Applied Soft Computing 14 (2014) 504–527

Contents lists available at ScienceDirect

Applied Soft Computing

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

A multi-objective dynamic vehicle routing problem with fuzzy time windows: Model, solution and application

Seyed Farid Ghannadpour [a,*], Simak Noori [a], Reza Tavakkoli-Moghaddam [b], Keivan Ghoseiri [c]

[a] Department of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran
[b] Department of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran
[c] Department of Civil and Environmental Engineering, University of Maryland, MD, USA

ARTICLE INFO

Article history:
Received 15 August 2011
Received in revised form 17 February 2013
Accepted 28 August 2013
Available online 16 October 2013

Keywords:
Dynamic vehicle routing problem
Multi-objective optimization
Fuzzy time windows
Satisfaction level
Genetic algorithm

ABSTRACT

In this paper, a multi-objective dynamic vehicle routing problem with fuzzy time windows (DVRPFTW) is presented. In this problem, unlike most of the work where all the data are known in advance, a set of real time requests arrives randomly over time and the dispatcher does not have any deterministic or probabilistic information on the location and size of them until they arrive. Moreover, this model involves routing vehicles according to customer-specific time windows, which are highly relevant to the customers' satisfaction level. This preference information of customers can be represented as a convex fuzzy number with respect to the satisfaction for a service time. This paper uses a direct interpretation of the DVRPFTW as a multi-objective problem where the total required fleet size, overall total traveling distance and waiting time imposed on vehicles are minimized and the overall customers' preferences for service is maximized. A solving strategy based on the genetic algorithm (GA) and three basic modules are proposed, in which the state of the system including information of vehicles and customers is checked in a management module each time. The strategy module tries to organize the information reported by the management module and construct an efficient structure for solving in the subsequent module. The performance of the proposed approach is evaluated in different steps on various test problems generalized from a set of static instances in the literature. In the first step, the performance of the proposed approach is checked in static conditions and then the other assumptions and developments are added gradually and changes are examined. The computational experiments on data sets illustrate the efficiency and effectiveness of the proposed approach.

1. Introduction

One of the main requirements of transportation management is to provide goods and/or service from a supply point to various geographically dispersed points with significant economic implications. One of the most important and widely studied combinatorial optimization problems, which continue to draw attention from researchers, is the vehicle routing problem with time windows (VRPTW). This problem is an extension of the vehicle routing problem (VRP). Because of its inherent complexities and usefulness in many real-world applications, it has become a well-known combinatorial optimization problem arising in transportation logistic. The VRPTW seeks to determine the optimal number of routes and the optimal sequence of customers (from a set of geographically dispersed locations that pose a daily demand for deliveries) visited by each vehicle, taking into account constraints imposed by the vehicle capacity, service times and time windows, and defined by the earliest and latest feasible delivery time.

The VRP also has several variants involving in different constraints; for instance. In this paper, the dynamic version of the VRP with hard time windows is considered. In this problem, customer orders for service are called over time in a given planning horizon. In this case, not all information related to the planning of the routes is known by the planner, in which the location, size, and time window of an order become known only after the order arrives. A sufficient number of homogeneous vehicles for serving requests are located at a single depot at the beginning of the planning horizon. Obviously, this type of problem is more challenging and sophisticated than the conventional static VRPTW. Hence, if the classes of the VRPTW and the dynamic VRPTW are denoted by P(VRPTW) and P(DVPRTW) respectively, then P(*VRPTW*) $\subseteq$ P(DVPRTW). As a conventional static VRPTW is NP-hard [1], it is not always possible to find optimal solutions to realistic-sized problems in a reasonable amount of computational time. This implies that the DVRPTW

* Corresponding author. Tel.: +98 9125883092; fax: +98 2123152824.
E-mail address: ghannadpour@rail.iust.ac.ir (S.F. Ghannadpour).

1568-4946/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2013.08.015

also belongs to the class of NP-hard problems. Since a static VRPTW should be solved each time, a new immediate request is received. Then, the DVRPTW can be formulated as a sequence of static VRPTW models. This idea was also used by Kilby et al. [2]. So due to the NP-hardness of the DVRPTW and its wide applicability in real-life situations, optimization techniques (i.e., meta-heuristics) are of major importance, which are capable of producing high quality solutions within a limited amount of time.

In practice, transportation often involves routing vehicles according to customer-specific time windows, which are highly relevant to the customers' satisfaction level. In these many realistic applications, the concept of classical time windows does not model the preference of customers very well. Even though customers provide a fixed time window for service, they really hope to be served at a desired time if possible. This preference information of customers can be represented as a convex fuzzy number with respect to the satisfaction for service time. So, this paper aims to study the mentioned dynamic VRPTW with the concept of fuzzy time windows and customers' satisfaction as a DVRPTW. Moreover, this model is interpreted as multi-objective optimization problem. Although such problems are frequently used to model real cases, they are often set up with the single objective of minimizing the cost of the solution, despite the fact that the majority of the problems encountered in industry, particularly in logistics, are multi objectives in nature. In real-life, for instance, there may be several costs associated with a single tour. However, when multiple objectives are identified, the different objectives frequently conflict. For this reason, adopting a multi-objective point of view can be advantageous and the optimization process needs to provide arrange of solutions that represent the trade-offs between the objectives, rather than a single solution. So, a direct interpretation of the DVRPFTW is used in this paper as a multi-objective problem in such a way that the total required fleet size, overall total traveling distance and waiting time imposed on vehicles are minimized and the overall customers' preferences for service is maximized.

The remainder of this paper is organized as follows. Section 2 reviews literatures of the VRPTW with fuzzy and dynamic assumptions and discusses recent developments in their solution approaches. Section 3 defines the multi-objective dynamic VRPTW. The structure of the solution technique is discussed in Section 4. Section 5 describes the computational experiments carried out to investigate the performance of the proposed method. Section 6 applies the presented multi-objective model on a case study, and finally Section 7 provides the concluding remarks.

2. Literature review

The literature of the VRPTW, due to its inherent complexities and usefulness in real life is rich in different solution approaches. Different types of heuristic methodologies, which seek approximate solutions in polynomial time instead of exact solutions with an intolerably high cost, are available in the literature of the VRPTW. In addition, it is shown that heuristics based on decomposition techniques (e.g., column generation and Lagrangian relaxation) may provide very good quality solutions when sufficient computational time is available [3]. Thus, various heuristic approaches have been developed, ranging from local search methods to methods based on mathematical programming decomposition techniques and meta-heuristics. Applying different meta-heuristics to solve the VRPTW can be extensively found in the literature and there are many papers used evolutionary algorithms for it [4–7]. In addition to describing the basic features of each method, experimental results for the benchmark test problems have been presented and analyzed. Eksioglu et al. [8] also presented a taxonomic framework for defining and integrating the domain of the extant VRP literature in terms that is operationally meaningful.

In using the fuzzy approach for VRPTW, Li et al. [9] solved the VRP with fuzzy demands using a hybrid differential evolution algorithm. In this study, a fuzzy chance-constrained program model was designed and a hybrid intelligent algorithm was proposed to solve the model. In this regard, Erbao and Mingyong [47] used the similar approach and designed fuzzy chance-constrained program model based on fuzzy credibility theory and applied it for the open VRP. Moreover, Tanga et al. [10] proposed and solved a static VRP with fuzzy time windows (VRPFTW) and crisp travel times. Service level issues associated with violation of time windows in a vehicle routing problem were described using fuzzy membership functions, and the concept of fuzzy time windows was proposed. In this paper as a classical point of view, if a customer is served during its predefined time window, the grade of service level is 1 (maximum level). Otherwise the grade of service level is decreased when this customer is served out of its time window. So, to deal with the issues pertaining to the violation of time windows, the concept of "fuzzy soft time windows" is proposed. So the service level cannot be described by only two states (0 or 1). In this paper VRPFTW is formulated as a multi-objective programming problem with two objectives: minimizing the travel distance and maximizing the sum of service levels of all customers. In order to find a Pareto solution, a two-stage algorithm was developed to decompose the original problem into two sub problems, and sequentially solves these objectives to optimization (travel distance first, service levels second).

This paper studies the dynamic version of the VRPTW. Because of its usefulness in realistic applications, it has become a well-known problem in network optimization. Therefore, many authors developed different solution approaches categorized in two major classes [11]. They differ mainly in the way with which the dynamism of the underlying problem is dealt. One class of methods, hereafter called a-priori optimization-based method, is based on probabilistic information on future events. This method tries to incorporate probabilistic features of future events or forecasted future information into the static problem at each decision epoch. It requires advance information about future events and can be used for situations where at least some probabilistic information about future orders is known in advance. Within the vehicle routing context, a-priori based solutions mean that the planner determines one or more routes based on probabilistic information on future requests for service, customers demands, travel times, etc. The probabilistic traveling salesman problem (PTSP) and the probabilistic vehicle routing problem (PVRP) as well as the stochastic vehicle routing problem (SVRP) are examples of problems solved by using the a-priori optimization-based method. For more details, see [12–16].

The other class of methods, hereafter called the real-time optimization method, plans the routes solely based on known information without looking into the uncertain future. At each decision epoch, a static problem consisting of known orders up to this point in time, which has not been covered, is solved. These methods do not need any advance information about future events and can be used for situations where future orders are difficult to predict. The previous methods have used stochastic/probabilistic information on future events to construct the routes. Within this setting, routes will be planned before the vehicle leaves the depot in the morning. The dynamic version of the well-known traveling salesman problem, dynamic traveling repairman problem, dynamic dial-a-ride problem, dynamic pickup and delivery problem [17] and dynamic vehicle routing problem can be considered in this class. In this area, Núñez et al. [18] proposed a hierarchical multi objective model based predictive control approach is presented for solving a dynamic pickup and delivery problem. In hierarchical multilayer systems, the system is divided into different functional layers, and the control structure consists of algorithms dealing with different components of the system, working at different temporal and spatial scales. In the bottom layer, the dispatcher (re)routes the vehicles when a new request appears, and minimizes user and operator costs. The dispatcher participates in the dynamic routing decisions by expressing his/her preferences in a progressively interactive way the results show that the potential benefits in the operator cost and in the quality of service perceived by the users. In the similar works, Cortés et al. [19] and Sáez et al. [20] considered a dynamic pickup and delivery problem and proposed hybrid adaptive predictive control for it. Cortés et al. [19] developed a formal adaptive predictive control framework to model the DPDP, and they also developed a particle swarm optimization (PSO) algorithm to efficiently solve it. In the similar way, Sáez et al. [20] considered future demand and prediction of expected waiting and travel times experienced by customers and proposed the use of genetic algorithms (GA) that provide near-optimal solutions for the three, two and one-step ahead problems. In the case of dynamic VRPTW, Gendreau et al. [21] described a dynamic routing problem with time windows motivated by a courier service application. The customers appear in real-time and must be served within a given soft time window. The TS method used in their work was originally designed for the static version of this problem and was therefore modified in order to deal with a dynamic version. Moreover, Chen and Xu [22] studied a version of the dynamic VRP with hard time windows, in which a set of customer orders arrives randomly over time to be picked up within their time windows. The dispatcher does not have any deterministic or probabilistic information on the location and size of a customer order until it arrives. This paper proposed column-generation-based dynamic approach for the problem. This approach generates single-vehicle trips over time in a real-time fashion by utilizing existing columns, and solves at each decision epoch a set-partitioning-type formulation of the static problem consisting of the columns generated up to this time point. In the same concept, Khouadjia et al. [23] proposed a particle swarm optimization and variable neighborhood search paradigms for dynamic VRP when new orders arrive when the working day plan is in progress. In this paper, the behavior of proposed methods is deeply analyzed on the set of benchmarks.

Du et al. [24] considered the dynamic VRP as a major part of delivery system of the business-to-customer environment. An efficient simulator was proposed in this paper that collects the necessary information of system and takes the order through the internet in an online B2C environment. The results indicated the efficiency of proposed approach. Lorini et al. [25] studied developments in mobile communication technologies, which are a strong motivation for the study of dynamic vehicle routing and scheduling problems. They developed a problem-solving approach for a VRP with dynamic requests and dynamic travel times and extended to account for more sophisticated communication means between the drivers and the central dispatch office. They focused on the second class of the Solomon's VRPTW benchmark problem instances [26] and the results demonstrated the benefits of this extension. Jemai and Mellouli [27] proposed a neural-tabu search heuristic for the real time VRP, which is a dynamic routing problem and requests are generated dynamically during the operation horizon without any previous knowledge. They developed an approach composed by two phases; the first part included of learning and reproducing the previous routing decisions using a feed forward neural network with a particular structure. The second phase was based on TS taking its initial solution from the assignment provided by the neural module. Haghani and Jung [28] studied a dynamic pick-up or delivery VRPTW and time-dependent travel times. They considered multiple vehicles with different capacities, real-time service requests, and real-time variations in travel times between demand nodes and proposed an efficient solution based on GA. The performance of their algorithm was evaluated by comparing its results with exact solutions and lower bounds for randomly generated test problems. Other very good papers in a dynamic routing problem as view of techniques and applications can be found in [29–33].

Eventually, this paper aims to propose an efficient real-time optimization method based on known information without looking into the uncertain future and model the multi-objective dynamic VRPFTW as a sequence of multi-objective static VRPFTW models. Not many studies can be found in the literature on multi-objective VRPTW. In the multi-objective area, Tan et al. [6] and Ombuki et al. [5] proposed a hybrid multi-objective evolutionary algorithm (MOEA) that incorporates various heuristics for local exploitation in the evolutionary search and the concept of Pareto's optimality for solving the multi-objective VRPTW. In the similar way, Tan et al. [34] proposed a similar approach for vehicle routing problem with stochastic demand with minimum travel distance, driver remuneration, and number of vehicles, subject to a number of constraints such as time windows and vehicle capacity. This paper presented a multiobjective evolutionary algorithm that is capable of finding useful tradeoff solutions for this model. Ghoseiri and Ghannadpour [7] used a direct interpretation of the VRPTW as a multi-objective problem where both the total required fleet size and total traveling distance are minimized. They used a goal programming approach for the formulation of the problem and an adapted efficient GA to solve the problem. They also used a similar model for formulation of a locomotive assignment problem and solved the model [35]. Gambardella et al. [36] studied a type of multi-objective implementation of the VRPTW by minimizing a hierarchical objective function, where the first objective was to minimize the number of vehicles and the second one was to minimize the total travel time. With the exception of the study of Ghoseiri and Ghannadpour [7], the remaining multi-criteria studies mentioned above that tackled the Solomon's instances did not make their results available in a proper multi-objective manner. Instead, Ombuki et al. [5] reported their results for the solution with the smallest number of routes and for that with the shortest travel distance. Tan et al. [6] presented their results only for the solution with the shortest travel distance. Recently, Najera and Bullinaria [37] proposed and analyzed a novel MOEA, which incorporates methods for measuring the similarity of solutions, to solve the VRPTW. This method was applied to a standard benchmark problem set [26] and illustrated competitive results. Moreover, Chevrier et al. [38] proposed a hybrid evolutionary approach for multi objective dial-a-ride problem where, three objectives of minimizing the number of vehicles used, minimizing the journey durations and minimizing the delays are optimized. In this work, the comparison of three state-of-the-art evolutionary algorithms: the Non-dominated Sorting Genetic Algorithm II (NSGA-II), the Strength Pareto Evolutionary Algorithm 2 (SPEA-2) and the Indicator Based Evolutionary Algorithm (IBEA) are done. Results obtained on random and realistic problems are detailed to compare three state-of-the-art algorithms and discussed from an operational point of view

Not many studies can be found in the literature on dynamic multi-objective VRP. In this area, Jun et al. [39] proposed a hybrid multi-objective ant colony algorithm for dynamic VRP where two objectives as vehicle number and time cost were simultaneously considered. In this paper the inherent conflicts between the defined objectives were also analyzed. Other invaluable study in this area belongs to work of Tang and Hu [40]. The courier mail services by the concept of dynamic VRP was proposed in this paper in a multi-objective approach. Three objectives were considered: maximization of the number of serviced customers, minimization of customer waiting, and minimization of total travel time. A solution (simulation) framework, was proposed to tackle this multiobjective problem and based on the results the multi-objective could offer much more attractive solutions to decision makers than single-objective models. Other very good studies in multi-objective optimization in dynamic setting can be found in [41,42].

