A time-dependent vehicle routing problem in multigraph with FIFO property

ARTICLE *in* JOURNAL OF MANUFACTURING SYSTEMS · APRIL 2015

Impact Factor: 1.68 · DOI: 10.1016/j.jmsy.2014.11.016

4 AUTHORS, INCLUDING:

Setak Mostafa

Khaje Nasir Toosi University of Technology

Hossein Karimi

University of Bojnord


ELSEVIER
Journal of
Manufacturing
SYSTEMS
SME
Society of
Manufacturing
Engineers

Journal of Manufacturing Systems 35 (2015) 37–45

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

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

Technical Paper

A time-dependent vehicle routing problem in multigraph with FIFO property

Mostafa Setak*, Majid Habibi, Hossein Karimi, Mostafa Abedzadeh

Department of Industrial Engineering, K.N. Toosi University of Technology, Tehran, Iran

A R T I C L E I N F O

Article history:
Received 9 December 2012
Received in revised form
24 November 2014
Accepted 24 November 2014

Keywords:
Time-dependent vehicle routing problem
Multigraph
FIFO property
Tabu search

A B S T R A C T

The classic vehicle routing problems are usually designed with one edge (or arc) between two nodes. However, sometimes, there are situations in which one node can be accessible from another with more than one edge. In this paper, a new extension of the time-dependent vehicle routing problem was studied. In this problem, the existence of more than one edge between the nodes was allowed. This property can be important in competitive markets in order to achieve better and faster product deliveries/services. So, the problem was modeled as the time-dependent vehicle routing problem in multigraph which provided the FIFO property. In addition, a heuristic tabu search (TS) algorithm was proposed to solve the problem, in which neighborhood search was randomly selected among swap or reverse exchanges. Finally, computational results of TS algorithm and exact solution method were represented over forty instances. It was shown that TS was more effective than exact solution method in terms of the quality of results and computational time.

1. Introduction

Nowadays, supply chain and logistics management play a significant role in the reduction of costs. In the meantime, transportation is one of the most crucial processes in the supply chain network. Transportation costs significantly contribute to production and operational costs [1]. In particular, vehicle routing problem (VRP) is one of the most important problems in transportation cost optimization. In recent years, different formats of VRP have attracted great attention of researchers in the literature [2].

VRPs can be classified into two major types: static and dynamic problems. In the static problems, parameters such as required time to cross route, demands, and costs are assumed static over time. This type is not an acceptable approximation when firms or people need results that are more exact. Therefore, another class of VRPs has been introduced which is called dynamic vehicle routing problem (DVRP) and solves some of the deficiencies in the static vehicle routing problem. For example, in the real world, demand of customers and travel time at an edge are not always fixed. Additionally, due to the fractional probability, the number of available vehicles is not exactly obvious. Different types of DVRP address these challenges.

For instance, time-dependent vehicle routing problem (TDVRP) is one kind of VRP, in which the travel time in the network depends on time. Furthermore, the travel time between two certain locations is changed by different factors such as weather condition, accident, and traffic jam. The objective of this problem is to minimize the total travel time and routing cost.

In time-dependent problems, the original work was done by Bowman [3], in which time dependency was presented in scheduling problems. Then Picard and Queyranne [4] and Lucena [5] have used time-dependent costs in traveling salesman problem. After that, Malandraki and Daskin [6] introduced it to the VRP and assigned a mixed integer linear programming solution to TDVRP with the time windows, in which travel time was considered as a step function. Thereafter, other people began to explore TDVRP. For instance, Park [7] formulated this problem as a multi-objective problem. Ichoua et al. [8] and Donati et al. [9] have applied tabu search (TS) and ant colony optimization (ACO) to solve TDVRP. Ichoua et al. [8] wrote the first paper that introduced an FIFO property into TDVRP. Haghani and Jung [10] considered travel times as a continuous function over time and suggested a genetic algorithm (GA) to solve the model. Rizzoli et al. [11] studied the application of ACO in various forms of VRP, including TDVRP. Ibaraki et al. [12] and Hashimoto et al. [13] have designed a couple of models based on predetermined feasible paths and applied the approaches using iterated local search algorithms to solve these models. Woensel et al. [1] considered traffic congestion factor in TDVRP and used queuing theory to deal with this type of models. Wang and Wang

* Corresponding author. Tel.: +98 2184063373.
E-mail address: setak@kntu.ac.ir (M. Setak).

http://dx.doi.org/10.1016/j.jmsy.2014.11.016

[14] utilized a novel two-phase heuristic to solve TDVRP with backhauls. Soler et al. [15] transformed TDVRPTW into a capacitated vehicle routing problem at several stages. Kuo et al. [16] employed an optimization method based on TS for commodity assignment and VRP. Androutsopoulos and Zagrafos [17] suggested a heuristic which used dynamic programming for multicriteria TDVRP. Kuo [18] investigated TDVRP to minimize fuel consumption and used simulation annealing (SA) to solve it. Kok et al. [19] represented a heuristic based on dynamic programming for TDVRP while considering driving hour regulations. Figliozzi [20] studied the impact of traffic congestion on the vehicle characteristics and travel costs. Balseiro et al. [21] provided a method using ACO to solve TDVRP. Flamini et al. [22] represented methods for calculating the value of information driven from intelligent transportation systems (ITS) in the urban freight distribution of perishable goods. Androutsopoulos and Zagrafos [23] solved a two-objective TDVRP for hazardous materials with a heuristic method. Kok et al. [24] evaluated the impact of congestion avoidance on TDVRP. Figliozzi [25] designed a new algorithm to solve time-dependent VRP that was used for problems with both soft and hard time windows. In order to highlight the contributions of this research, contents of the mentioned studies are summarized in Table 1.

Thus far, the possible existence of more than one edge between two specified nodes has not been considered in the TDVRP literature. The studied problems are limited to the existence of only one edge between two nodes. Although it is reasonable to assume that there is only one edge with minimum travel time between different points in suburban transportation, it is not very acceptable in large urban environments according to the complexities of urbanism and traffic restriction.

Urban areas usually have a complicated structure, which provides accessibility to different nodes by more than one edge. In this condition, traffic rules for edges (e.g. determining maximum allowed speed and vehicle traffic constraints) affect edge selection. Choosing suitable edges is important for distribution and transportation firms due to the reduction of costs. In this paper, the modeled problem was called time-dependent vehicle routing problem in multigraph (TDVRPM) which allowed for considering more than one edge between two points in TDVRP.

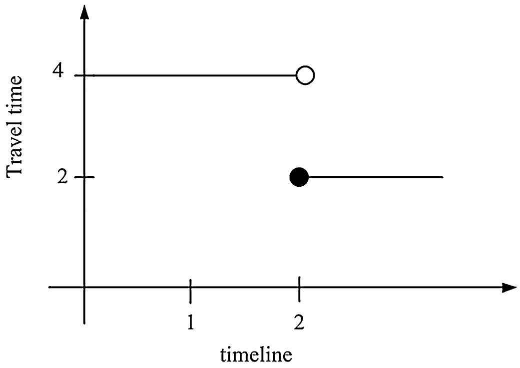

Fig. 1. Discrete travel time function.

