 min instead of the maximum duration of a route as in the case of Cordeau et al. [33]. The service times of the delivery and installation vehicles are considered equal. The first part of the table shows the results of the heuristic: the CPU time in seconds, the transportation distances of the delivery and installation vehicles, and the total objective function. The second part of the table shows the results of the genetic algorithm. The values in the table are the same as for the heuristic, but the CPU time is set to 3 min for the instances. We examined the gap between the objective function values obtained by the heuristic and those obtained by the genetic algorithm. In group (a), the average gap is 3.7%, but the average gap is 1.5% in group (b). The improvement of the genetic algorithm over the heuristic in group (b) is much less than the improvement in group (a). The performance of the heuristic is much better with the larger time windows. In addition, it is more difficult for the genetic algorithm to construct routes within each depot because the ranges of the earliest and latest time are narrow in group (b).

Based on these results, the genetic algorithm yields a better solution for most of the instances. Although the total transportation distance increases, the total objective value decreases. It appears that the fixed costs for the vehicles decrease, and the total cost is reduced. This computational study demonstrates that the genetic algorithm outperforms the heuristic algorithm, but the developed heuristic algorithm can efficiently solve comparatively large instances.

6. Conclusions

The MDVRPTW is an important NP-hard problem that has inspired the development of numerous heuristic algorithms. In this paper, we have considered the problem of scheduling installation vehicles for electronics through synchronization with delivery service vehicles from multiple depots. We have assumed that there is a service level for installation vehicles, which implies that the arrival times of delivery and installation vehicles at the same customer must not exceed the maximum allowable time interval. We have proposed a mathematical model and efficient algorithms for the MDVRPTW with respect to delivery and installation vehicles and their dual time constraints. We have developed a heuristic algorithm and a genetic algorithm for the MDVRPTW, and conducted numerical experiments for the MDVRPTW. We have shown that the solutions achieved by a separate scheduling of delivery and installation service are better than those serviced by one vehicle when time windows are tighter and large service levels are allowed. The results of the computational experiments also show the applicability and effectiveness of the developed genetic algorithm. A real-life case study based on these meta-heuristic algorithms would make an interesting research problem because meta-heuristics effectively solve realistic and practical logistics problems.

Acknowledgments

The authors are grateful for the useful comments from an associate editor and two anonymous referees. This research was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2015R1A2A1A15053948).

References

[1] J. Cordeau, M. Maischberger, A parallel iterated tabu search heuristic for vehicle routing problems, Comput. Oper. Res. 39 (2012) 2033–2050.
[2] J.W. Escobar, R. Linfati, P. Toth, G.M. Baldoquin, A hybrid granular tabu search algorithm for the multi-depot vehicle routing problem, J. Heuristics 20 (2014) 483–509.
[3] Y. Shimizu, T. Sakaguchi, A hierarchical hybrid meta-heuristic approach to coping with large practical multi-depot VRP, Ind. Eng. Manag. Syst. 13 (2014) 163–171.
[4] A. Subramanian, E. Uchoa, L.S. Ochi, A hybrid algorithm for a class of vehicle routing problems, Comput. Oper. Res. 40 (2013) 2519–2531.
[5] T. Vidal, T.G. Crainic, M. Gendreau, C. Prins, Implicit depot assignments and rotations in vehicle routing heuristics, Eur. J. Oper. Res. 237 (2014) 15–28.
[6] M. Drexl, M. Schneider, A survey of the standard location-routing problem. Working Paper LPIS-03/2014, Logistics Planning and Information Systems, TU Darmstadt, Germany, 2014.
[7] M. Drexl, M. Schneider, A survey of variants and extensions of the location-routing problem, Eur. J. Oper. Res. 241 (2015) 283–308.
[8] C. Prodhon, C. Prins, A survey of recent research on location-routing problems, Eur. J. Oper. Res. 238 (2014) 1–17.
[9] C. Contardo, J. Cordeau, B. Gendron, A GRASP + ILP-based metaheuristic for the capacitated location-routing problem, J. Heuristics 20 (2014) 1–38.
[10] V.C. Hemmelmayr, J.-F. Cordeau, T.G. Crainic, An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics, Comput. Oper. Res. 39 (2012) 3215–3228.
[11] Z. Özyurt, D. Aksen, Solving the multi-depot location-routing problem with lagrangian relaxation, in: Baker E., Joseph A., Mehrotra A., Trick M. (Eds.), Extending the Horizons: Advances in Computing, Optimization, and Decision Technologies (ICS 2007 Conference), vol. 37, 2007, pp. 125–144.
[12] J.W. Escobar, R. Linfati, P. Toth, A two-phase hybrid heuristic algorithm for the capacitated location-routing problem, Comput. Oper. Res. 40 (2013) 70–79.
[13] J.W. Escobar, R. Linfati, M.G. Baldoquin, P. Toth, A granular variable tabu neighborhood search for the capacitated location-routing problem, Transp. Res. Part B 67 (2014) 344–356.
[14] J. Liu, V. Kachitvichyanukul, A pareto-based particle swarm optimization algorithm for multi-objective location routing problem, Int. J. Ind. Eng.: Theory Appl. Pract. 22 (2015) 314–329.
[15] C.J. Ting, C.H. Chen, A multiple ant colony optimization algorithm for the capacitated location routing problem, Int. J. Prod. Econ. 141 (2013) 34–44.
[16] I. Sariçiçek, Y. Akkuş, Unmanned aerial vehicle hub-location and routing for monitoring geographic borders, Appl. Math. Model. 39 (2015) 3939–3953.
[17] S. Mousavi, B. Vahdani, R. Tavakkoli-Moghaddam, H. Hashemi, Location of cross-docking centers and vehicle routing scheduling under uncertainty: a fuzzy possibilistic-stochastic programming model, Appl. Math. Model. 38 (2014) 2249–2264.
[18] B. Golden, A. Assad, L. Levy, F. Gheysens, The fleet size and mix vehicle routing problem, Comput. Oper. Res. 11 (1984) 49–66.
[19] S. Liu, A hybrid population heuristic for the heterogeneous vehicle routing problems, Transp. Res. Part E 54 (2013) 67–78.