Eventually, among the number of solving methods explored in the literature, this paper concentrates on using evolutionary algorithms for solving the multi-objective dynamic vehicle routing problem with fuzzy time windows. These algorithms examined in greater depth have a natural approach for dealing with multi-objective problems, and they have been successful in many practical situations.

3. Multi-objective dynamic vehicle routing problem with fuzzy time windows

3.1. Dynamic VRPTW

The vehicle routing problem with time windows (VRPTW) tries to design the least cost routes by a fleet of homogeneous vehicles from a central depot to a set of geographically customers with various demands and time windows. All homogeneous fleet of vehicles are located at the depot and they must leave from and return to it. In some studies, the number of vehicle is fixed and it is not optimized. But in other studies, it is assumed that the number of vehicles is infinite at first and the minimum required number of vehicles will be found after solving the model. This paper uses this approach and tried to find the minimum number of required fleets. However in order to facilitate the model formulation, the maximum possible size of the fleet is denoted by *K*. Each customer is to be served exactly once by only one vehicle, and each vehicle has a limited capacity. Moreover, a service time is associated with each customer that represents the time required to serve each of them. This important routing and planning is applied on static or dynamic conditions. In the static version of the VRPTW, all information relevant to the routing process is assumed to be known before the beginning of the planning. The information, which is assumed to be relevant, includes all attributes of the customers, such as the geographical location of the customers, the on-site service time, and the demand and time window of each customer. Furthermore, the planner should know about the system information, for example the travel times of the vehicle among the customers.

In the dynamic condition, which is counterpart of the static VRPTW, all data related to the routing process are not known before planning and they can change during the planning horizon. In this paper, the customer requests are considered as the dynamic condition. So, the planner encounters with the information of the limited number of customers at the beginning of the planning. During the routing process, new requests can arrive into the system. Thus, the dynamic VRPTW is strongly related to the static VRPTW. The main difference is that new orders arrive when the working day has already started, dynamically changing the optimization problem. The DVRPTW can be consequently modeled as a sequence of the static VRPTW-like instances (see Section 3). In particular, each static VRPTW will contain all the customers known at that time, but not yet served. An example of the DVRPTW is shown in Fig. 1.

As shown in Fig. 1, some customer orders (i.e., black nodes) are known in advance and an initial route schedule is generated to serve these customers. These initial routes are *Route* (1): *Depot*–1–2–3–*Depot* and *Route* (2): *Depot*–4–5–6–7–*Depot*. As time elapses and when vehicle 1 is approaching customer 2 (after serving customer 1) and vehicle 2 is approaching customer 4, new customers/orders (i.e., white nodes) arrive, and these new addition requires the rescheduling of one or more of the routes. After the re-optimization stage, customer 8 is considered in the planning of vehicle 1 and for serving customers 9 and 10, an extra vehicle must be distributed. The new planning routes at this decision stage are as follows: *Route* (1): *Depot*–1–2–8–3–*Depot*, *Route* (2): *Depot*–4–5–6–7–*Depot* and *Route* (3): *Depot*–9– 10–*Depot*.

The most important data for the re-optimization stage are relevant to information of real time requests and dispatched vehicles. The required information of new customers is identified when they call in for services to a dispatching center. However, the information of vehicles is determined by the continual communication between vehicles and the depot, which is essential for feeding the most up-to-date information into the routing and scheduling plan. Fig. 1 illustrates the most important equipments for this connection, which includes positioning equipments, such as global positioning system (GPS), for determining the current position of vehicles and the communication equipment for passing information on between the depot and vehicles. According to this figure, the signals from GPS satellites can provide receivers to calculate the coordinates of the vehicles. These positioning data with any other information (e.g., the current vehicle status) should be transferred to a dispatching center by the drivers of vehicles at any given point in time via communication equipment between the vehicle and the dispatching center. This data transmission is illustrated in Fig. 1 by information flow (1) between a vehicle and a depot. In addition, when the dispatching center knows, in which state the vehicle and the driver are at any time, it will have to re-optimize the routing plan with new other information and call the all drivers to inform them about the changes in the current routes. This data transmission is illustrated in Fig. 1 by information flow (2) between a depot and a vehicle.

3.2. Dynamic VRP with fuzzy time windows

This section describes the use of fuzzy time windows in the dynamic VRPTW, which was described in the previous section, especially its assumptions and necessities. In the conventional VRPTW, any customer *i* must be served within a pre-defined time interval $[e_i, l_i]$, limited by an earliest arrival time (e_i) and latest arrival time (l_i). This paper uses the hard time windows, in which delay in service is not allowed to the customers. However, waiting time is permitted at no cost and if the vehicles arrive earlier than the earliest arrival time, they incur a waiting time until the start of servicing. So, the start time of service for each customer *i* is calculated by:

$$t_i = \begin{cases} e_i & \text{if } t_{i-1} + f_{i-1} + T_{i-1,i} \le e_i \\ t_{i-1} + f_{i-1} + T_{i-1,i} & \text{Otherwise} \end{cases} \tag{1}$$

where f_i is the service time for customer *i* and $T_{i-1,i}$ is the travel time between the customer $i-1$ and *i*. The arrival time to customer *i* (at_i) is $t_{i-1} + f_{i-1} + T_{i-1,i}$. Moreover, vehicles are also supposed to complete their individual routes within the total route time, on which is essentially the time window of the depot or horizon planning. It must be mentioned that, in this conventional approach, each customer has the same satisfaction for service during the time window and it is just desirable for it to receive service within specified time interval. But this paper uses the fuzzy time windows for considering the different satisfaction rates for service in time window. This preference information of customers can be naturally represented as a convex fuzzy number with respect to the satisfaction for service time. Fig. 2 shows the satisfaction rate of each customer for service in the conventional approach. This conventional time window cannot meet the need of the real life and it does not reflect the customers' satisfaction for receiving the services, as each customer may prefer to be served at a certain time within the time window. This figure also depicts the typical fuzzy time window that can reflect the customers' satisfaction for service for different kind of customers. Every customers can be assigned by the expert to one

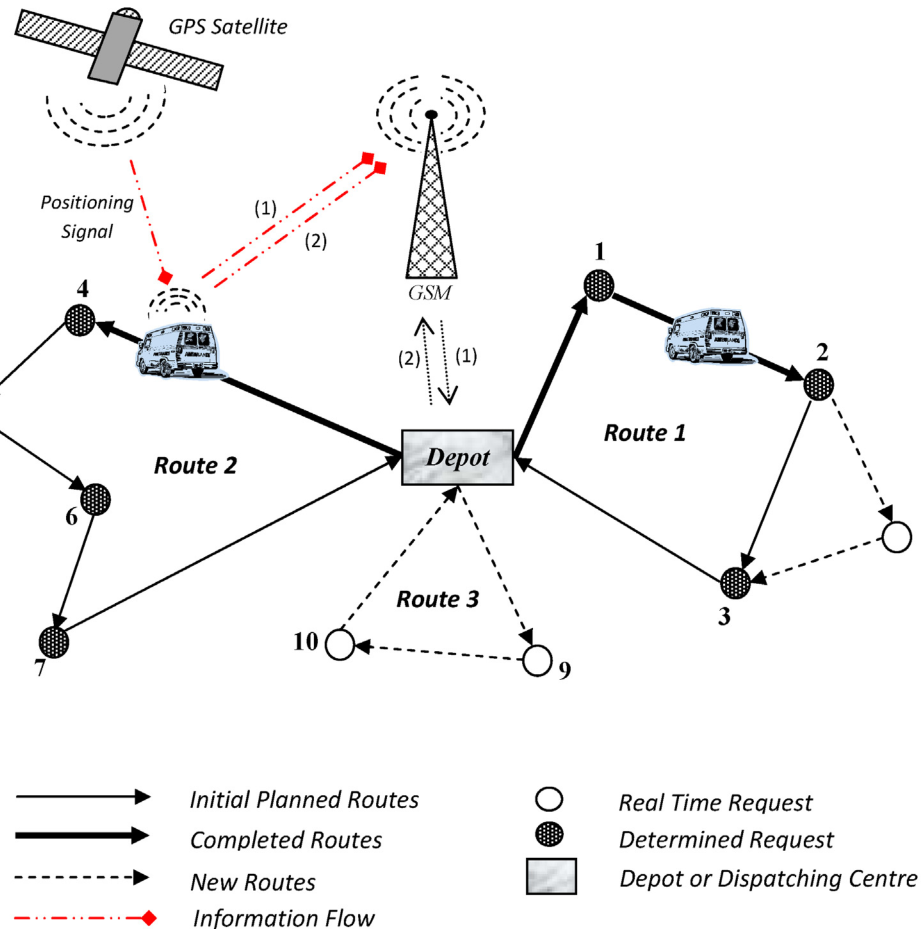

Fig. 1. Example of the DVRPTW.

of groups (e.g., very important, important, casual and unimportant) and predefined by the expert. The more important the customer, the tighter time window is. In an extreme case, the fuzzy time window is tighter than the classical counterpart. Thus, this figure shows an example of fuzzy time windows for casual and important customers. According to this figure, the proposed model is able to cope with the evaluation of deliveries for casual customers that violate the hard time widows.

According to Fig. 2, the classical time window is changed to the triple $[e_i, u_i, l_i]$, $[e_i', u_i, l_i']$ and $[e_i', u_i, l_i']$ for important, casual and very important customers, respectively. For example, if a customer is served at its desired time, the grade of its satisfaction is 1; otherwise, the grade of satisfaction gradually decreases along with the increase of difference between the arrival time of vehicle and desired times. The grade of satisfaction will be 0 (i.e., no satisfaction) if the arrival time falls in outside of the time interval. The membership function of an important customer i or $\mu_i(t_i)$, which represents the grade of satisfaction when the start of service time is t_i defined by:

$$\mu_i(t_i) = \begin{cases} 0 & t_i < e_i \\ \dfrac{t_i - e_i}{u_i - e_i} & e_i \le t_i \le u_i \\ \dfrac{l_i - t_i}{l_i - u_i} & u_i \le t_i \le l_i \\ 0 & t_i > l_i \end{cases} \tag{2}$$

According to this figure, the α-cut approach could also be used to satisfy the customers' needs. If very important customers occur during a period, the algorithm could include it by this approach. It means that these customers should be served with at least satisfaction rate of α%. By this approach, the time window of $[e_i, u_i, l_i]$ is changed to $[e_i'', u_i, l_i'']$ where, $e_i'' = \alpha(u_i - e_i) + e_i$ and $l_i'' = -\alpha(l_i - u_i) + l_i$.

It should be noted that the delivery process and the grade of satisfaction based on the start time of service for each customer i is easily computed when the travel time is crisp or static. Based on the mentioned assumptions, the travel time between customers are fixed and predefined by the planner before the beginning time of the working day. The travel times in reality and in urban areas fluctuate due to a variety of factors, such as accident, traffic conditions and weather conditions. This parameter can be modeled by the probability theory and based on the historical data as well as real times on urban roads. The probability theory can be used when the representative historical data are available. In contrast, fuzzy sets do not require such assumptions and can be used with little knowledge about the historical data. They express intuitive knowledge rather than exact uncertainty distribution. Moreover, the use of fuzzy sets is simple and naturally an extended version of arithmetic on real numbers. For these reasons, because fuzzy sets have been used to model tie windows already, the fuzzy numbers are used as the representation of these uncertain values. So this value is modeled as triangular fuzzy numbers $(\underline{x}, x^*, \bar{x})$ where value x^* (one-element core) represents knowledge of type "travel time is around x^*" and support $[\underline{x}, \bar{x}]$ models statement "travel time lays between $\underline{x}$ and $\bar{x}$". The uncertainty travel times naturally is propagated to arrival times (at_i) and start of service times (t_i) calculated according to formula (1) for crisp travel times. So when the fuzzy

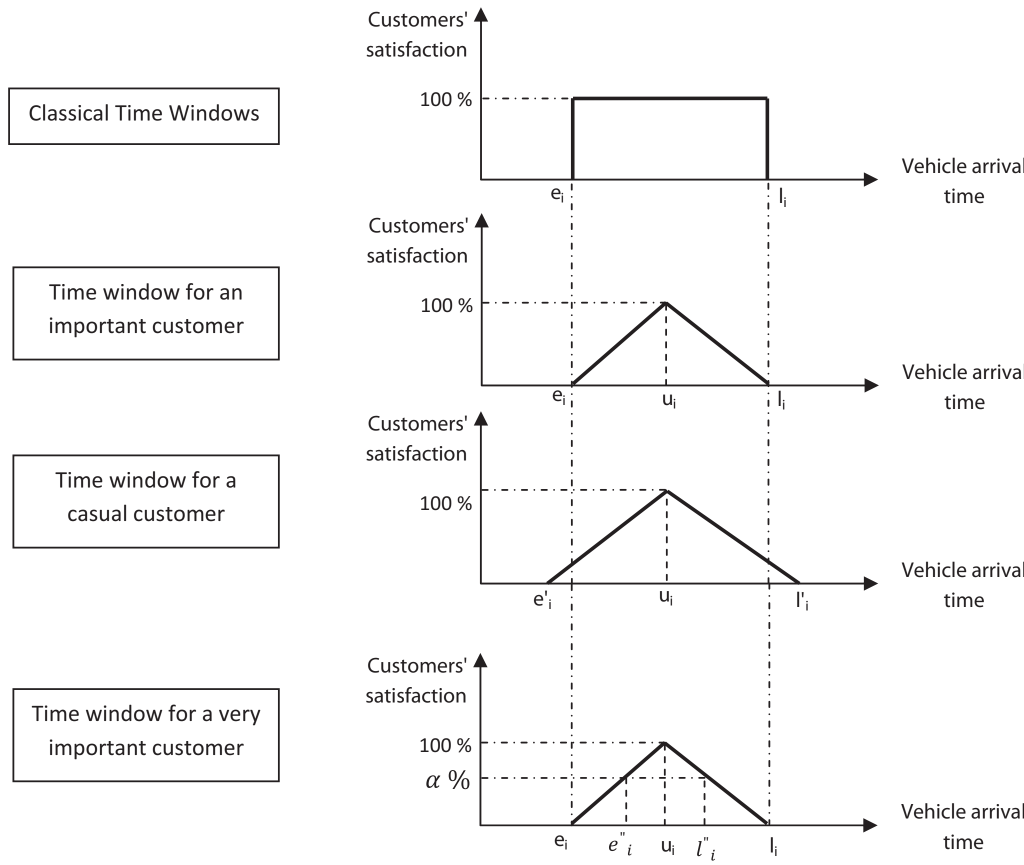

Fig. 2. Conventional and fuzzy time window for each customer.

travel time is considered, these parameters need to be modified to $\widetilde{t_i}$ and $\widetilde{at_i}$. Thus there is $\widetilde{at_i} = \widetilde{t_{i-1}} \oplus \widetilde{T_{i-1,i}}$, in which $\widetilde{T_{i-1,i}}$ is the fuzzy travel time between customers $i-1$ and i. Fig. 3 shows the service start time in customer i for crisp and fuzzy travel times.