In Section 2, TDVRPM is described and modeled using mixed integer linear programming. In Section 3, a tabu search algorithm is employed to solve this problem, since it is NP-hard. In Section 4, computational results are presented for some benchmark instances. Finally, the results are briefly summarized.

2. Problem description and modeling

2.1. FIFO property satisfaction

In this paper, the FIFO property was assumed. Since, in TDVRP, FIFO ensures that, when the vehicle "A" traverses the distance from node i to node j and a similar vehicle "B" starts to move from node i to node j after "A", then vehicle "B" will reach node j later than "A" [8].

The early papers related to TDVRP have a main shortcoming: they have modeled travel time as a discrete function of time [8]. For instance, Fig. 1 shows this function as it pertains to an edge

Table 1
TDVRP studies.

Papers	Solution method		Customer demand		Time window structure		Backhaul		Fleet of vehicle		Multigraph structure
	Heu.	Meta-Heu.	Deter.	Dyna.	Soft	Hard	Pic. & Del.	Line. & Back.	Sim.	Heter.	
Malandraki and Daskin [6]	√		√			√				√	
Park [7]	√		√						√		
Ichoua et al. [8]		√	√		√				√		
Donati et al. [9]		√	√		√				√		
Haghani and Jung [10]		√		√	√					√	
Rizzoli et al. [11]		√	√	√		√	√			√	
Ibaraki et al. [12]		√	√		√				√		
Hashimoto et al. [13]		√	√		√				√		
Woensel et al. [1]		√	√						√		
Wang and Wang [14]		√	√					√	√		
Soler et al. [15]	√		√			√			√		
Kuo et al. [16]		√	√						√		
Androutsopoulos and Zagrafos [17]	√								√		
Kuo [18]		√	√						√		
Kok et al. [19]	√		√			√			√		
Figliozzi [20]			√						√		
Balseiro et al. [21]		√	√			√			√		
Flamini et al. [22]		√	√		√				√		
Androutsopoulos and Zagrafos [23]	√		√			√			√		
Kok et al. [24]	√		√		√				√		
Figliozzi [25]	√		√		√	√			√		
This paper		√	√						√		√

Heu., heuristic; Meta-Heu., meta-heurist; Deter., deterministic; Dyna., dynamic; Pic. & Del., pickup and delivery services; Line. & Back., Linehual and backhaul services; Sim., similar; Heter., heterogeneous.

with the length of 1. So, if vehicle #1 leaves the origin node at time $t_1 = 1$, then it reaches the destination node at $t_2 = 5$. Nonetheless, when vehicle #2 leaves the origin node at time $t_3 = 2$ ($>t_1$), it will reach the destination at $t_4 = 4$ ($<t_2$), which means that, despite the fact that vehicle #2 leaves the origin node later than vehicle #1, it reaches the destination node earlier. Thus, the result does not provide the FIFO property.

Currently, researchers apply continuous travel time functions over the time horizon, instead of discrete travel time functions. In order to do so, travel speed function is applied, as shown in Fig. 2.

In this paper, a process was provided to transform a travel speed function into a continuous travel time function according to the represented approach by Ichoa et al. [8]. First, it is necessary to determine new time intervals for each edge: it is assumed that the head points of initial time intervals are $T = [T_1, T_2, \ldots, T_{U_{m_{ij}}}]$. In fact, this vector shows that there are U initial intervals for the mth edge between nodes i and j. Consequently, new time interval for each nodes i, j, and edge m is calculated by Eq. (1). In this equation, $v^{u}_{ijm_{ij}}$ is the travel speed at the mth edge between nodes i and j for the uth related time interval and d_{ijm} is the length of the mth edge between nodes i and j.

$$\begin{aligned} Tnew_{ijm_{ij}} &= [Tnew^{1}_{ijm_{ij}}, Tnew^{2}_{ijm_{ij}}, Tnew^{3}_{ijm_{ij}}, \ldots, Tnew^{H_{m_{ij}}-1}_{ijm_{ij}}, Tnew^{H_{m_{ij}}}_{ijm_{ij}}] \\ &= [T_1, (T_2 - T_1) - \frac{d_{ijm}}{v^{1}_{ijm_{ij}}}, T_2 \ldots, (T_{U_{m_{ij}}} - T_{U_{m_{ij}}-1}) - \frac{d_{ijm}}{v^{U-1}_{ijm_{ij}}}, T_{U_{m_{ij}}}] \\ \text{if} \quad & Tnew^{h_{m_{ij}}}_{ijm_{ij}} < Tnew^{h_{m_{ij}}-1}_{ijm_{ij}} \quad \text{then} \quad Tnew^{h_{m_{ij}}}_{ijm_{ij}} = Tnew^{h_{m_{ij}}-1}_{ijm_{ij}} \end{aligned} \tag{1}$$

Tnew vector length (H) is calculated based on the number of T vector members (U) using Eq. (2).

$$H = 2(U-1) + 1 \tag{2}$$

where U is the number of time intervals in the time speed function and H is the number of time intervals in its travel time function. Next, to compute travel time, two coefficients must be calculated using Eqs. (5) and (6):

$$x_{ijm_{ij}} = \left[\frac{d_{ijm_{ij}}}{v^{1}_{ijm_{ij}}}, \frac{d_{ijm_{ij}}}{v^{2}_{ijm_{ij}}}, \ldots, \frac{d_{ijm_{ij}}}{v^{U}_{ijm_{ij}}}\right] \tag{3}$$

$$y_{ijm_{ij}} = [x_{ijm_{ij(1)}}, x_{ijm_{ij(1)}}, x_{ijm_{ij(2)}}, x_{ijm_{ij(2)}}, \ldots, x_{ijm_{ij(U-1)}}, x_{ijm_{ij(U-1)}}, x_{ijm_{ij(U)}}] \tag{4}$$

$$b_{ijm_{ij}} = \left[\frac{y_{ijm_{ij(2)}} - y_{ijm_{ij(1)}}}{Tnew_{ijm_{ij(2)}} - Tnew_{ijm_{ij(1)}}}, \frac{y_{ijm_{ij(3)}} - y_{ijm_{ij(2)}}}{Tnew_{ijm_{ij(3)}} - Tnew_{ijm_{ij(2)}}}, \ldots, \frac{y_{ijm_{ij(H_{m_{ij}})}} - y_{ijm_{ij(H_{m_{ij}}-1)}}}{Tnew_{ijm_{ij(H_{m_{ij}})}} - Tnew_{ijm_{ij(H_{m_{ij}}-1)}}}\right] \tag{5}$$

$$a_{ijm_{ij}} = \left[y_{ijm_{ij(1)}} - b_{ijm_{ij}} \times Tnewy_{ijm_{ij(1)}}, \ldots, y_{ijm_{ij(H_{m_{ij}})}} - b_{jm_{ij(H_{m_{ij}})}} \times Tnew_{jm_{ij(H_{m_{ij}})}}\right] \tag{6}$$