[20] Z. Naji-Azimi, M. Salari, A complementary tool to enhance the effectiveness of existing methods for heterogeneous fixed fleet vehicle routing problem, Appl. Math. Model. 37 (2013) 4316–4324.
[21] S. Salhi, N. Wassan, M. Hajarat, The fleet size and mix vehicle routing problem with backhauls: formulation and set partitioning-based heuristics, Transp. Res. Part E 56 (2013) 22–35.
[22] I. Moon, J. Lee, J. Seong, Vehicle routing problem with time windows considering overtime and outsourcing vehicles, Expert Syst. Appl. 39 (2012) 13202–13213.
[23] J. Lee, K. Kim, An auction-based dispatching method for an electronic brokerage of truckload vehicles, Ind. Eng. Manag. Syst. 14 (2015) 32–43.
[24] T. Kim, H. Lee, Delivery management system using the clustering based multiple ant colony algorithm: korean home appliance delivery, Int. J. Ind. Eng.: Theory Appl. Pract. 21 (2014) 53–65.
[25] J. Sun, Synchronizing the schedule of delivery and service vehicle using endosymbiotic evolutionary algorithm, J. Korean Soc. Supply Chain Manag. 7 (2007) 79–90.
[26] K. Kim, J. Sun, S. Lee, A hierarchical approach to vehicle routing and scheduling with sequential services using the genetic algorithm, Int. J. Ind. Eng. 20 (2013) 99–113.
[27] I. Sim, T. Kim, J. Cha, H. Lee, A design and analysis of scheduling for the dual of appliances delivery and installation services, J. Korean Soc. Supply Chain Manag. 11 (2011) 41–53.
[28] J.K. Lenstra, A.H.G. Rinnooy Kan, Complexity of vehicle routing and scheduling problems, Networks 11 (1981) 221–227.
[29] G. Reinelt, The Traveling Salesman: Computational Solutions for TSP Applications, Springer, New York, 1994.
[30] M. Gen, R. Cheng, Genetic Algorithms and Engineering Design, Wiley, New York, 1997.
[31] M. Gen, F. Altiparmak, L. Lin, A genetic algorithm for two-stage transportation problem using priority-based encoding, OR Spectr. 28 (2006) 337–354.
[32] O. Bräysy, M. Gendreau, Vehicle routing problem with time windows, part II: metaheuristics, Transp. Sci. 39 (1) (2005) 119–139.
[33] J.F. Cordeau, G. Laporte, A. Mercier, A unified tabu search heuristic for vehicle routing problems with time windows, J. Oper. Res. Soc. 52 (2001) 928–936.