$\mu_i(t_i)$ for the crisp travel time is computed as formulas (1) and (2) and the computational procedure of $\mu_i(\widetilde{t_i})$, which is the developed approach of [43], is discussed in the next sections.

3.3. Multi-objective dynamic VRP with fuzzy time window

As mentioned earlier, this paper uses a direct interpretation of the DVRPFTW as a multi-objective model where the distance traveled by the vehicles, the total number of vehicles used to serve the customers and the total waiting time of vehicles are minimized and the total satisfaction rates of customers is maximized. Moreover, the following constraints should be satisfied.

- Vehicle capacity constraint is observed.
- Time window constraint should be observed.
- Each customer is served exactly once.
- Each vehicle is starting from depot and ending at the depot.

Minimizing the total distance traveled and the total number of vehicles are the most common objectives used by other researchers alternatively. These objectives are as follows:

$$\text{Min} f_1 = \sum_{i \in N} \sum_{j \in N, j \neq i} \sum_{k \in K} D_{ij}.x_{ij}^k \tag{3}$$

$$\text{Min} f_2 = \sum_{j \in N, j \neq 0} \sum_{j \in N, j \neq 0} x_{0j}^k \tag{4}$$

where N and K are the set of customers and vehicles, respectively. For simplicity, the depot is denoted as customer 0. The travel distance between customers i and j is denoted as D_{ij}. Moreover, the decision variable x_{ij}^k is equal to 1 if vehicle k drives from customer i to customer j, and 0 otherwise. Moreover, this paper uses the fuzzy time windows that reflect the customers' satisfaction for receiving the services. Thus, the model tries to serve all the customers such that the summation of their satisfaction rates is maximized. When all customers are important, this objective is as follows:

$$\text{Max} f_3 = \sum_{i \in N} \mu_i(t_i) \tag{5}$$

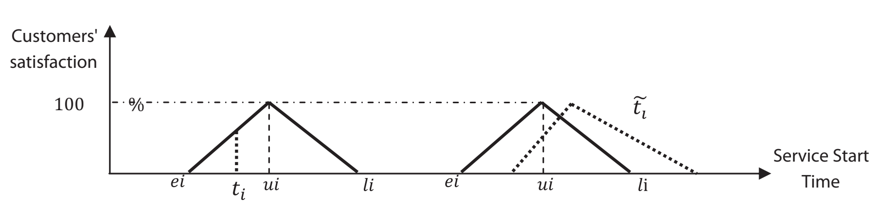

Fig. 3. Service start time for crisp (left hand side) and fuzzy travel times (right hand side).

where $\mu_i(t_i)$ is the satisfaction grade of customer i and it is calculated according to relation (2). However, when there are the customers with different importance classes, the above objective can be written by:

$$\text{Max}f_3 = \sum\nolimits_{i \in N} PR_i \times \mu_i(t_i) \tag{6}$$

where PR_i is the importance degree of customer i and $\mu_i(t_i)$ is the satisfaction level of customer i calculated according to relation (2) and based on its importance as shown in Fig. 2. When the fuzzy travel time is considered, the satisfaction level of customer i should be modified to $\mu_i(\widetilde{t_i})$ where its computational procedure and concepts are discussed in next sections.

It should be mentioned that the vehicles undergo a waiting time when their arrival time is before customers' earlier arrival time, in which e_i and e'_i are for important and casual customers, respectively. The summation of this waiting time, on which is undesirable for the decision maker and affects more vehicle and labor costs, for customers on routes should be minimized according to relation (7).

$$\text{Min}f_4 = \sum_{i \in N} w_i \tag{7}$$

where the waiting time imposed on each vehicle for customer i is calculated by $w_i = t_i - (t_{i-1} + f_{i-1} + T_{i-1,i})$ and it is $\widetilde{w_i} = \widetilde{t_i} - \widetilde{at_i}$ when the fuzzy travel time is considered.

Eventually, the multi-objective problem (MOP) studied in this paper is stated by:

$$(\text{MO} - \text{DVRPFTW}) = \begin{cases} F(\vec{x}) = \{f_1^-, f_2^-, f_3^+, f_4^-\} \\ s.t. \vec{x} \in D \end{cases} \tag{8}$$

where $\vec{x}$ is the decision variable vector, D is space, and $F(\vec{x})$ is the objective vector. The solution to a an MOP is the set of non-dominated solutions, called the Pareto set (PS), where dominance is defined as:

Definition. A solution $y = (y_1, y_2, \ldots, y_n)$ dominates (denoted $\prec$) a solution $z = (z_1, z_2, \ldots, z_n)$ if and only if $\forall i \in \{1 \ldots n\}, y_i \le z_i$ and $\exists i \in \{1 \ldots n\}, y_i < z_i$.

As mentioned earlier, this paper uses a posteriori approach, in which a set of potentially non-dominated solutions is first generated, and then the decision-maker chooses among those solutions. This procedure is described later completely.

4. Solution procedure

This section tries to design an efficient procedure and framework for solving the multi-objective VRP with fuzzy time windows as described in the previous section entirely. In this section, a brief description of the simulation environment is given, and then the routing heuristic method is discussed. The structure of the simulator is sketched in Fig. 4 consisting of three basic modules; namely, management module, strategy module and optimization module.

4.1. Management module

The management module tries to check the state of the system including information of vehicles and customers each time. The customers' information, which is assumed to be relevant, includes all attributes of the customers, such as the geographical location of the customers, the on-site service time, and the demand and time window of each customer. According to Fig. 4, initially at time $t = 0$, the pool of the customers' information may consist of all determined request customers who are remained from the previous working day and they should be served today. As time elapses and during the simulation ($t > 0$), this pool of request is enlarged if a new customer (i.e., real time request) for service is received and reduced whenever the service of a request or customer is ended. Thus, the management module tries to maintain and control the pool of the customers' information as $P_i = \{ct_i, x_i, y_i, q_i, f_i, e_i, u_i, l_i\}$ for each customer i and update them each time where, ct_i is the call time of customer i with the central dispatching center (i.e., depot) and based on this parameters, the rate of requests to happen is identified and the management module could be updated. It should be noted that these information like location, demands and etc. are not the variable and they are the fixed parameters and passed to the algorithm at each decision epoch as the necessary inputs.

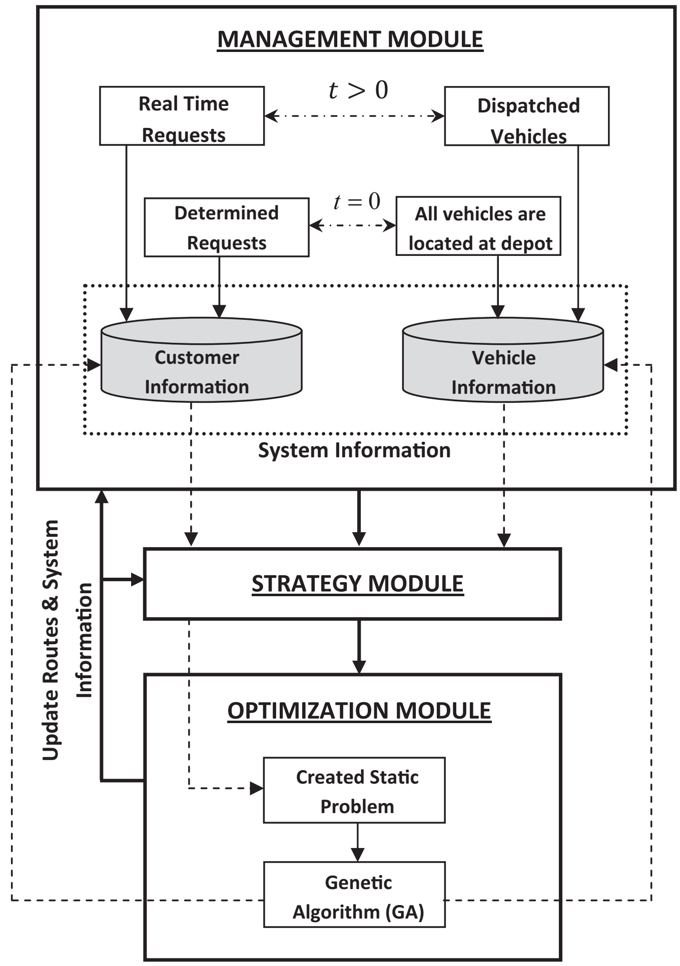

Fig. 4. Structure of the simulator.

All information relevant to customer (i) is unknown for the decision maker before ct_i and whenever his/her connection with depot is made, all required information for planning, such as geographical location (x_i, y_i), amount of demand (q_i), on-site service time (f_i) and time window (e_i, u_i, l_i) for service are determined. It should be noted that the planning horizon is considered as $[0, H]$ where; time 0 is the beginning time of working day. This means that the vehicles should be dispatched from the depot during this interval and after going through a number of customers, they end the constructed route at the depot before time H. Moreover, all customers should be served within this period. So, the customers, whose receiving time of services causes delay in service or whose call in for service at the time when the call center closes, should be rejected or carried over to the next working day as the determined requests. Hence, the inequality $0 \le ct_i < H$ should be considered each time. The procedure of setting the ct_i in our experiments and the call-in time for each customer is described in Section 5.

On the other hand, the management module should control the information of vehicles according to Fig. 4 and update them each time. Initially at time $t = 0$, all vehicles are located at depot and all required information is available. As time elapses ($t > 0$), the management module should control the state of dispatched vehicles and update their information continually for subsequent planning. The information, which should be checked by a dispatcher, includes the geographical locations, the residual capacity, the state of vehicle (i.e., driving, servicing, and waiting), and the like.

Eventually, the information of customers and vehicles are maintained in this module and a static model is solved by the proposed optimization module in each decision epoch. After that some customers may be served until this time, some new customers could be arrived to system up to this time and the vehicles enter a new state. So the management module should be updated after each decision epoch and it is enlarged if a new customer (i.e., dynamic request) for service is received and reduced whenever the service of a request or customer is ended. This procedure is continued until all customers are served. According to Fig. 4, the optimization module receives the necessary inputs from the management module.

4.2. Strategy module

According to Fig. 4, this module tries to organize the information reported by management module and construct an efficient structure for solving in the subsequent phase (i.e., optimization module). In this phase by using the main idea of the *batching strategy*, the planning of the routes is delayed until a certain amount of times has elapsed since the last planning was performed. Therefore, a working day is sub-divided into a number of discrete periods, on which each of these time slices represents a partial static VRP with fuzzy time windows, where the vehicles should serve all known customers without necessarily returning to the depot. Hence, this concept of a discrete time slice bounds the time given to each static model and provides an orderly way to serve new requests.

Therefore, the K discrete time periods are considered in each working day as $t_1, t_2, \ldots, t_k$ where, $t_1 = 0 < t_2 < \ldots < t_k < H$. Moreover, each time slice is defined as $t_i = (i - 1) \times \Delta$ where, $i = 1, \ldots, K$ and Δ ($\Delta > 0$) is the time interval between two consecutive steps. It should be noted that the Δ depends on the system strongly and it is assumed very short in the systems with a high degree of dynamism (e.g., emergency services and taxi cabs), which try to minimize the response time of service. The designed procedure is illustrated in Fig. 5. According to this figure, in each time step t_i, a certain amount of times (δ) as adjusting time should be spent to organize the required information and then construct the static model. This model in each time step t_i is solved in $[t_i + \delta, t_{i+1}]$ to find the solution S_{i+1}, which should be implemented in the next time slice and within $[t_{i+1}, t_{i+2}]$. Moreover, in time step t_i, solution S_i found in the previous time step ($[t_{i-1} + \delta, t_i]$) is implemented within $[t_i, t_{i+1}]$. According to this figure, solution S_i is a solution found during $[t_{i-1} + \delta, t_i]$ based on the available information of customers and vehicles adjusted in time step t_{i-1} and in $[t_{i-1}, t_{i-1} + \delta]$. Adjustment of required information of customers and vehicles to find each solution S_i in time step t_{i-1} is described later. At the beginning of the horizon, i.e., time 0, there is already a set of initial customers who would have missed servicing yesterday because of the time cut-off. So the first static problem created for the first time slice consists of all orders left over from the previous working day.

The required information of customers and vehicles for constructing the partial static model in time step t_i to find the solution S_{i+1} are as follows:

1- Information of vehicles: The set of vehicles information that should be considered in time slice t_i, is as $K_i = U_i \cup \{$infinitely many vehicles located at depot$\}$, where, U_i is the set of vehicles en routes until t_{i+1} with the information of their status, residual capacity and geographical location. These vehicles continue their routes from any committed customer it has just visited and when any vehicle has used all its capacity, it is sent back to the depot. Due to the solution S_{i+1} of the step t_i is implemented in next time slice, the dispatching time of new vehicles which should be dispatched from the depot is t_{i+1} and their horizon of planning is set as $[t_{i+1}, H]$.