Thus, with t_0 as the vehicle departure time from node i to node j throughout time interval $[Tnew_{ijm_{ij(h_{m_{ij}})}}, Tnew_{ijm_{ij(h_{m_{ij}}+1)}}]$, the travel time at the mth edge between nodes i and j is calculated as follows:

$$t = a^{h_{m_{ij}}}_{ijm_{ij}} + b^{h_{m_{ij}}}_{ijm_{ij}} t_0 \tag{7}$$

2.2. Problem description

Thus far, TDVRP papers have been based on this assumption: arc (i,j) is the only shortest distance between two locations i and j. Accordingly, the transportation network is based on a simple graph in which there is only one edge between two specific nodes. In static problems, this assumption is suitable with the consideration of fixed travel time. However, in the real world, particularly in the urban transportation networks, there is more than one edge between two locations, in which their travel time is different according to the daytime. Thus, choosing the edge for traveling depends on the specific time of the day. Multigraph can be used for this type of problems. In contrast with the simple graph, multigraph lets the model to establish more than one edge (or parallel edges) between two nodes. In the multigraph, the edge is shown by (i,j,m), in which m shows the mth parallel edge between nodes i and j. Fig. 3 demonstrates an example of the multigraph vs. simple graph.

In basic TDVRP, the problem is defined on a simple graph such as Fig. 3(a), in which the travel speed changes in each time interval with congestion. However, the network with the lowest travel time is fixed. In contrast, TDVRPM employs a multigraph-based network, in which the network with the lowest travel time edges is not fixed. In fact, this network usually changes in each time interval. For additional explanations, see Fig. 4 which implies a simple multigraph-based network. It shows the network of edge with the lowest travel time in three time intervals by bold continuous lines. Based on Fig. 4(a), in time interval number 1, travel time throughout the edge (i,j,1) is less than (i,j,2). Consequently, this edge is selected for the network that includes the lowest travel time edges. The same explanation is used to determine other edges of the network.

In this paper, the minimum cost of product deliveries was aimed to be determined in the transportation networks based on multigraph.

2.3. Modeling framework

The main purpose of this paper was to minimize the transportation cost of product deliveries to customers. Here, transportation cost was the result of total travel time from all available vehicles. There were the following assumptions in this model:

1. It is possible to get from a location to another with more than one edge.
2. The shortest edge between two locations is different according to the traffic situation at different times of the day.
3. All vehicles leave the depot simultaneously.
4. All vehicles return to the depot after finishing the product delivery to customers.
5. Demands of customers are given and fixed.
6. Capacity of each vehicle is given and fixed.
7. The first purpose of this model is to minimize the fleet number. Then, the model calculates minimum cost of travel time for each vehicle.

For modeling TDVRPM problem, first, its notations are described: Suppose $G = (V,E)$ is a complete graph, in which V and E are the set of nodes and edges, respectively. Each edge can be defined by a regular ternary as (i, j, m_{ij}). Here, i, j, and m_{ij} represent origin node (first node of link), destination node (second node of link), and the mth edge between those two nodes, respectively. In this model, H and $Tnew^{h_{m_{ij}}}_{ijm_{ij}}$ are the number of new time intervals and the head point of new time intervals, respectively. Other used notations are as follows:

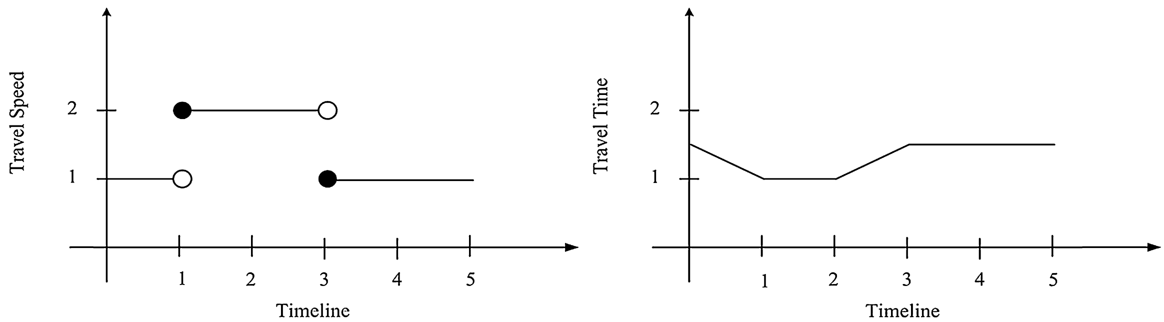

Fig. 2. A travel speed function and its travel time function.

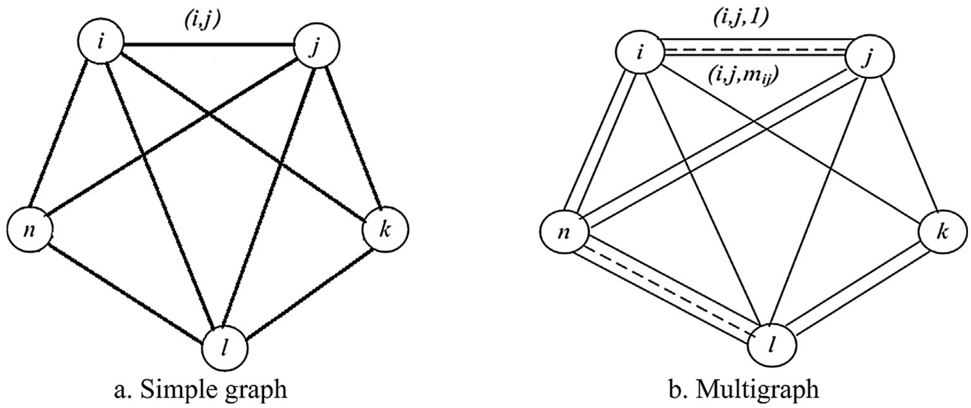

Fig. 3. Representation of multigraph vs. simple graph.

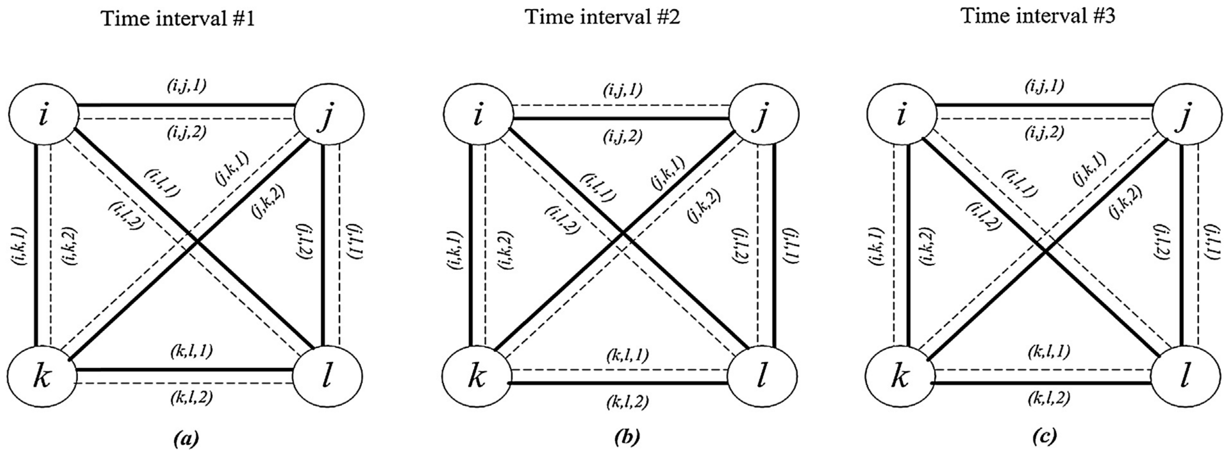

Fig. 4. Network of edges with less travel time in multigraph network transportation.

V_1 transportation variable cost per unit of travel time;
N number of customer nodes;
K number of available vehicles and number of depot copies;
E_1 fixed cost for vehicles that is very high. It can be considered as procurement or purchase cost of vehicles;
E_2, E_3 large numbers;
t travel start time for vehicles;
S_i service time for customer i;
Q capacity of vehicle;
M_{ij} number of edges between nodes i and j;
q_i demand of customer i;

Also, the following variables are calculated in the model: $x_{ijm_{ij}}^{h_{ij}} =$

$$\begin{cases} 1 & \text{if one of the vehicles moves through the} m\text{th edge from node} i \text{ to node} j \text{ in the} h\text{th time interval;} \\ 0 & \text{otherwise;} \end{cases}$$

t_i departure time from node i;
w_i amount (weight or volume) of freight carried by a vehicle while leaving node i;

Below, the formulation of TDVRPM model is presented:

$$Min \quad E_1 \sum_{i=1}^{N} \sum_{m=1}^{M_{ij}} \sum_{h=1}^{Hm_{ij}} \sum_{j=N+1}^{N+K} x_{ijm}^{h} + V_1 \sum_{i=N+1}^{N+K} t_i \tag{8}$$

s.t :

$$\sum_{i=0}^{N}\sum_{m=1}^{M_{ij}}\sum_{h=1}^{Hm_{ij}} x_{ijm}^{h} = 1 \quad \forall j \in \{1, \ldots, N\}, \quad i \neq j \tag{9}$$

$$\sum_{j=1}^{N+K}\sum_{m=1}^{M_{ij}}\sum_{h=1}^{Hm_{ij}} x_{ijm}^{h} = 1 \quad \forall i \in \{1, \ldots, N\}, \quad i \neq j \tag{10}$$

$$\sum_{j=1}^{N+K}\sum_{m=1}^{M_{ij}}\sum_{h=1}^{Hm_{ij}} x_{0jm}^{h} = \sum_{i=0}^{N}\sum_{j=N+1}^{N+K}\sum_{m=1}^{M_{ij}}\sum_{h=1}^{Hm_{ij}} x_{ijm}^{h} \tag{11}$$

$$t_0 = t \tag{12}$$

$$t_j - t_i - E_1 x_{ijm}^{h} \geq a_{ijm}^{h} + b_{ijm}^{h} t_i + s_j - E_2$$
$$\forall i \in \{0, \ldots, N\}; \quad \forall j \in \{1, \ldots, N+K\}; \quad \forall h \in \{1, \ldots, H_{m_{ij}}\};\ i \neq j; \quad \forall m \in \{1, \ldots, M_{ij}\} \tag{13}$$

$$t_i + E_3\, x_{ijm}^{h} \leq Tnew_{ijm}^{h} + E_3 \quad \begin{array}{l}\forall i \in \{0, \ldots, N\}; \quad \forall j \in \{1, \ldots N+K\};\\ \forall h \in \{1, \ldots, H_{m_{ij}}\};\ i \neq j; \quad \forall m \in \{1, \ldots, M_{ij}\}\end{array} \tag{14}$$

$$t_i - Tnew_{ijm}^{h-1} x_{ijm}^{h} \geq 0 \quad \begin{array}{l}\forall i \in \{0, \ldots, N\}; \quad \forall j \in \{1, \ldots N+K\};\\ \forall h \in \{1, \ldots, H_{m_{ij}}\};\ i \neq j; \quad \forall m \in \{1, \ldots, M_{ij}\}\end{array} \tag{15}$$

$$w_i - w_j - Q\sum_{m=1}^{M_{ij}}\sum_{h=1}^{Hm_{ij}} x_{ijm}^{h} \geq q_j - Q \quad \forall i \in \{0, \ldots, N\}; \quad \forall j \in \{1, \ldots, N+K\}; \tag{16}$$

$$w_0 \leq Q \tag{17}$$

$$x_{iim}^{h} = 0 \quad \forall i \in \{0, \ldots, N+K\}; \quad \forall m \in \{1, \ldots, M_{ij}\}; \quad \forall h \in \{1, \ldots, H_{m_{ij}}\} \tag{18}$$

$$x_{i0m}^{h} = 0 \quad \forall i \in \{0, \ldots, N+K\}; \quad \forall m \in \{1, \ldots, M_{ij}\}; \quad \forall h \in \{1, \ldots, H_{m_{ij}}\} \tag{19}$$

$$x_{ijm}^{h} = \{0, 1\} \quad \begin{array}{l}\forall i, j \in \{0, \ldots, N+K\};\\ \forall m \in \{1, \ldots, M_{ij}\}; \quad \forall h \in \{1, \ldots, H_{m_{ij}}\};\end{array} \tag{20}$$

$$t_i \geq 0 \quad \forall i \in \{0, \ldots, N+K\} \tag{21}$$

$$w_i \geq 0 \quad \forall i \in \{0, \ldots, N+K\} \tag{22}$$

The objective function (8) includes the fixed cost of used vehicles and cost of total travel time. In this formulation, there is a hierarchical function where the minimum vehicles are needed to be searched and then the minimum total routing cost is calculated based on the selected vehicles. In order to reduce the number of vehicles, E (fixed cost for vehicles) is considered as a large number. Constraints (9) and (10) ensure that each node is visited exactly once. Constraint (11) expresses that the number of departures from the depot to other nodes and the number of arrivals to the copies of the depot are the same. Constraints (12) and (13) show the departure time from depot and customer nodes, respectively. Constraints (14) and (15) determine the appropriate time interval according to the vehicle departure time from the origin node. Constraints (16) and (17) are related to the capacity of vehicles. Constraint (18) ensures that the origin and destination nodes of each vehicle are not similar. Constraint (19) states that none of the vehicles exit from the copied nodes of the depot. Finally, constraints (20)–(22) represent the type of variables.