2- Information of customers: It should be noted that the solution S_{i+1} is found based on the solution S_i. According to Fig. 5, some customers (denoted by the set of $\hat{N}_{i-1}$) may be served by the solution S_i until the time of t_{i+1} and some new customers could be arrived to system. According to Fig. 5, new requests are received within $[t_{i-1}, t_i]$ and they are denoted by the set of O_{i-1}. The new request that received after this time, are the set of O_i which they will come to system within $[t_i, t_{i+1}]$ and they would be considered in the next time step. So, the set of customers' information which is necessary in time step t_i to find the solution S_{i+1}, is denoted by N_i and it is calculated as Eq. (9):

$$N_i = \frac{N_{i-1}}{\hat{N}_{i-1}} \cup O_{i-1} \tag{9}$$

where N_{i-1} is the set of customers planned in previous time slice t_{i-1} and $\hat{N}_{i-1}$ is the set of customers which are served within $[t_i, t_{i+1}]$ by implementing of solution S_i. It is obviously $\hat{N}_{i-1} \subset N_{i-1}$ and the set of $\hat{N}_{i-1}$ should be omitted from N_{i-1}. Moreover, O_{i-1} is the set of new customers, which called in for service within $[t_{i-1}, t_i]$. So, new requests, which may be received by dispatcher during the previous period, should be considered in this time step.

Eventually, in each time slice t_i a partial static multi-objective VRP with fuzzy time windows is constructed by the mentioned information and solved by an efficient method in the optimization module.

It should be noted that, when the fuzzy travel times are considered, $\hat{N}_{i-1}$ could not be found easily. Because, the travel times are not crisp and identifying the customers which are served within $[t_i, t_{i+1}]$ at time t_i is not possible. In this case, the solution S_i is implemented within $[t_i, t_{i+1}]$ and the system should be checked again in time $t_{i+1} - \sigma$. So, $\hat{N}_{i-1}$ is easily determined based on the real travel times that the vehicles travelled. In this strategy, σ is the amount of remaining time to the end of each decision epoch. Eventually, in each time step t_i, the solution S_i found in the previous time step ($[t_{i-1} + \delta, t_i]$) is implemented within $[t_i, t_{i+1}]$ and the system is checked again when the σ units of time is remained until the time of t_{i+1}. So the certain amount of times (δ) as adjusting time should be spent to organize the required information and then construct the static model. This new model is solved in $[t_{i+1} - \sigma + \delta, t_{i+1}]$ to find the solution S_{i+1}. In this regard, the allowable time for solving each static model is changed from $[t_i + \delta, t_{i+1}]$ to $[t_{i+1} - \sigma + \delta, t_{i+1}]$. It is clear that, the time interval for solving each partial static model is more limited than the previous strategy and the quality of solution may be deteriorated compared with the case of crisp travel times.

4.3. Optimization module

The optimization module solves each static model of time slice t_i within $[t_i + \delta, t_{i+1}]$ and passes the new solution vector on to the management and strategy modules for updating and implementing. It should be noted that the new solution should be thought of as a tentative solution, which is subject to be changed, whenever a request is received. Naturally, changes can only be made to the unvisited parts of the routes.

In this paper, among the number of the solving methods explored in the literature, a genetic algorithm is examined in greater depth and used for solving the model in each re- optimization stage. Before deciding on using a GA for solving each static model, other search heuristics are considered and performed poorly [28]. Moreover, other meta-heuristics (e.g., TS and ACO) are

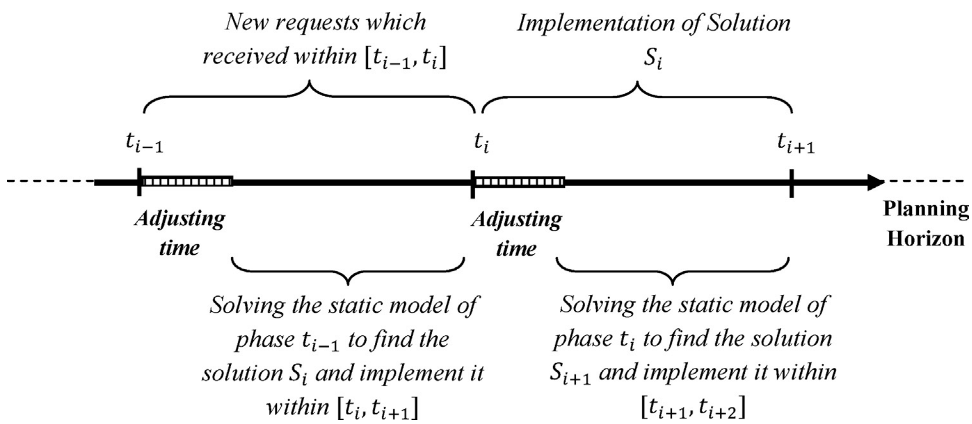

Fig. 5. Dynamic structure for the MO-DVRPFTW.

considered and based on the Ref. [28] providing a performance comparison of meta-heuristics, in which TS outperforms other meta-heuristics in terms of the solution quality, but requires a large number of parameters.

Eventually, the GA is adapted to solve the model because of its advantages in finding an easy way of the solution representation and because of its ease of implementation and ability of incorporation with the different operators that improve the solutions. It should be noted that this paper uses the modified GA, which was developed in our recent research [7] and is combined with *push forward insertion heuristic* to find an initial solution, λ-*interchange mechanism* to generate neighborhoods, *hill-climbing* to improve the quality of solutions and *Pareto ranking scheme* in order to find the non-dominated solutions. Moreover, the best cost -best route crossover (BCBRC) and sequenced based mutation (SBM) are used as recombination operators. Therefore, this paper tries to adapt this algorithm for the dynamic and fuzzy condition of the proposed model. The detailed descriptions of this algorithm and its other combined methods are in [7] completely. The supplementary operators and descriptions are given below.

The GA is a class of adaptive heuristics based on the drawing concept of evolution, known as "survival of the fitness". It starts with a set of chromosomes referred to as initial population and each of them represents a solution to the given problem. A solution of the model in time slice t_1, which no vehicles have been commissioned yet, is represented by an integer string of length n_0, where n_0 is the number of determined requests remained from the previous working day. This string of customer identifiers represents the sequence of deliveries that must cover vehicles during their routes. It is the typical representation of chromosomes in static VRPTW in the literature. So, this representation is used only in time slice t_1 (at the beginning of horizon) that in which, the decision maker encounters a static situation of customers who are remained from the previous working day and they should be served today. On the subsequence time slices t_i ($i > 1$), the solution of the previous steps are implemented and some new request may be received. Moreover some customers are also served and some vehicles are enroutes. So, the changes and re-optimization stage must only be made to the unvisited parts of the routes for planning the remained customers. A solution and its representation time slices t_i ($i > 1$), is a variable length chromosome representation as depicted in Fig. 6.

According to this figure, this chromosome representation gives the order, in which customers are to be visited and includes some information on previously visited customers. Two types of nodes are used in this representation, namely positive and negative nodes. The positive nodes represent the new customers that have been added to this day's schedule during $[t_{i-1}, t_i]$ and the customers, which came to system in the previous planning and not yet served until now. In Fig. 6 the positive nodes of genotype solution and the white nodes of phenotype solution illustrate the customers that not yet served up to this time. The negative nodes represent the group of clustered customers that have already been visited by the dispatched vehicles during the previous time slices. So these negative nodes are the indices of dispatched vehicles as a place holder and include the information of their partial routes and previously visited customers. For example, the negative integer of -1 in the genotype solution corresponds the vehicle (-1) dispatched in previous time slices will add orders to its schedule if possible. It is assumed that this negative node (-1) includes the partial route $\{\text{Depot} \rightarrow 4 \rightarrow \ldots\}$ illustrated in phenotype solution of Fig. 6 as a bolded route (completed route) and the black node as a visited or served customer. This partial route can add orders 9 and 8 to its schedule feasibly and make a complete route as the route 2 in

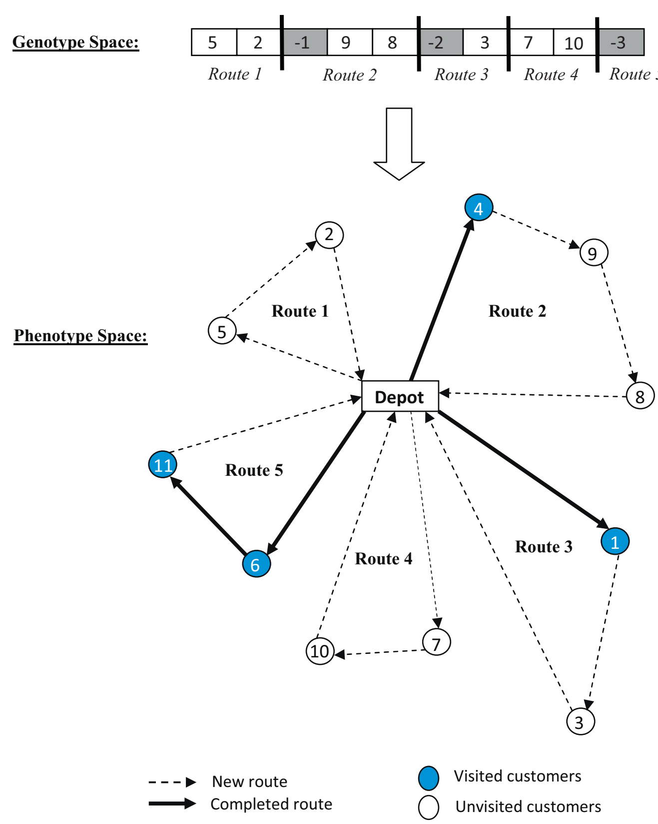

Fig. 6. A solution and its chromosome representation of step t_i ($i > 1$).

phenotype solution. Eventually, a typical representation with five routes is depicted in genotype solution of Fig. 6, in which routes 2, 3 and 5 are constructed on the previous time slices and the routes 1 and 4 should be constructed subsequently. When this chromosome is decoded, new customers can be added to these pre-existing routes if they still satisfy the feasibility conditions. According to this figure, decoding begins by the creation of a new vehicle with the index (-4) because the first element of the chromosome is a positive integer. This vehicle adds customer number 5 and customer number 2 to its schedule, assuming these requests do not violate feasibility conditions. This route is illustrated as the route 1 in phenotype solution of Fig. 6. Then the decoder encounters a negative integer (-1) in genotype solution. This means that additions to vehicle (-4) can no longer occur, and vehicle (-1) dispatched in previous time slices will add orders to its schedule if possible. As described before, this negative node (-1) includes the partial route $\{\text{Depot} \rightarrow 4 \rightarrow \ldots\}$ and it can add orders 9 and 8 to its schedule feasibly. Moreover, route 3 including the partial route of vehicle index (-2) can be finished by adding customer 3 to its schedule. This route is illustrated as the route 3 in phenotype solution of Fig. 6. No more additions to this route can be occurred because of the feasibility conditions and a new route (i.e., route 4) with the vehicle index (-5) should be constructed. It should be noted that the information of these vehicle indices should be stored in the list of dispatched vehicles, which maps the negative integer to the partial route of dispatched vehicle. Finally, the chromosome representation of Fig. 6 is corresponding to the phenotype solution of this figure, in which the bolded lines are the completed routes and the black nodes are the served or visited customers. Moreover, the white nodes are the unvisited customers and the dotted lines are the new routes that not yet travelled by the vehicles up to this time

The above-mentioned representation has the variable length and it grows in length with the new addition of customers, and reduces as collections of the visited customers are assigned to respective groups. A part of the initial population in the first step is generated by λ -interchange mechanism (50%) and part is generated randomly (for more detailed description of this procedure, see [26]). However, the initial populations in the subsequent steps $(t_i; i > 1)$ are obtained by generating a random permutation of size N_{POP}^{i} where N_{POP}^{i} is the number of positive and negative nodes in time slice t_i. Moreover, the initial solution of the first step is obtained by a push forward insertion heuristic (*PFIH*) method [26]. Besides, for each time slice $t_i; i > 1$, the initial solution is obtained by the feasible insertion of new customers (O_{i-1}) to the best position of the implemented solution S_i. In other words, by implementation of solution S_i within $[t_i, t_{i+1}]$, the string of negative and positive nodes are identified and then by the best feasible insertion of customers O_{i-1} in this string as the remained positive nodes, the feasible initial solution of step t_i is completed.

In the GA for evaluation of each chromosome, special fitness function is defined. However, in multi-objective applications of the GA, the Pareto ranking scheme has been often used [5–7]. The Pareto ranking process tries to rank the solutions to find the non-dominated solutions. It should be noted that the lower ranks are preferable and the chromosomes within rank 1 are the best in the current population. So according to this process, each solution gives a rank in respect of different objective values that shows the quality of it in comparison with other solutions. It is easily incorporated into the fitness evaluation process within a GA, by replacing the raw fitness scores with Pareto ranks (for more detailed descriptions, see [7]). Using a single fitness function as a weighted summation of different objective functions is usual for multi-objective applications. However for many MOP's, finding an effective weighting for the multiple dimensions is difficult and results in unsatisfactory performance and solutions. Therefore, this paper uses a ranking based approach, in which the chromosomes in each rank set represent

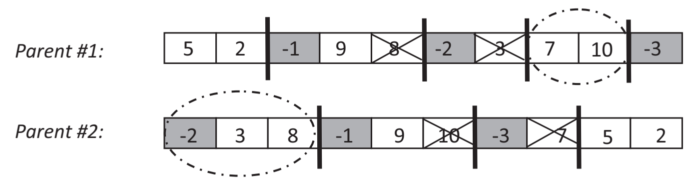

Fig. 7. Best cost best route crossover.

solutions that are in some sense incomparable with one another. Moreover, chromosomes assigned rank 1 are non-dominated, and inductively, those of rank $i+1$ are dominated by all chromosomes of ranks 1 through i. It should be noted that the non-dominated solutions in each population are identified based on the concept of dominance defined in Section 3.3.

The four objectives explained before define four independent dimensions in a multi-objective fitness space. Thus, each chromosome in the population is associated with a vector $\vec{v} = (n^-, d^-, c^-, s^+)$ where n^-, d^- and c^- are the objective values for number of vehicles, total travel distance and total waiting time imposed on each vehicle, respectively. Also, s^+ is the objective value for the total satisfaction rate of customers. These four dimensions are retained as independent values, to be eventually used by the Pareto ranking procedure to create for the population a set of integral ranks ≥ 1. These ranks are then used by the proposed GA as fitness values for generating the next population.

This paper uses a standard k-tournament selection where a tournament set of size k is randomly drawn from the population and the chromosome with a lower rank is selected and will then be recombined via the recombination operators to create potential new population.