3. Proposed tabu search algorithm

Since the solution time increases non-polynomially according to the enlargement of problem scale by solving the problem by exact solvers, an approach based on tabu search (TS) was proposed to solve the problem in less computational time. This algorithm is one of the three methods (other methods are genetic algorithm and particle swarm optimization) that have been proposed to solve the model. Tabu search obtains better solutions in less computational time in most cases than the other three. Thus, the proposed tabu search algorithm was explained and its results were represented in this paper.

One of the basic elements of TS is adaptive memory which makes a more flexible search behavior and attempts to avoid falling into earlier local optimal [26]. In the previous TDVRP papers, Ichoua et al. [8] employed a parallel tabu search algorithm to solve TDVRP with soft time windows. Woensel et al. [1] proposed a TS algorithm for TDVRP and concluded that the difference between the heuristic and exact solutions was acceptable. In addition, Kuo et al. [16] used an optimization method based on TS.

3.1. Feasibility of a solution

At the first stage, the required number of vehicles is calculated. A sequence of customer nodes is provided to create an initial solution. To ensure that the final sequence is feasible, it is necessary to check the vehicle capacity constraints. The approach shown in Fig. 5 can be used to obtain a feasible solution.

Based on this structure, first, an initial solution is provided. Then, customers are assigned to vehicles and the total assigned demand of each vehicle is determined. If the total assigned demand is less than the vehicle capacity, then this solution is approved. However, if the total demand is more than the capacity of the vehicle, the vehicle with maximum demand is chosen and a customer of this vehicle is randomly selected. Next, the vehicle with minimum demand is chosen and then the selected customer in the previous step is assigned to this vehicle. In the next step, it is re-checked whether the demands of vehicles are more than their capacities or not. These steps are continued to provide a feasible solution.

3.2. Objective function

Travel time is computed for each sequence. First, the new time intervals and coefficients used in Eq. (7) should be obtained. In order to determine the travel time between two nodes, the travel time through each available edge is determined based on the vehicle departure time of origin node and new time intervals that are calculated using (1). The related new time interval is determined according to the vehicle departure time of origin node for each available edge. Consequently, the required coefficients are determined. Now, by inserting the obtained value in (1), the edge with minimum travel time is selected for travel between two nodes, which is done for all pairs of nodes and travel routes.

3.3. Parameter tuning

The proposed tabu search had three main parameters which included length of tabu list, tabu tenure, and maximum number of iterations. $N(N-1)/2$ is used for the length of the tabu list. Here, N is the size of the problem (number of customer nodes). Taguchi analysis method [27] is applied to determine two other parameters. The algorithm is implemented on an example with five levels for tabu tenure ($N/12$, $N/6$, $N/3$, $N/2$, and $3N/2$) and five levels for the maximum number of iterations ($10N$, $12N$, $14N$, $16N$, and $18N$). Finally, $N/3$ and $14N$ are obtained for tabu tenure and maximum number of iterations, respectively.

3.4. Neighborhood search

Neighborhoods are checked $N(N-1)/2$ times in each iteration of the algorithm. This means that, all pairs of combinations are evaluated. The neighborhood search type is randomly selected among swap and reverse strategies (Fig. 6). The suggested neighborhoods search created diversification in exploration.

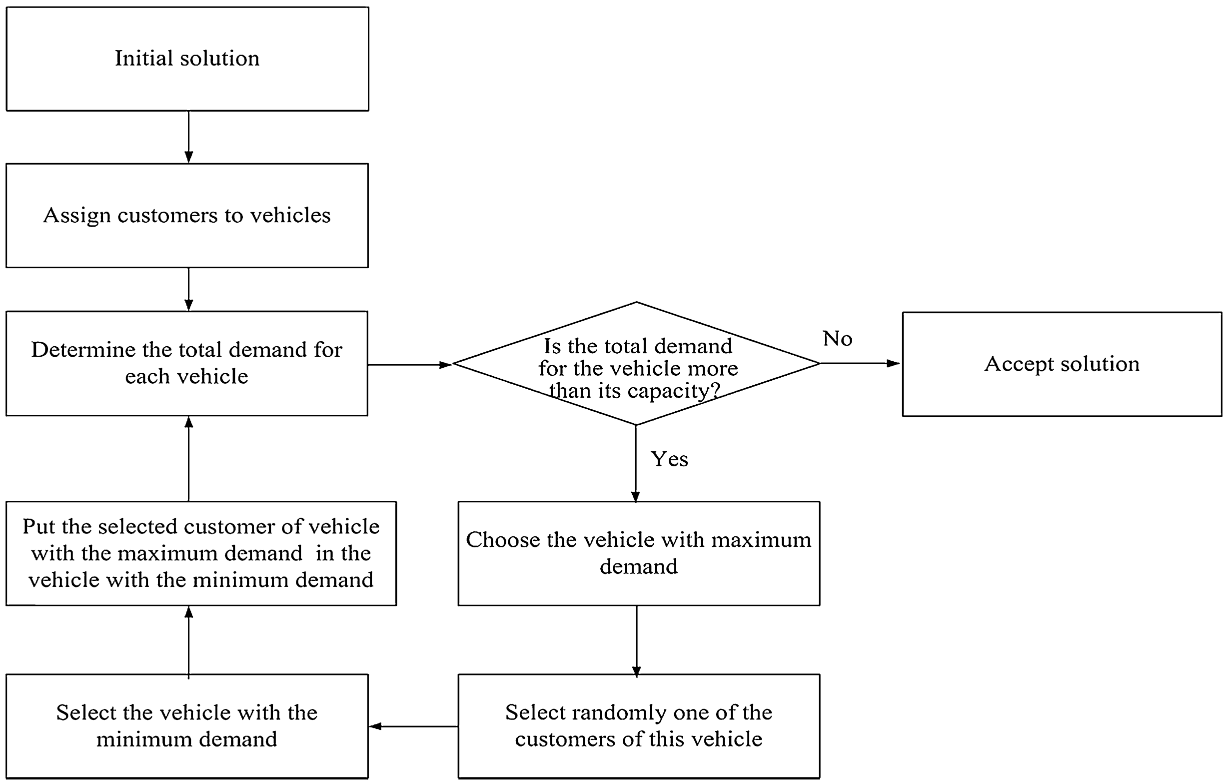

Fig. 5. A process for producing a feasible solution.

3.5. *The proposed tabu search structure*

The overall structure of the proposed TS algorithm is shown in Fig. 7. At first, a feasible initial solution is created. At this stage, this solution is the best solution and its cost is the minimum cost. Afterwards, the nodes are selected for the exchange. If the selected move is not in the tabu list, then this exchange is randomly applied based on swap or reverse strategy. The cost function of the obtained solution is calculated and compared with the previous best solution cost. The solution with minimum cost function is put in the best solution. Next, the tabu list, tabu tenure, and number of nodes are updated for the exchange. This process is continued until the maximum number of algorithm iterations is obtained.