One of the unique and important aspects of the GA is the important role of the crossover operator. The classical crossovers (e.g., one-point crossover and n-point crossover) are not appropriate for this sequencing model because of duplication and omission of vertices. Our recent paper [7] uses the best cost-best rout crossover (BCBRC), which selects a best route from each parent and then for a given parent, the customers in the chosen route from the opposite parent are removed. The final step is to locate the best possible locations for the removed customers in the corresponding children. This paper uses the modified BCBRC, which is similar to the BCBRC selects the best route from each parent and then removes the chosen customers of one parent from the opposite parent. This procedure is illustrated in Fig. 7. According to this figure, route 4 from parent #1 is selected and the customers on this route are removed from the routes of parent #2. This process is done similarly for another parent. Hereinafter for each parent the best location of removed customers are determined by the insertion procedure one at a time. This procedure is continued until two feasible offspring are produced.

The main point considered is related to map the genotypic information (i.e., chromosome) to the explicit fitness or phenotype, in this case a set of routes. Chromosome encoding as mentioned earlier, represents the existing routes and new customer requests. The customers in the existing routes constructed in the previous time slices are placed in groups, and each group is assigned a negative number. During the proposed process, when the solving algorithm encounters a negative node, it looks this integer up in a list of distributed vehicles, which maps the negative integer to a distributed vehicle index. To give an example of chromosome decoding the chromosome (parent #1) shown in Fig. 7 is used. This chromosome has 7 positive nodes that are new orders or came to a system in the previous planning and not yet served until now. Moreover, this chromosome has three vehicles visited a number of customers represented by the negative integers. Decoding begins by the creation

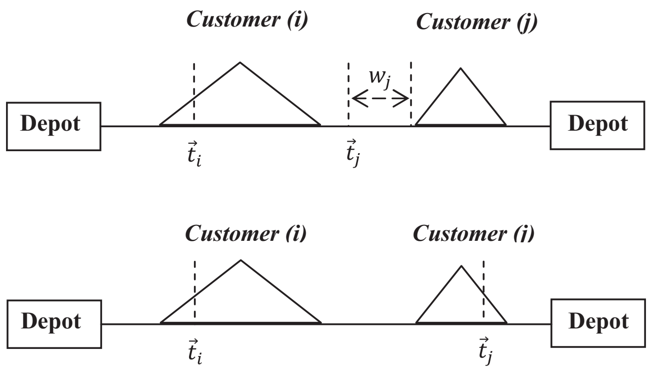

Fig. 8. Typical routes with and without waiting times.

of a new vehicle (i.e., vehicle 4) because the first element of the chromosome is a positive integer. This vehicle adds order number 5 and order number 2 to its schedule, assuming these requests do not violate its capacity constraints. Then, we encounter a negative integer. This means that the current route must be closed and vehicle 1 corresponding to negative node (−1) will add orders to its schedule if possible. Assuming that this vehicle has enough capacity, orders 9 and 8 are added to its schedule. This route should be closed again because of the next negative node (−2) and vehicle 2 should be called. It is assumed that this vehicle has enough capacity to allow the addition of order 3. However, order 7 does not since it violates the feasibility constraints of this vehicle. Thus, this route should be closed and a new vehicle (i.e., vehicle 5) will be created to accept orders 7 and 10. Finally, another negative node (−3) is encountered, corresponding to the third vehicle, labeled as vehicle 3, with no new orders placed.

As mentioned earlier, the customers' satisfaction for receiving the services is reflected by fuzzy time windows. Thus, except to the previous operators that focused on travel time and travel distance, we need a special operator to improve the satisfaction rate of each customer. This operator hereafter called satisfaction improvement operator (SIO). This operator tries to determine the best arrival time for each vehicle to maximize the total grade of satisfaction without increasing the waiting time and by pushing the waiting time of vehicles on each customer along the routes. Therefore, this operator scans the feasible schedule from left to right and tries to find out a possible forward push within the route. This push will increase the total degree of satisfaction along the route without violating the feasibility conditions. In general, the SIO operator is applied on the chromosomes with the following characteristics:

1- SIO is applied on the solutions that has at least one vehicle with non-zero waiting time. Fig. 8 shows the typical routes with and without waiting time, where $\vec{t}_i$ is the arrival time of vehicle to customer i. According to this figure, the first vehicle incurs waiting time (w_j) for customer j.

2- If a vehicle incurs more than one w_j along a route, the route should be divided to some sections (each section is named "path") according to the number of vehicles waiting time. Then, the SIO operator is applied on each path that has one vehicle waiting time (w_j). This procedure is shown in Fig. 9, in which the SIO operator is applied on path 1 and 2.

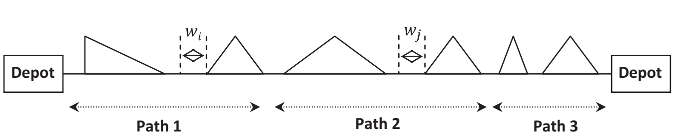

Fig. 9. Dividing the route to three paths.

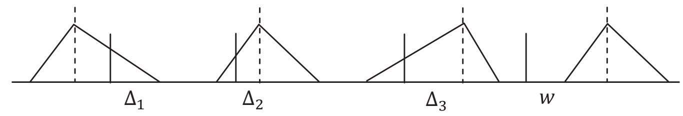

Fig. 10. Typical path for applying the SIO.

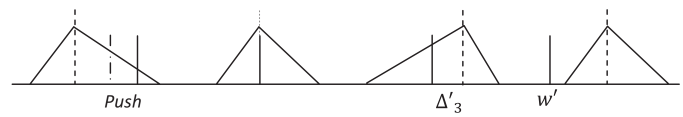

Fig. 11. Improved satisfaction rate by new vehicle arrival times.

3- The sum of the slope of the satisfaction function at the vehicle arrival time to each customer for each path should be larger than zero. In this case, a possible forward push will cause the increase of the total grade of satisfaction on the path.

Thus, this operator is applied on each path with the above characteristics. In each step, the feasible forward push should be calculated and the vehicle arrival time to customers should be pushed according to it. The feasible forward push in each step is as Push $= \min(\Delta_i, w)$ where:

$$\Delta_i = \begin{cases} u_i - t_i & \text{if} \quad e_i < t_i < u_i \\ l_i - t_i & \text{if} \quad u_i < t_i < l_i \end{cases}$$

According to this formula, when the arrival time is before than u_i, the maximum feasible push for customer i is from arrival time (t_i) to the time that the satisfaction of customer i is maximum. Moreover when the arrival time is before than l_i and after the u_i, the maximum feasible push is from arrival time until the latest arrival time to keep the feasibility conditions of hard time windows. After this push, the part of path from the *Customer** to the end of the path is considered and the above characteristics are checked again. The *Customer** is the customer that the previous minimum push has been found on it. Then, if the characteristics are satisfied, the new feasible forward push will be found and the operator will apply on this section again. This procedure is repeated until the new feasible forward push could not be found. For more in-depth understanding, the applying of the proposed satisfaction improvement operator (SIO) is illustrated in Fig. 10. This figure shows a typical path along the vehicle arrival time to each customer and the feasible forward push of each route customer.

By assuming that the mentioned characteristics are satisfied and the feasible forward push is minimized on the second customer, the SIO is applied to improve the total satisfaction rate of customers. Therefore, the path of Fig. 10 affected by this operator is converted to Fig. 11.

According to this figure, the total satisfaction rate of customers is increased by assuming that the sum of the slope of the satisfaction function at the vehicle arrival time to each customer for this path is positive and the total waiting time is not changed. The total waiting time in Fig. 10 is w and in Fig. 11 is w' + Push where; $w = w'$ + Push. Next, by considering the second customer as *Customer**, the new feasible forward push is found in accordance to Push$' = \min(\Delta'_3, w')$ and the operator is applied again. This procedure is repeated until the new feasible forward push cannot be found. The performance of the proposed simulator structure is examined in the next section.

The mentioned procedure and operators are applicable in the networks with crisp travel times. However, some of them should be modified when the fuzzy travel times are considered. As mentioned

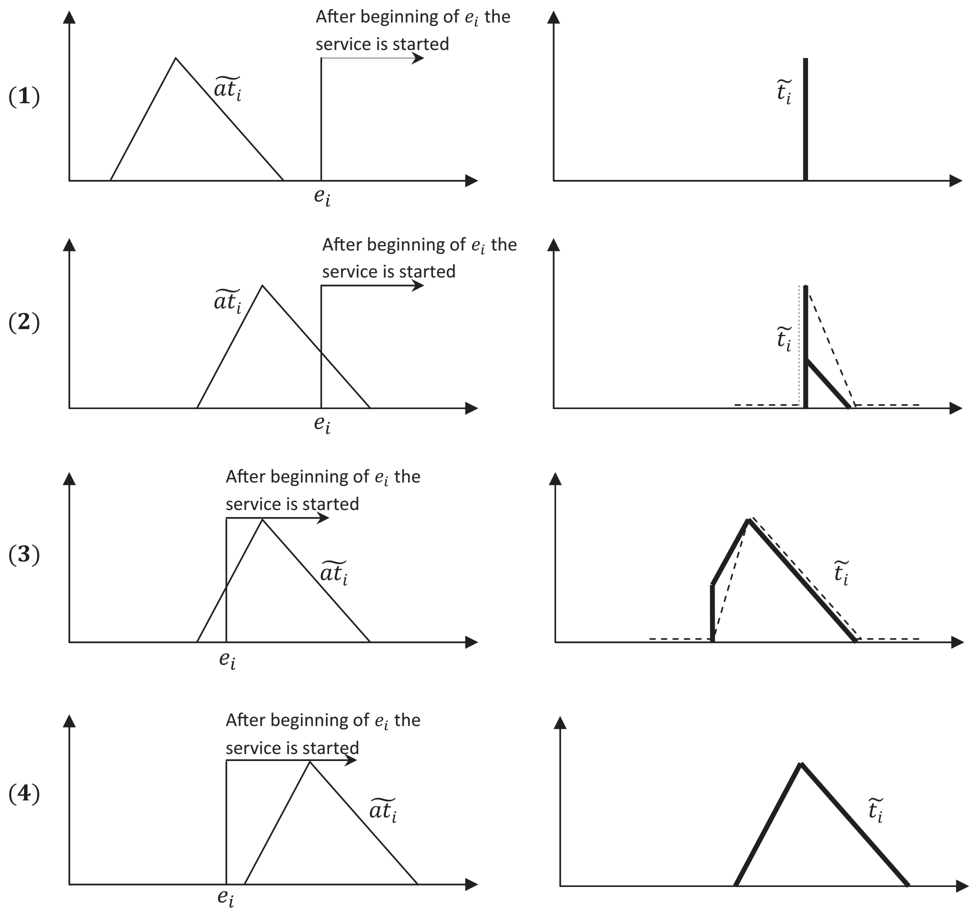

Fig. 12. Service start time based on the arrival time and earliest start time.

before, the service at customer i cannot start before the earliest start time (e_i or e'_i). After beginning of e_i or e'_i, service can be started and there is no possibility to start service before e_i even if the vehicle is already there. So the service start time on customer i ($\tilde{t}_i$) is more dependent to earliest service time (e_i or e'_i) and the possible arrival time ($\widetilde{at}_i$). In this regard, the value of $\tilde{t}_i$ can be calculated based on four possible relation between e_i and $\widetilde{ar}_i$ as $\max\{\widetilde{ar}_i, e_i\}$ as illustrated in Fig. 12. In this figure, the dashed lines are the approximation made to adjust the fuzzy number to triangular fuzzy representation.

The first case of this figure identifies the service start time when all possible arrival time is before the beginning of time window. In the second case, the possible arrival is before and after the earliest start time; however, the modal value is before it. The third case is similar the previous case; however, the modal value is after the beginning of time window. In the last case, all possible arrival time occur after the beginning of the time window and it is considered as the service start time. When the service start time is identified, the value of satisfaction level for each customer ($\mu_i(\tilde{t}_i)$) should computed based on this start time ($\tilde{t}_i$) and the predefined fuzzy time window. It should be noted that the concept of necessity measure $N(X)$, which describes to what extent to be believe in $x \in X$, is used here. In other word, when X is the set of time values not later than service start time t_i, then $N(X)$ is as the certainty that the service has already been started by the moment t_i. So the greater necessity measure value is as more confidence of the start of service by the time of interest. According to this concept and the value of $N(X)$, the satisfaction level of each customer ($\mu_i(\tilde{t}_i)$) is easily computed based on its service start time and time window. This procedure is illustrated in Fig. 13. The left hand side of this figure computes the necessity measure based on the service start time and the other figure computes the satisfaction level based on the fuzzy time window.

Based on Fig. 13, the value of the satisfaction level for each customer is computed by:

$$\sup_{t_i} \min\{\mu_{FTW}(t_i), N(-\infty, t_i)\} \tag{11}$$

The other important point considered here is the maximum allowable travel time of vehicles that forces them to return to depot under the condition of capacity constraint. This feasibility condition is easily controlled for crisp travel times. However, when the fuzzy travel time is considered, an efficient approach should be taken. Based on the descriptions of Section 3.2 and modeling the fuzzy travel times as triangular fuzzy number, the total travelled time of vehicles is a triangular fuzzy number. So, when the maximum allowable travel time of vehicles is denoted by R, the available time of each vehicle k after serving the n customers ($\widetilde{AT}_n^k$) is $\widetilde{AT}_n^k = R - \widetilde{TO}_n^k$. Where, $\widetilde{TO}_n^k$ is the total travelled time by vehicle k after serving the nth customer and it is $\widetilde{TO}_n^k = \widetilde{t_{n-1}} + T_{n-1,n}$. It is clear that the "strength" of preference for this vehicle to serve the next customer after serving n customers depends on available time $\widetilde{AT}_n^k$. This preference can be "LOW", "MEDIUM" or "HIGH" and the preference index is denoted by $p_n \in [0, 1]$, which describes the strength of this preference to send the vehicle to the next customer after it has served n customers. When $p_n = 1$, the vehicle is absolutely certain to serve the next customer. When $p_n = 0$, the vehicle should return to the depot. Available time ($\widetilde{AT}_n^k$) can also be subjectively estimated as "SMALL", "MEDIUM" and "LARGE". The membership functions of these fuzzy sets are shown in Fig. 14.

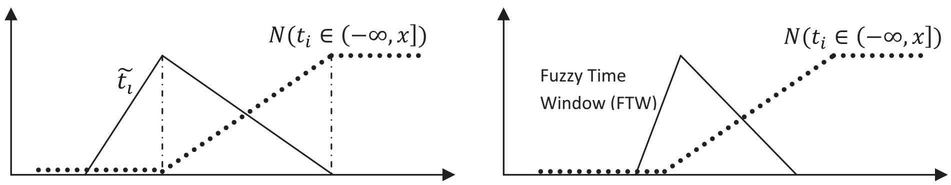

Fig. 13. Satisfaction level of each customer.