4. Experimental study

4.1. *Generating problem instances*

It is the first time that this problem is raised in the TDVRP literature. In the light of this reason, forty benchmark instances were randomly generated to evaluate the proposed model and meta-heuristic. Length of the edge between two nodes and mean of travel speed for each edge in each time interval were obtained from the uniform [30,110] and [2,7] distributions, respectively. Two or 3 edges between each two node were used. The number of nodes in these sample problems was equal to the number of customers plus one (depot node). The number of time interval was set at 3 or 4 for each sample. Also, the demands of customers were randomly generated. Other parts of these instances are shown in Table 2.

4.2. *Computational results*

The sample problems were solved by CPLEX 12 solver in GAMS 23.5 and the proposed TS algorithm in Matlab 2010a on a system with CPU core i7 1.8 GHz and 6 GB of RAM. The results are represented in Table 3. In this table, the second column shows the capacity of each vehicle. Third column indicates the obtained number of vehicles for each sample problem. The next three columns are related to CPLEX results. These columns demonstrate the lower bound, upper bound, and time of branch and bound method in CPLEX implementation, respectively. Columns 7–9 in Table 3 display results of the proposed TS algorithm which are the best solution, mean of solution, and computational time, respectively. These results demonstrated that the exact solution was obtained

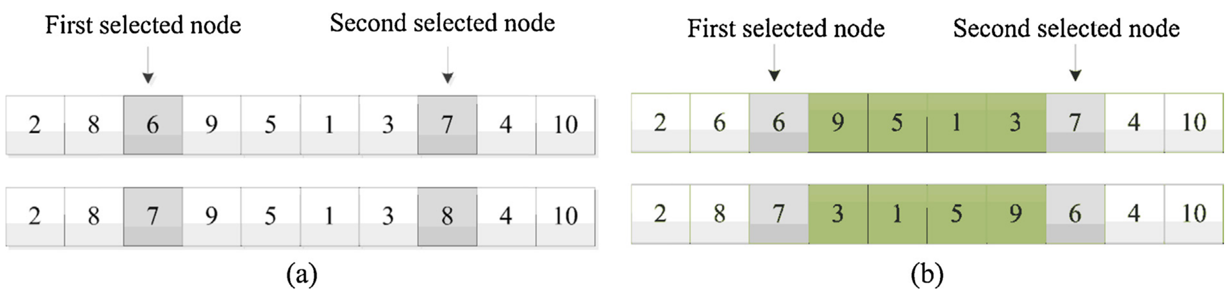

Fig. 6. Performance of swap strategy (a) and reverse strategy (b).

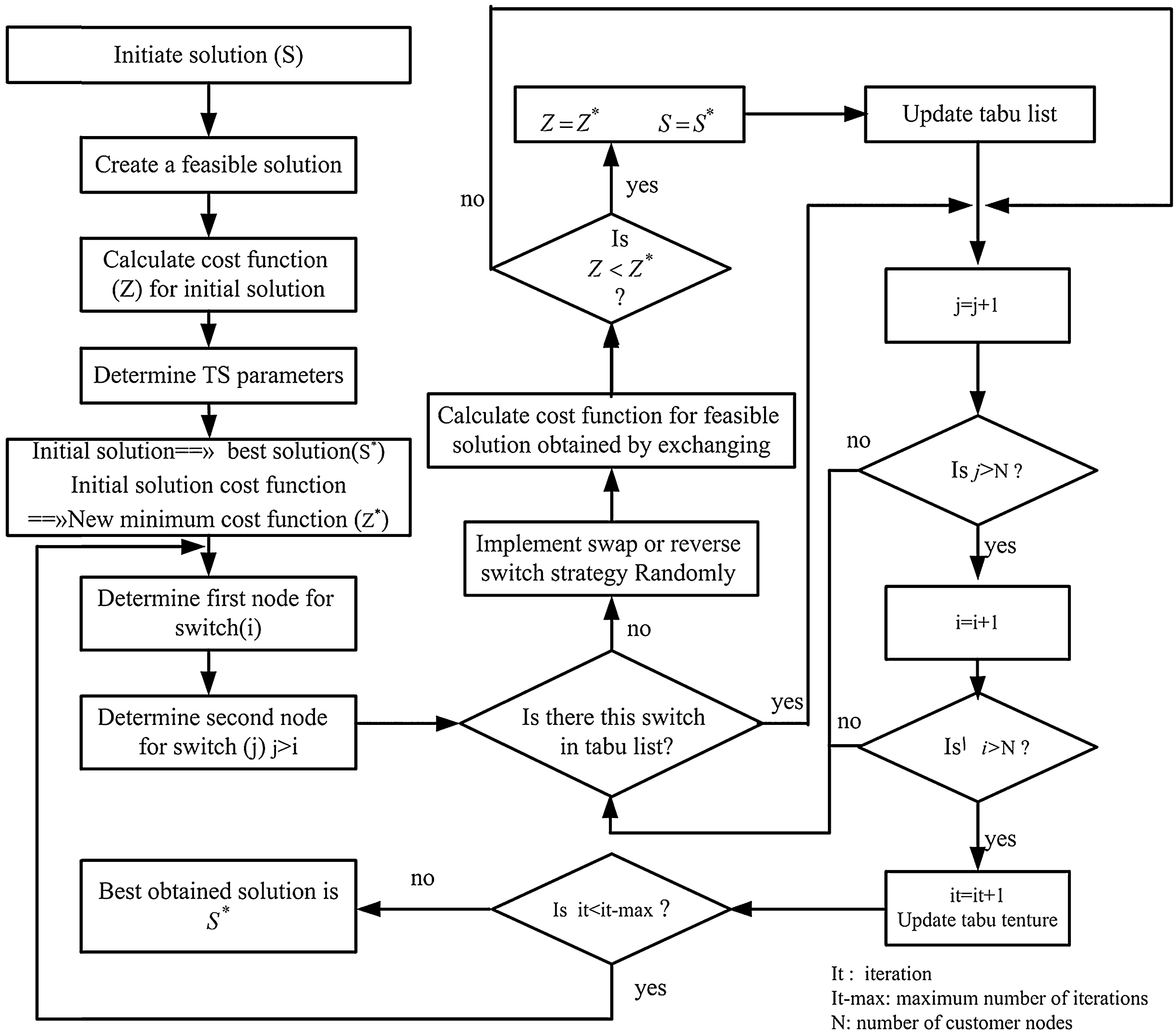

Fig. 7. The proposed Tabu search algorithm for TDVRPM.

for the problem with four customers by CPLEX solver. However, it is not possible to provide an exact solution in many cases of larger problems. Based on the literature, VRP problem is NP-hard in the strong sense [28]. Since TDVRPM is an extension of VRP, this problem is NP-hard as well; i.e. the computational time for solving TDVRPM increases for larger scale examples. Indeed, the quality of the solution decreases severely and computational time increases with the size growth of the problem. This means that the performance of CPLEX solver is not very satisfactory, even in small problems.

Results of the proposed algorithm are also shown in Table 3. In the problems in which CPLEX obtains an exact solution, this algorithm acquires the same result at all running times. Different results are observed with increasing the number of customers. However, the quality of the TS solution is more appropriate in many of the cases, in which CPLEX is not able to create the exact solution. This issue is also shown by RPD index. RPD index is calculated by (23).

$$\mathrm{RPD} = \frac{\mathrm{Method}_{\mathrm{sol}} - \mathrm{Best}_{\mathrm{sol}}}{\mathrm{Best}_{\mathrm{sol}}} \tag{23}$$

Table 2
Features of the sample problems.

Example number	Number of customers	Number of edges	Number of initial time interval	Example number	Number of customers	Number of edges	Number of initial time interval
1,2	4	2	3	21,22	6	3	4
3,4	4	2	4	23,24	6	3	3
5,6	4	3	4	25,26	7	2	3
7,8	4	3	3	27,28	7	2	4
9,10	5	2	3	29,30	7	3	4
11,12	5	2	4	31,32	7	3	3
13,14	5	3	4	33,34	8	2	3
15,16	5	3	3	35,36	8	2	4
17,18	6	2	3	37,38	8	3	4
19,20	6	2	4	39,40	8	3	3

Table 3
Results of CPLEX and the proposed TS.

Problem number	Capacity of vehicle	Number of vehicle	GAMS			TS			RPD for TS results (%)
			Lower bound	Upper bound	Time (s)	Best solution	Mean of solution	Time (s)	
1	100	2	**93.68**	**93.68**	65.46	**93.68**	**93.68**	0.60	0
2	200	1	**78.17**	**78.17**	55.14	**78.17**	**78.17**	0.54	0
3	100	2	**129.48**	**129.48**	250.33	**129.48**	**129.48**	0.61	0
4	200	1	**116.08**	**116.08**	73.18	**116.08**	**116.08**	0.58	0
5	100	2	**127.15**	**127.15**	3189.84	**127.15**	**127.15**	0.76	0
6	200	1	**115.07**	**115.07**	1595.33	**115.07**	**115.07**	0.77	0
7	100	2	**126.97**	**126.97**	442.52	**126.97**	**126.97**	0.80	0
8	200	1	**114.48**	**114.48**	180.53	**114.48**	**114.48**	0.71	0
9	150	2	**107.31**	**107.31**	11,482.06	**107.31**	**107.31**	1.83	0
10	250	1	**93.60**	**93.68**	616.36	**93.68**	**93.68**	1.68	0
11	150	2	–	**107.31**	17,057.06	**107.31**	**107.31**	1.83	0
12	250	1	**95.60**	**95.60**	7630.42	**95.60**	**95.60**	1.83	0
13	150	2	–	**104.67**	13,012.35	104.74	104.74	2.52	0.07
14	250	1	**92.57**	**92.57**	34,149.88	**92.57**	**92.57**	2.59	0
15	150	2	–	106.85	14,077.89	**104.67**	**104.67**	2.42	0
16	250	1	**92.55**	**92.55**	10,792.79	**92.55**	**92.55**	2.28	0
17	100	3	**170.33**	**170.33**	28,445.96	**170.33**	**170.33**	4.49	0
18	200	2	–	151.25	19,468.60	**151.25**	151.80	4.47	0
19	100	3	–	172.00	10,559.81	**170.33**	**170.33**	4.77	0
20	200	2	–	154.75	12,907.07	**151.71**	152.51	4.55	0
21	100	3	–	168.53	11,869.76	**168.58**	**168.58**	6.19	0
22	200	2	–	155.52	12,218.92	**150.71**	151.00	5.75	0
23	100	3	–	168.58	9787.80	**168.58**	**168.58**	6.24	0
24	200	2	–	**150.25**	10,787.04	150.31	150.84	5.71	0.04
25	150	3	–	197.57	19,402.17	**192.01**	**192.01**	9.35	0
26	200	2	–	184.17	19,361.45	**176.28**	176.32	8.89	0
27	150	3	–	197.32	15,069.86	**193.60**	**193.60**	9.33	0
28	200	2	–	182.94	11,933.79	**178.45**	178.58	9.12	0
29	150	3	–	204.12	8032.18	**196.64**	196.73	13.15	0
30	200	2	–	190.60	12,032.18	**177.22**	177.33	13.34	0
31	150	3	–	**191.27**	12,996.69	196.87	196.71	14.55	2.93
32	200	2	–	185.27	13,475.46	**176.41**	**176.41**	12.26	0
33	150	3	–	222.59	17,700.45	**213.75**	**213.75**	18.03	0
34	200	2	–	200.55	20,753.48	**195.08**	196.44	18.01	0
35	150	3	–	240.17	2077.39	**214.14**	214.70	19.33	0
36	200	2	–	218.93	18,478.97	**200.20**	200.86	18.24	0
37	150	3	–	235.10	13,001.62	**208.97**	209.87	27.08	0
38	200	2	–	211.42	15,154.80	**202.14**	202.24	28.31	0
39	150	3	–	213.09	15,989.39	**206.52**	207.5	27.78	0
40	200	2	–	201.83	10,964.56	192.54	192.70	26.52	0

Bold and underline: solutions of CPLEX and TS are equal. Bold: best solution.

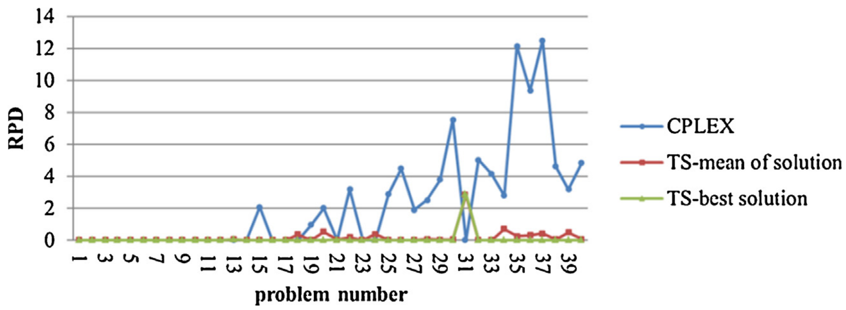

Fig. 8. RPD index of TS and CPLEX solutions.