It is assumed that the strength of preference depends on available time, and hereby three main rules can be considered:

- Rule 1: if $\widetilde{AT}_n^k$ = SMALL then p_n = LOW,
- Rule 2: if $\widetilde{AT}_n^k$ = MEDIUM then p_n = MEDIUM,
- Rule 3: if $\widetilde{AT}_n^k$ = LARGE then p_n = HIGH

Every rule represents a fuzzy relation between the available time and preference strength. So for known available time $\widetilde{AT}_n^k$ that remains after serving n customers, the strength of preference (p_n^*) to send the vehicle to the next customer is easily calculated. This approximate reasoning procedure is graphically shown in Fig. 15. This figure presents the membership function of the index preference obtained by applying the approximate rules and its center of gravity.

Finally, based on the value of chosen preference index (p_k^*), a decision should be made whether to send the vehicle to the next customer or return it to the depot. This decision is made as follows: the vehicle should be sent to the next customer when $p_k^* \geq p^{**}$ where p^{**} is given from interval [0,1]. Otherwise, it should be return to the depot. It should be noted that the lower values of p^{**} represent the endeavor to use the travelled time of vehicle as much as possible.

5. Computational analysis

In this section, we report on the computational results obtained using the proposed model and structure in test problems and compare these results to the best known solutions obtained for these problems. As described before, the proposed dynamic structure consists of several consecutive static models and in the first stage, the performance of the proposed method should be considered in static conditions, in which all the customers are known at that time. Then, the other model assumptions and developments are added gradually and changes are examined. So in the beginning, the proposed model is considered in static conditions with two objectives minimizing the total distance traveled and the total number of vehicles that are the most common objectives used by other researchers alternatively. After that two another defined objective functions, namely minimizing the vehicles waiting time and maximizing the customers' satisfaction, are added and the developed model is considered in dynamic conditions. The experimental results use the standard Solomon's VRPTW benchmark problem instances that are available in [26]. The Solomon's problems consist of 56 data sets, which have been extensively used for benchmarking different heuristics in the literature over the years. The problems vary in a fleet size, vehicle capacity, traveling time of vehicles, spatial and temporal distribution of customers. In addition to that, the time windows allocated for every customer and the percentage of customers with tight time-windows constraint also vary for different test cases.

The Solomon's data are clustered into six classes; namely C1, C2, R1, R2, RC1 and RC2. Problems in the C category mean the problem is clustered. It means that customers are clustered either geographically or according to time windows. Problems in category R mean the customer locations are uniformly distributed whereas those in category RC imply hybrid problems with mixed characteristics from both C and R categories. Furthermore, for C1, R1 and RC1 problem sets, the time window is narrow for the depot, hence only a few customers can be served by one vehicle. Conversely, the remaining problem sets have wider time windows hence many customers can be served by main vehicles. It should be noted that the proposed algorithm is coded and run on a PC with Core 2 Duo CPU (3.00 GHz) and 2.9 GB of RAM. Moreover, the model is implemented under parameters of population size = 100, generation number = 1000, crossover rate = 0.80, mutation rate = 0.40, improve the solution by 2-interchange (GB) and 1-interchange (FB) operators [7], selection rate of improvement operators = 0.5 and repetition for experiments = 10.

Table 1 presents a summary of results when the static VRPTW with bi-objectives minimizing the total distance traveled and the total number of vehicles is considered. This table also compares the findings with the best known solutions reported in the literature. These results are as same as our recent experience on the bi-objective static VRPTW [7], because of using the same main operators in the static conditions. Bolded numbers in this table indicate that the solutions are obtained by the suggestive method are almost same as the best known or indicate an improvement on the best currently known results from the literature while considering either number of vehicles or distance cost. However, some minor improvements are observed in some instances over 10 runs. In this table, the non-dominated solutions of each instance are reported over 10 runs. Even though there is just a single Pareto solution for most of the instance problems, conflicting behavior for some instances are observed. As shown in some experiments, there is a single Pareto solution that is optimal to the best known in both vehicle and distance dimensions. Other solutions, such as some

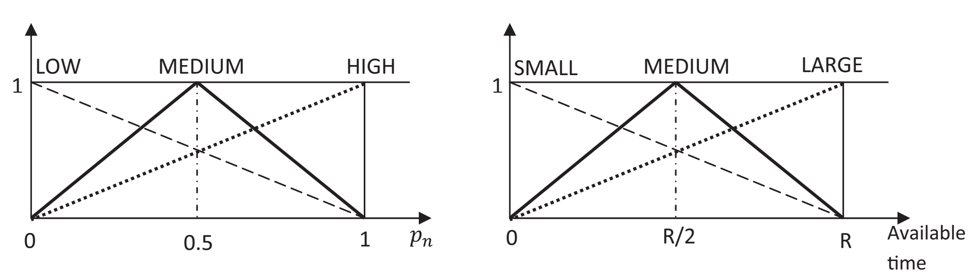

Fig. 14. Fuzzy sets for preference strength (left hand) and available time (right hand).

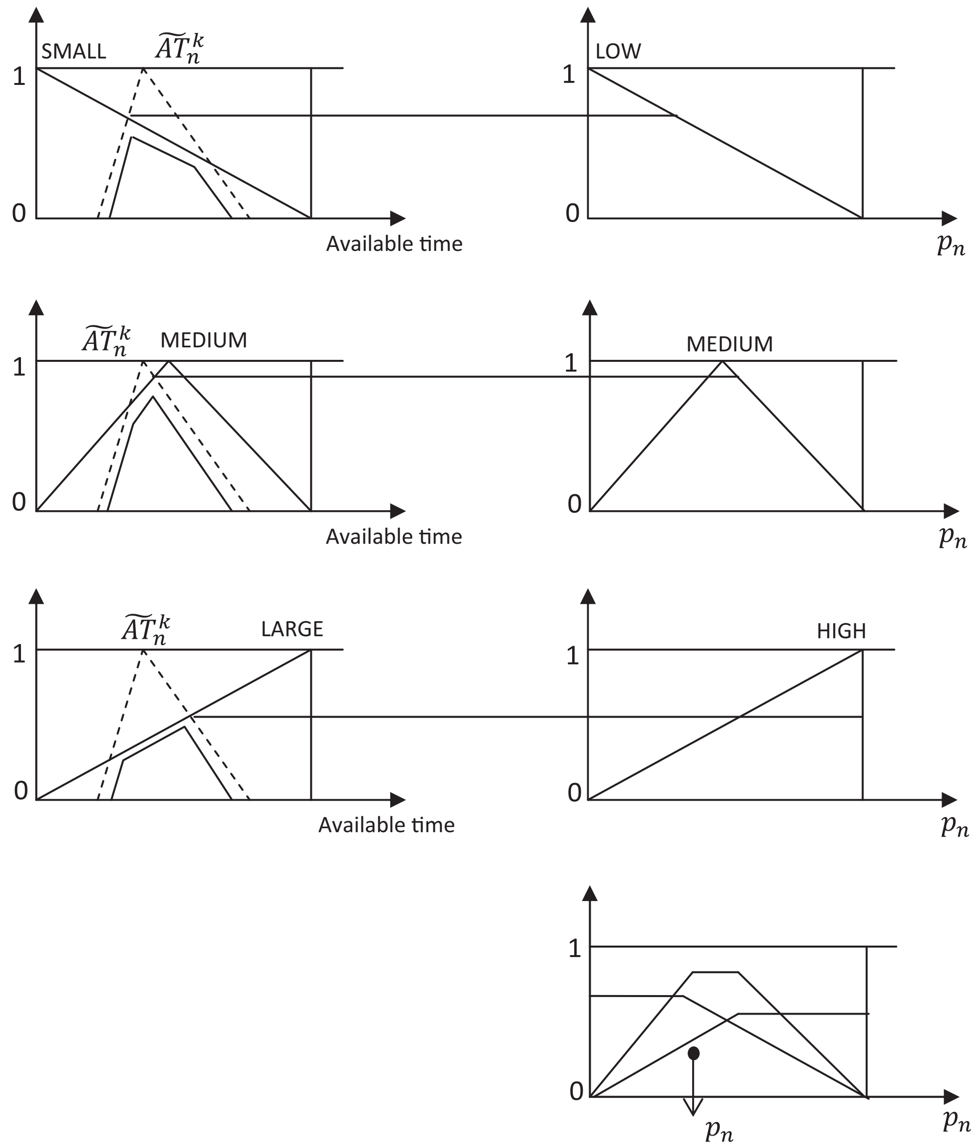

Fig. 15. Determining preference strength (p_n^*) for known available time ($\widetilde{AT}_n^k$).

instances in classes R and RC, reduce the distance significantly, but at the expense of adding extra vehicles. For example, in the R103 instance, the distance cost of a solution is reduced as the number of vehicles is increased. This behavior is observed in some instances of class R and RC in this table. In contrast, all instances in class C are having positively correlating objectives. When the number of vehicles is increased, the routing cost of a solution is increased too.

A more detailed analysis about the population distribution and different objectives behavior of all Solomon's classes is described later. Moreover, the average computational time for classes C1, R1 and RC1 varies between 2 and 3 h with 1000 generations and is between 5 and 7 h for classes C2, R2 and RC2. The second classes require a larger CPU time due to the longer time windows, which allow a more flexible arrangement in the routing construction process. Eventually, the results obtained by this bi-objective model in a static condition in general are quite good in respect of time and quality as compared to the best published results.

One of the most important points in making and implementation of proposed method is relevant to parameters tuning. In the field of evolutionary computing one traditionally distinguishes two approaches to choosing parameter values, following the scheme offered in [44]: *parameter tuning*, where (good) parameter values are established before the run of a given EA. In this case, parameter values are fixed in the initialization stage and do not change while the EA is running; *parameter control*, where (good) parameter values are established during the run of a given EA. In this case, parameter values are given an initial value when starting the EA and they undergo changes while the EA is running. In this paper the first approach is used and in this regard, to make this approach applicable in other problems, it is tried to adjust the parameters based on the different classes of considered instances. For this propose, the changes of each parameter in the pre-specified variation range are considered whenever the other parameters are fixed. For instance, when all parameters of proposed method are fixed on a specific value, the rate of crossover operator is tuned according to Fig. 16. According to this figure when the mutation rate is fixed on 0.4; population size is fixed on 100; generation number is 1000, the variation of crossover rate between 0.4, 0.6 and 0.8 is checked over all problems of class R. This figure only shows the performance of different crossover rates for class R and it should be noted that the similar figures are considered for all other class of Solomon's instances and the final decision is made

Table 1
Testing results of the Solomon's instances for the bi-objective static VRPTW.

Pro.	Best known [5–7,36]		Proposed method (bi-objective static VRPTW)			
	Vehicle #	Distance cost	Pareto solutions		Average in 10 runs	
			Vehicle #	Distance cost	Vehicle #	Distance cost
C101	10	828.94	**10**	**828.94**	10	828.94
C102	10	828.94	**10**	**828.94**	10	828.94
C103	10	828.06	**10**	**828.06**	10	864.20
C104	10	824.78	**10**	**824.78**	10	833.23
C105	10	828.94	**10**	**828.94**	10	828.94
C106	10	828.94	**10**	**828.94**	10	828.94
C107	10	828.94	**10**	**828.94**	10	840.15
C108	10	828.94	**10**	**828.94**	10	829.04
C109	10	828.94	**10**	**828.94**	10	828.94
C201	3	591.56	**3**	**591.56**	3	591.56
C202	3	591.56	**3**	**591.56**	3	593.85
C203	3	591.17	**3**	**591.17**	3	620.10
C204	3	590.60	**3**	599.96	3	613.32
C205	3	588.16	**3**	**588.16**	3	588.82
C206	3	588.49	**3**	588.88	3	590.02
C207	3	588.29	**3**	591.56	3	591.97
C208	3	588.32	**3**	**588.32**	3	590.39
R101	19	1650.80	19	1677.0	19.7	1664.7
			20	1651.1		
R102	17	1434	17	1566.8	18.3	1576.5
			18	1504.1		
			19	1487.5		
R103	14	1237.05	14	1287.0	14.8	1290.7
			15	1264.2		
R104	10	974.24	**10**	**974.24**	11.2	1001.0
R105	14	1377.11	15	1424.6	15.7	1456.3
			16	1382.5		
R106	12	1252.03	13	1269.0	13.7	1366.8
R107	11	1100.52	11	1108.8	11.9	1201.9
R108	10	960.26	10	971.91	10.3	1014.9
R109	12	1169.85	12	1222.7	13.5	1210.8
			14	1203.0		
R110	11	1112.21	12	1156.5	12	1161.8
R111	10	1096.72	11	1111.9	12.3	1154.4
R112	10	976.99	**10**	**977.0**	10.2	1011.5
R201	5	1206.42	**4**	1351.4	4	1359.0
R202	4	1091.21	**4**	**1091.22**	4	1173.1
R203	4	935.04	3	1041.0	4.5	1012.5
			5	995.8		
			6	978.5		
R204	3	789.72	3	1130.6	5.4	839.82
			4	927.7		
			5	831.8		
			6	826.2		
R205	3	994.42	4	1087.8	3.4	1168.9
			3	1422.3		
R206	3	833	3	940.12	3	1024.1
R207	3	814.78	3	904.90	3	1001.0
R208	2	726.823	3	**715.38**	3	778.25
R209	3	855	4	979.7	4	998.7
R210	3	954.12	**3**	**938.58**	3.2	1045.1
R211	4	761.10	4	1101.5	3.6	1205.8
			3	1310.4		
RC101	15	1636.92	15	1690.6	15.3	1698.4
			16	1678.9		
RC102	13	1470.26	15	1493.2	14.5	1509.8
RC103	11	1261.67	12	1331.8	12	1357.8
RC104	10	1135.48	11	1156.1	11	1293.0
RC105	16	1590.25	**15**	**1611.5**	15.7	1610.5
			16	**1589.4**		
RC106	11	1427.13	13	1437.6	13.7	1454.0
			14	**1425.3**		
RC107	11	1230.48	**11**	**1222.1**	12.2	1287.9
RC108	10	1142.66	11	1156.2	11.3	1186.3
RC201	6	1134.91	**4**	1423.7	4	1457.0
RC202	4	1181.99	4	1369.8	4.7	1422.4
RC203	4	1026.61	4	1060.0	5.0	1178.1
			6	**1020.1**		
RC204	3	798.46	**3**	799.52	3	926.74
RC205	4	1300.25	4	1410.3	4	1454.3

Table 1 (*Continued*)

Pro.	Best known [5–7,36]		Proposed method (bi-objective static VRPTW)			
	Vehicle #	Distance cost	Pareto solutions		Average in 10 runs	
			Vehicle #	Distance cost	Vehicle #	Distance cost
RC206	3	1153.93	4	1194.8	4	1195.5
RC207	4	1040.67	**4**	**1040.6**	4	1123.5
RC208	4	785.93	3	898.50	3.7	915.9

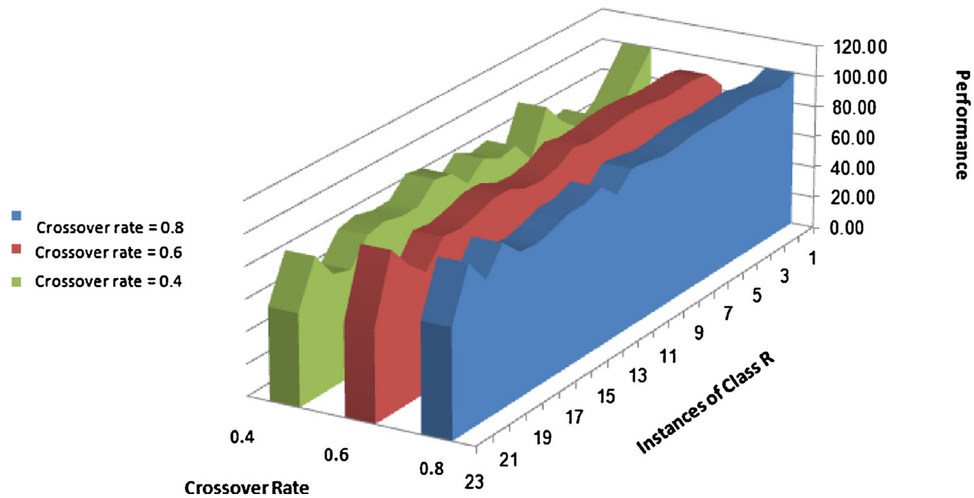

Fig. 16. Tuning of crossover rate when other parameters are fixed.

based on the average of performance values over all the different classes of C, R and RC. This figure shows 3D slices corresponding to specific Crossover rate, thus specific problem instances. Such a slice shows how the performance of an instance varies over the range of problem instances. According to this figure the average performance over all instances is 80.61% for crossover rate 0.4; 91.69% for crossover rate 0.6 and 95.19% for crossover rate 0.8. This procedure is done for all other variation of parameters and the best value for applying crossover rate is chosen based on the average performance over all instances.