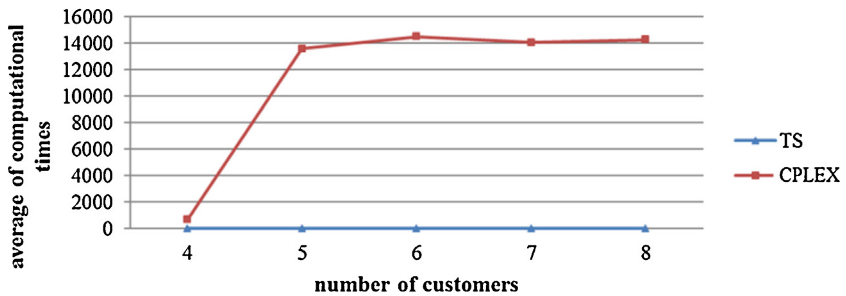

Fig. 9. Number of customers vs. computational time.

Fig. 8 shows the RPD index for TS and CPLEX solutions. This figure indicates that the obtained solutions of the TS are better than CPLEX upper bounds in most cases.

Also, the computational time for solving the proposed algorithm was less than the time needed for solving the problems using GAMS. Fig. 9 indicates comparison of the computational time for the two methods. In this figure, the average time for the solved problems is obtained based on the number of customers.

5. Summary and conclusion

This paper studied a vehicle routing problem in which more than one edge was possible between nodes. Such problems can be demonstrated in urban areas with different traffic conditions. The model could help transportation, distribution, and service firms to choose the best paths between the locations with tacking into every edge traffic situation at each time. This issue results in a better service in the shortest time. This model can be extended for important applications such as disaster and emergency services. For these applications, time is a vital object and the best edge and path selection is very important. To satisfy the FIFO property, the travel speed function was transformed into a continuous travel time function over time, which was done using a special procedure. The model was called TDVRPM and formulated as a mixed integer linear programming. Forty sample problems were randomly generated. Furthermore, in order to decrease the computational time, a meta-heuristic method was represented based on tabu search algorithm. Comparison of the obtained results from the CPLEX solver in GAMS V23.5 and the proposed algorithm confirmed the efficiency and effectiveness of the proposed TS.

In future works, other important actual features of VRPs, such as a heterogeneous fleet of vehicles, non-constant travel time functions (i.e. fuzzy, probabilistic, and uncertainty), and time windows, can be added to this model. Moreover, more efficient and effective meta-heuristics to solve large-scale problems can be employed in future.

References

[1] Woensel TV, Kerbache L, Peremans H, Vandale N. Vehicle routing with dynamic travel times: a queueing approach. Eur J Oper Res 2008;186:990–1007.
[2] Eksioglu B, Vural AV, Reisman A. The vehicle routing problem: a taxonomic review. Comput Ind Eng 2009;57:1472–83.
[3] Jung S [Ph.D. thesis] A genetic algorithm for the vehicle routing problem with time dependent travel time. University of Maryland; 2000.
[4] Picard JC, Queyranne M. The time-dependent travelling salesman problem and its application to the tardiness problem in one-machine scheduling. Oper Res 1978;26:87–110.
[5] Lucena A. Time-dependent travelling salesman problem: the deliveryman case. Networks 1990;20:753–63.
[6] Malandraki C, Daskin M. Time dependent vehicle routing problems: formulations, properties, and heuristic algorithms. Transp Sci 1992;26: 185–200.
[7] Park YB. A solution of the bicriteria vehicle scheduling problems with time and area-dependent travel speeds. Comput Ind Eng 2000;38:173–87.
[8] Ichoua S, Gendereau M, Potvin JY. Vehicle dispatching with time-dependent travel times. Eur J Oper Res 2003;144:379–96.
[9] Donati AV, Montemanni R, Gambardella LM, Rizzoli AE. Integration of a robust shortest path algorithm with a time dependent vehicle routing model and applications. In: Computational Intelligence for Measurement Systems and Applications. CIMSA'03. 2003 IEEE International Symposium on. 2003. p. 26–31.
[10] Haghani A, Jung S. A dynamic vehicle routing problem with time-dependent travel times. Comput Oper Res 2005;32:2959–86.
[11] Rizzoli AE, Montemanni R, Lucibello E, Gambardella LM. Ant colony optimization for real-world vehicle routing problems. Swarm Intell 2007;1: 135–51.
[12] Ibaraki T, Imahori S, Nonbe K, Sobue K, Uno T, Yagiura M. An iterated local search algorithm for the vehicle routing problem with convex time penalty functions. Discrete Appl Math 2008;156:2050–69.
[13] Hashimoto H, Yagiura M, Ibaraki T. An iterated local search algorithm for the time-dependent vehicle routing problem with time windows. Discrete Optim 2008;5:434–56.
[14] Wang Z, Wang Z. A novel two-phase heuristic method for vehicle routing problem with backhauls. Comput Math Appl 2009;57:1923–8.
[15] Soler D, Albiach J, Martinez E. A way to optimally solve a time-dependent vehicle routing problem with time windows. Oper Res Lett 2009;37:37–42.
[16] Kuo Y, Wang CC, Chuang PY. Optimizing goods assignment and the vehicle routing problem with time-dependent travel speeds. Comput Ind Eng 2009;57:1385–92.
[17] Androutsopoulos KN, Zografos KG. Solving the multi-criteria time-dependent routing and scheduling problem in a multimodal fixed scheduled network. Eur J Oper Res 2009;192:18–28.
[18] Kuo Y. Using simulated annealing to minimize fuel consumption for the time-dependent vehicle routing problem. Comput Ind Eng 2010;59:157–65.
[19] Kok AL, Hans EW, Schutta JMJ, Zijm WHM. A dynamic programming heuristic for vehicle routing with time-dependent travel times and required breaks. Flex Serv Manuf J 2010;22:83–108.
[20] Figliozzi MA. The impacts of congestion on commercial vehicle tour characteristics and costs. Transp Res E: Logist 2010;46:496–506.
[21] Balseiro SR, Loiseau I, Ramonet J. An Ant Colony algorithm hybridized with insertion heuristics for the time dependent vehicle routing problem with time windows. Comput Oper Res 2011;38:954–66.
[22] Flamini M, Nigro M, Pacciarelli D. Assessing the value of information for retail distribution of perishable goods. Eur Transp Res Rev 2011;3:103–12.
[23] Androutsopoulos KN, Zografos KG. A bi-objective time-dependent vehicle routing and scheduling problem for hazardous materials distribution. Eur J Transp Logist 2012;1:157–83.
[24] Kok AL, Hans EW, Schutta JMJ. Vehicle routing under time-dependent travel times: the impact of congestion avoidance. Comput Oper Res 2012;39:910–8.
[25] Figliozzi MA. The time dependent vehicle routing problem with time windows: benchmark problems, an efficient solution algorithm, and solution characteristics. Transp Res E: Logist 2012;48:616–36.
[26] Glover F, Kochenberger GA. Handbook of metahuristics. New York: Kluwer Academic Publishers; 2003.
[27] Wolf A, Henes D, Bogdanski S, Lutz ThE. Kramer, statistical analysis parameter variations using the Taguchi method. MUNA 2013;122:247–64.
[28] Toth P, Vigo D, editors. The vehicle routing problem. Philadelphia: Society for Industrial and Applied Mathematics; 2002.