Now the fuzzy time windows are considered instead of classical time windows and the changes of results via the proposed methods is analyzed on the Solomon's data sets. By considering fuzzy time windows, two another defined objective functions minimizing the vehicles waiting time and maximizing the customers' satisfaction are added. The proposed model should be implemented with four defined objective functions in a multi-objective manner and in static conditions. To use this model on the Solomon's instances, the predefined time windows of customers are changed to the triple $[e_i, u_i, l_i]$ where u_i is selected randomly within $[e_i, l_i]$. Table 2 presents a result of some randomly selected Solomon's instance problems from Table 1 where all customers are considered important. The column labeled "Best known" gives the best known published solutions in the literature; column "bi-objective VRPTW" gives the non-dominated solutions reported in Table 1 when the static VRP with classical time windows along with two objective functions is considered; column "multi-objective fuzzy VRPTW" reports the results of proposed model and solution when the static VRP with fuzzy time windows is interpreted as a multi-objective optimization problem with four objective functions. In this Table, the column "multi-objective fuzzy VRPTW" is divided in four other columns representing the objectives and non-dominated solutions.

In the proposed approach, when using the Pareto ranking, the DM's preference relation among the objectives, namely number of vehicles and total traveling distance, is taken into consideration to choose among the solutions. Nevertheless, a way to select a solution from the set of Pareto solutions is proposed and executed later. It should be noted that in some experiments there are more than 50 or 60 non-dominated solutions that some of them are reported in Table 2 for concisely. Moreover, it is tried to analyze the relationship between all objectives and check the behavior of them in each classes. In general, the relationship between these defined objectives in a routing problem is unknown until the problem is solved in a proper multi-objective manner. These objectives may be positively correlated with each other or they may be conflicting to each other. Obviously, such these relationships can be easily discovered using the proposed method; however, it is hard to be found if conventional single-objective vehicle routing approaches are used.

According to Table 2 showing the results of the Solomon's 56 data sets, all instances in the categories of C1 and C2 have the correlating objectives positively when the first two objectives are considered. In other words, the routing cost of a solution is increased as the number of vehicles is increased. For instance, Fig. 17 represents this behavior for problems C107 and C204 by considering the population distribution in respect of the distance cost and the number of vehicles.

Moreover, according to Table 2 with the fixed number of vehicles, the waiting cost of a solution is deteriorated as the routing cost is worsened. In this situation, the satisfaction rate of customers is improved very little and non-tangible. So in general, it can be expressed that the multi-objective manner is not required for the C category due to have the correlating objectives. Then, the weighted sum method can be applied properly for this category.

It should be note that the tentative observations of implementing the proposed method is representative. In this case, the conflicting behaviors are more in R and RC categories and most of these instances have the conflicting objectives in a population distribution of them. For instance in problem R103, by changing the number of vehicles, the distance cost of solution is changed in the opposite direction. According to Table 2, the distance cost of solution is decreased as the number of vehicles is increased from 13 to 14 and 15. This conflicting behavior is not applied when the

Table 2
Testing results of Solomon's instances for the multi-objective static VRPFTW.

Pro.	Best known [18,19,22,23]		Proposed method (bi-objective VRPTW)		Proposed method (multi-objective fuzzy VRPTW)			
	Vehicle #	Distance cost	Vehicle #	Distance cost	Vehicle #	Distance cost	Waiting cost	Customers' satisfaction
C101	10	828.94	10	828.94	10	828.94	0	62.9
					10	854.50	74.5	63
C106	10	828.94	10	828.94	10	828.94	0	63.15
					10	852.0	12.8	63.24
C201	3	591.56	3	591.56	3	591.56	0	65.990
					3	594.3	10.9	65.995
					3	635.1	73.3	67
R103	14	1237.05	14	1287.0	14	1287.0	545.87	36.51
			15	1264.2	15	1277.5	757.5	33.3
					15	1278.8	765.9	34.9
					15	1283.3	714.0	33.9
					16	1681.5	536.5	36.7
					17	1713.9	604.4	36.8
R108	10	960.26	10	971.91	11	995.8	324.1	45.1
					11	996.0	320.5	46
					11	1001.7	313.9	45.5
					11	1023.9	257	42.9
					12	1063.7	364.3	48.1
					12	1118.7	338	50.5
R203	4	935.04	3	1041.0	3	1041.1	2144.3	45.8
			5	995.8	4	1682.9	670.4	39.4
			6	978.5	4	1684.7	658.3	38.2
					5	1018.5	2017.9	46.6
					5	995.8	2455.1	44.1
					6	978.5	3231.2	46.3
					6	977.7	3231.2	46
					7	1510	1922.8	45
R204	3	789.72	3	1130.1	3	1130.6	857.8	46.5
			4	927.7	4	927.7	1301.9	47
			5	831.8	4	926.9	1308.9	46.6
			6	826.2	5	833.3	2288.1	50.5
					5	831.8	2289.5	51
					6	830.1	2730.5	50.5
					6	826.2	2774.6	48.8
RC101	15	1636.92	15	1690.6	16	1678.9	471.2	32.4
			16	1678.9	16	1680.5	471.2	32.8
					16	1680.6	464.5	31.7
					16	1685.8	463.6	33.7
					18	2130.9	459.6	36.1
					18	2186.4	469.5	36.3
RC102	13	1470.26	15	1493.2	15	1493.2	560.4	35.5
					15	1527.9	563.3	35.6
					15	1528.7	570.3	37.4
					15	1536.2	566.0	38.1
					15	1551.8	547.5	40.5
					16	1716.0	593.9	45.0
					17	1861.9	542.2	44.6
RC105	16	1590.25	15	1611.5	16	1589.4	577.2	33.7
			16	1589.4	16	1594.1	566.8	34.4
					16	1594.9	565.9	35.4
					16	1600.6	560.2	36.4
					16	1615.3	555.6	36.6
					16	1651.3	546.3	37.3
					17	1744.0	608.3	40.8
					17	1744.8	608.3	40.9
RC108	10	1142.66	11	1156.5	12	1200.9	229.1	39.7
					12	1206.4	216.4	41.7
					12	1226.2	200.0	43.0
					13	1363.3	268.1	44.6
					13	1552.9	205.2	46.3

vehicle number is increased to more than 15 vehicles. In this situation, the distance cost of solution is increased as the number of vehicle is increased from 15 to more vehicles. Moreover, according to Table 2, the total waiting time is increased when the number of vehicles is increased from 13 to 14 and 15 (or as the distance cost is decreased) and this increased waiting time does not help to improve the summation of the customers' satisfaction rate. This behavior is shown in Fig. 18, which is the population distribution of instance R103 in respect to the distance cost, total satisfaction rate and waiting cost.

According to Fig. 18-a, the customers' satisfaction rate is improved as the total travelling distance cost is deteriorated. This behavior shows a Pareto frontier of two objectives (Min–Max) in this figure. It should be noted that the some scatter seen in this

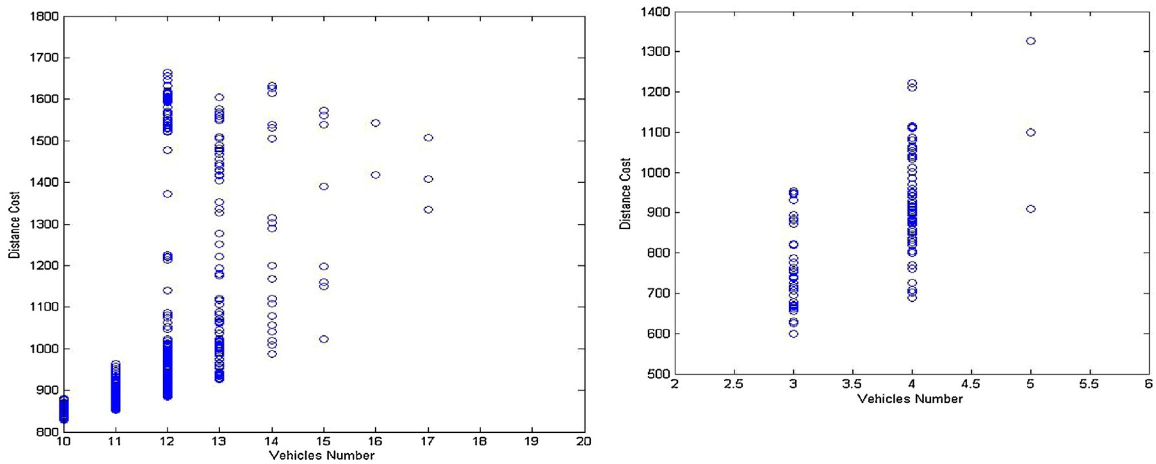

Fig. 17. Population distribution for problems C107 (left-hand) and C204 (right-hand).

figure is due to display of non-dominated solutions from four-dimensional space to two-dimensional space. Fig. 18-b illustrates the population distribution of this problem in respect to the distance and waiting cost. Based on this figure, the waiting cost is decreased as the distance cost is increased (or the number of vehicles is increased). Moreover, the relationship of the waiting cost and customers' satisfaction rate for problem R103 is illustrated in Fig. 18-c. In spite of the designed algorithm and operators (SIO) trying to improve the satisfaction rate of customers by using the current waiting time, these two objectives are independent of each other. This due to the nature of first categories of the Solomon's instances (C1, R1 and RC1) that have very lower waiting time than the second classes in general. In the second categories, the waiting time of vehicles is more because of being away of customers' time windows from the depot and the proposed improvement operator (SIO) can act efficiently. So in problem R103, the summation of the customers' satisfaction rate is improved by a more travelling distance cost and it is independent from the waiting time of vehicles.It should be noted that all instances in class R1 do not necessarily follow the mentioned behaviors for problem R103. For instance, in problem R108 unlike the previous example, the vehicle numbers and travelling distance cost are the correlating objectives positively and the distance cost of a solution is increased as the number of

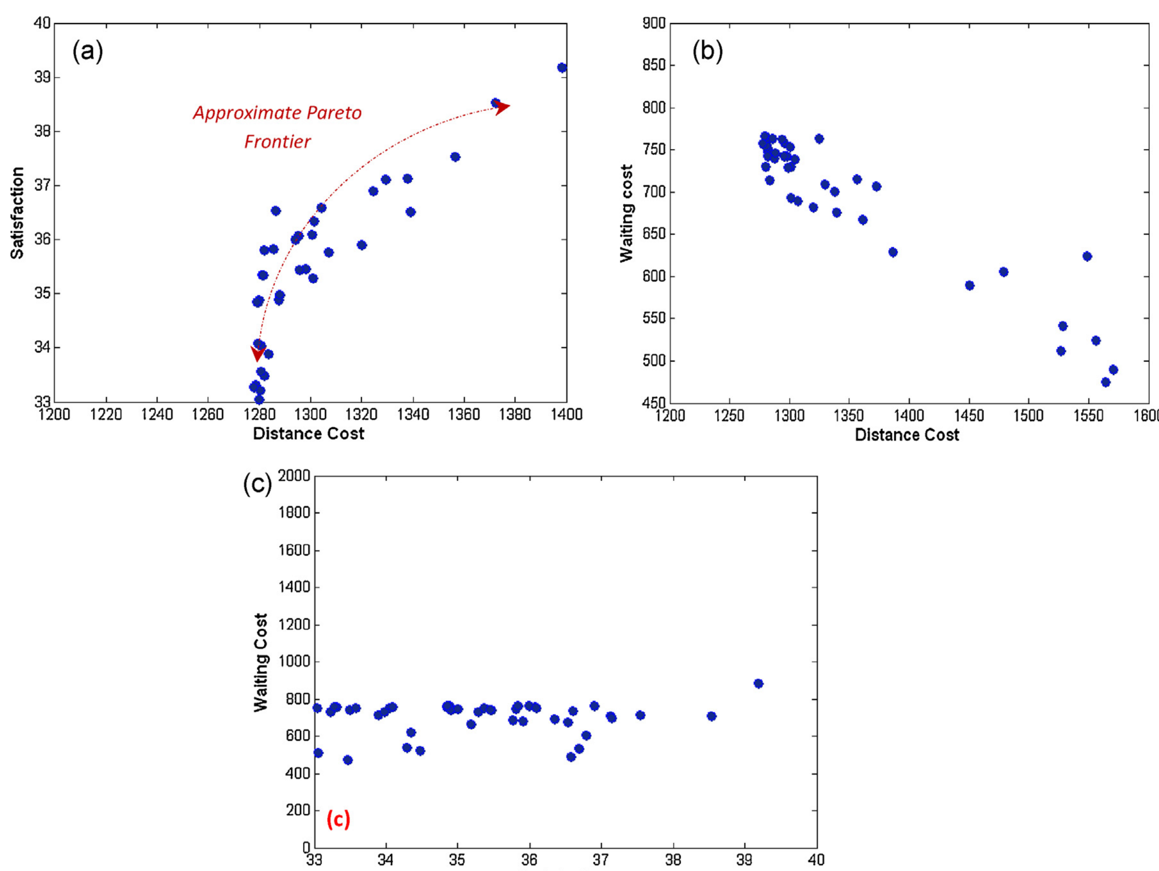

Fig. 18. Comparison of non-dominated solutions of problem R103.

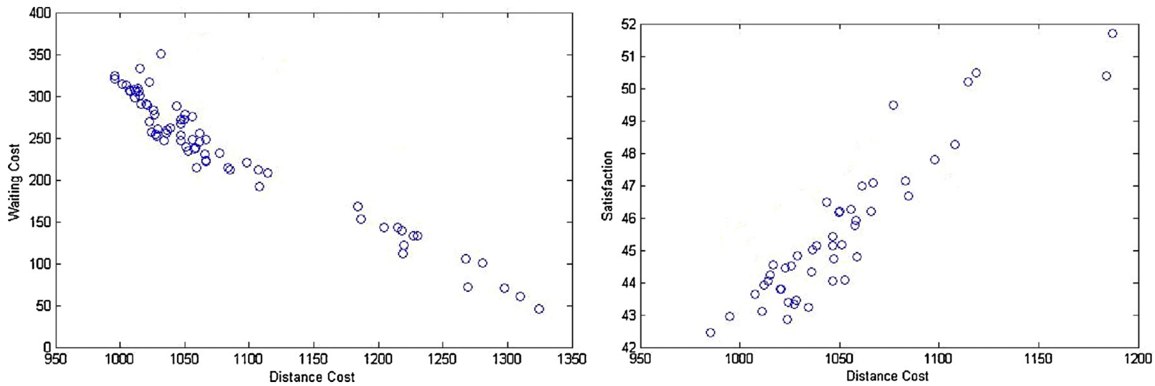

Fig. 19. Comparison of non-dominated solutions of problem R108.

vehicles is increased. However, the other behaviors of this problem are same as the problem R108 as illustrated in Fig. 19.

It should be noted that these conflicting and non-conflicting behaviors of the defined objectives are observed in different Solomon's classes as reported in Table 2. For example in problem R204, by increasing the number of vehicles, the distance cost of a solution is reduced in general. On the other hand, in the problem RC108, the distance cost of a solution is increased as the number of vehicles is increased and has the non-conflicting objectives unlike problem R204. These behaviors are shown in Fig. 20 for a population distribution in respect of the distance cost and the number of vehicles for this problem.

Moreover, in problem R204 unlike the first categories of the Solomon's instances (like R103 discussed before), the waiting time of vehicles is more because of being away of customers' time windows from a depot and the proposed improvement operator (SIO) can act efficiently. Thus, in this problem, the summation of the customers' satisfaction rate is increased by more waiting cost illustrated in Fig. 21. This figure also shows the relationship of the distance and waiting costs that has same behavior with the previous discussed instances.

Thus, as mentioned earlier, some instances have the conflicting objectives changed in the opposite direction and others have the non-conflicting objectives. Some of these instances are more detailed and described here, and rests of them are reported in Table 2. As mentioned earlier, the solved instances consider the important customers with a high degree of importance and the tight time windows (according to Fig. 2) are considered for them. Now, it is assumed that the planner encounters with the customers with different class of importance and there are some important and casual customers that should be served. So the model should be able to cope with the evaluation of deliveries for casual customers that violate the hard time widows. Table 3 shows the implementation results of this assumption for instance R204. It should be noted that the assignment of customers to one of groups is done randomly and it is assumed there are 20 casual customers selected randomly among the all customers.

According to this table, when the violation of time windows is permitted the solutions with better quality are obtained. For instance, the best distance cost for hard time windows was the cost of 826.2 with 6 vehicles and when the soft time windows are applied, the solution with the distance cost of 803.3 by 4 vehicles is obtained. Moreover, the waiting costs and customers' satisfaction are generally more better than the previous ones when the soft time windows are applied.

The results should be reconsidered, when the fuzzy travel times are used. One important parameter that should be determined is p^{**} where is given from [0,1]. This parameter can be evaluated for each objective function and the final value are determined by the planner or decision maker (DM). For instance, the obtained results of instance RC101 based on the first objective function (i.e., minimizing the total distance costs) and for different values of p^{**} are illustrated in Fig. 22.

According to Fig. 22, when $p^{**} = 0$, one big route is permitted to be planned and just the capacity constraint is considered. In this regard, when such vehicle has used all its capacity, it should be returned to the depot. When $p^{**} = 1$, the number of routes is equal to the number of customers and each route consists of only one customer. Based on this figure, the least total expected distance to be covered by the vehicles is realized when $0.35 < p^{**} < 0.6$. In this paper, it is assumed that the first objective function is more important than the other objectives, in which its value is considered

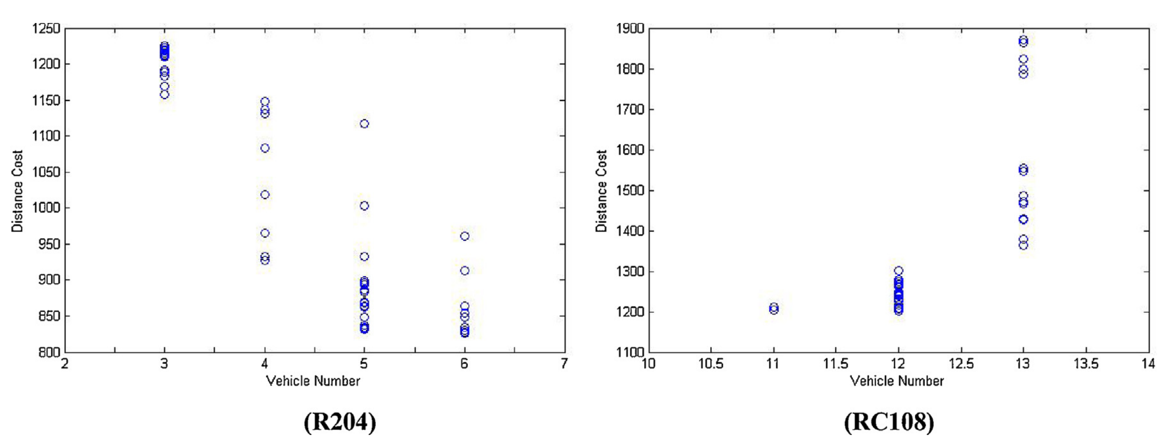

Fig. 20. Comparison of population distribution for problem R204 and RC108.

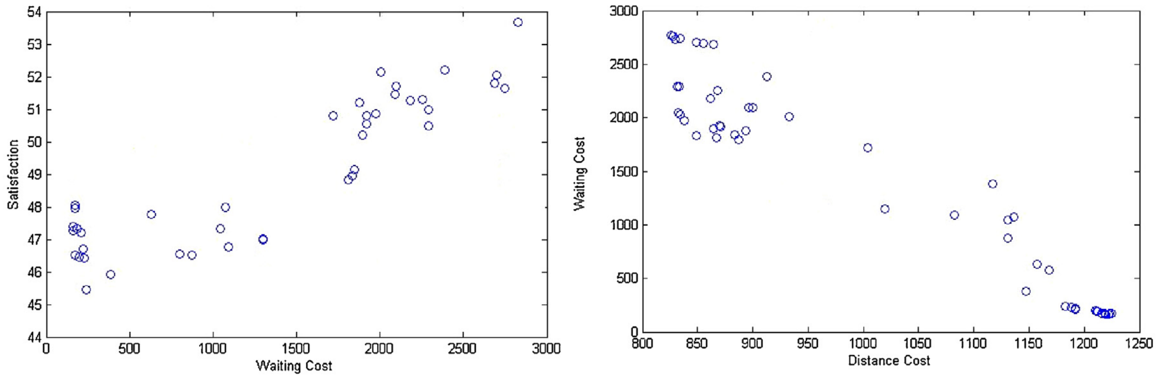

Fig. 21. Comparison of non-dominated solutions of problem R204.

Table 3
Testing results of MOF-VRPTW with different customers for instance R204.

Multi-objective fuzzy VRPTW with important customers				Multi-objective fuzzy VRPTW with casual and important customers			
Vehicle #	Distance cost	Waiting cost	Customers' satisfaction	Vehicle #	Distance cost	Waiting cost	Customers' satisfaction
				3	1129.0	242.9	49.0
3	1130.6	857.8	46.5	3	1096.3	199.6	46.8
4	927.7	1301.9	47	3	1082.4	265.6	48.3
4	926.9	1308.9	46.6	4	803.3	1283.3	49.2
5	833.3	2288.1	50.5	4	805.1	1282.1	49.2
5	831.8	2289.5	51	4	807.6	1298.0	50.5
6	830.1	2730.5	50.5	4	809.9	1275.2	49.6
6	826.2	2774.6	48.8	4	816.6	1289.0	51.3

0.45. Therefore, the results of applying the proposed method on the instance RC101 considering the fuzzy travel times is given in Table 4. To use this model on this instance, the predefined travel costs between each customer are changed to the triple $[t1_i, t2_i, t3_i]$. This table shows the Pareto solutions of instance RC101. The column labeled "Distance cost" is divided into two columns that show the total travelled distance and the total travelled time that are equal to each other in the previous scenario. It should be noted that the total travelled time is considered as an objective function alongside the other objectives and the total distance cost is just reported for the obtained solution. Moreover, the ranking method of fuzzy numbers based on the average and standard deviation is used in the mentioned Pareto ranking procedure to determine the Pareto solutions. According to this table, the minimum total number of vehicles and the minimum total distance travelled by them are deteriorated when the fuzzy travel times are considered. However, the satisfaction level of customers is improved in comparison with the previous scenario.

Now, the proposed model should be checked in a dynamic structure. In this situation, new requests are received during the working day and service to some previous requests is ended. As mentioned earlier, a working day is sub-divided into a number of discrete time periods (t_i), on which time slice t_i represents a partial static multi-objective VRP with fuzzy time windows, where the vehicles must serve all known customers without necessarily returning to the depot. Hence, at each step t_i, a static multi-objective VRP with

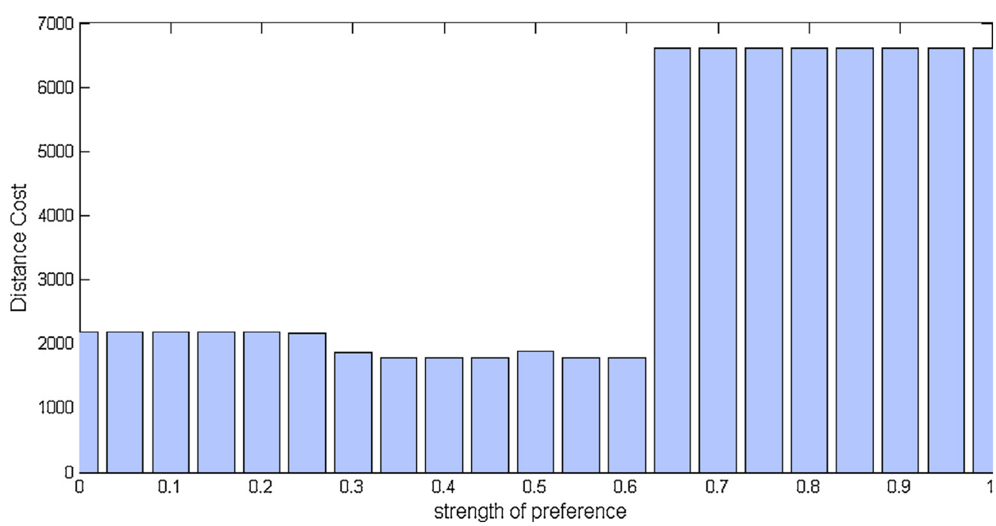

Fig. 22. Different distance costs based on different strength of problem RC101.

Table 4
Testing results of MOF-VRPTW with fuzzy travel times for instance RC101.

Multi-objective fuzzy VRPTW with crisp travel times				Multi-objective fuzzy VRPTW with fuzzy travel times				
Vehicle #	Distance cost	Waiting cost	Customers' satisfaction	Vehicle #	Distance cost		Waiting cost	Customers' satisfaction
					Total distance cost	Total travelled time		
16	1678.9	471.2	32.4	19	1778.9	(1801,1882,1885)	(493,521,527)	38.4
16	1680.5	471.2	32.8	19	1868.9	(1870,1967,1982)	(491,523,529)	39.0
16	1680.6	464.5	31.7	20	2187.4	(2301,2364,2365)	(525,541,545)	44.1
16	1685.8	463.6	33.7	20	2197.4	(2282,2355,2360)	(530,543,545)	45.1
18	2130.9	459.6	36.1	20	2197.4	(2284,2356,2360)	(529,540,543)	45.1
18	2186.4	469.5	36.3					

fuzzy time windows is solved in $[t_i+\delta, t_{i+1}]$ to find the solution S_{i+1} which should be implemented in the next time slice and within $[t_{i+1}, t_{i+2}]$. As observed before, at the end of each stage t_i, a set of non-dominated solutions are generated by applying the proposed static multi-objective VRP with fuzzy time windows. However, one solution (S_{i+1}) must be chosen by preferences of the decision maker (DM) to implement in the next time slice and within $[t_{i+1}, t_{i+2}]$. Hence, the displaced ideal method considering the LP metric is used for this propose. In this method, the ideal solution as $F^* = (f_1^*, f_2^*, f_3^*, f_4^*)$ is obtained by solving the model with each of objective functions separately. Then, the solution with the minimum distance in different metrics (e.g., $p = 1, 2$ or∞) from the ideal solution is chosen of non-dominated solutions. This procedure for problem R204 is applied as an example and the non-dominated solutions and their distances from the ideal solution with different metrics are computed. Based on the results; this problem has $F^* = (826.2, 3, 53.7, 160.1)$ as ideal solution and $fL = (1224.9, 6, 44.9, 2829.1)$ as an anti-ideal solution.

By considering the dynamic condition, the Solomon's data sets should be adapted for the proposed approach. Each customer in the Solomon's problem corresponds to an order (real time or determined request) with the same location and same weight. Hence, there are exactly 100 orders in each of our test problems, and the distance matrix between orders in our problems is exactly the same as that in the corresponding Solomon's problems. Also, the vehicles in our test problems have the same capacity as those in the Solomon's problems. The length of planning horizon denoted as H in our formulation and the time window of the depot is $[0, L_0]$ in the Solomon's problem. In this paper this time window corresponds to the planning horizon $[0, H]$ in our problems. If a decision maker sets the value of H in different from L_0, all the time-related data in the Solomon's problems have to be scaled by multiplying by H/L_0.

According to the previous descriptions, the pool of the customers' information is maintained to the management module as described before. One the most important of the customers' information is ct_i that is the call time of customer i with the central dispatching center (i.e., depot) to demand the requests for receiving. All information relevant to customer i is unknown for the decision maker before ct_i and whenever his/her connection with depot is made, all required information will be known. In the Solomon's problems, the call-in time for each customer is uniformly distributed in the following interval:

$$ct_i = [0.5 * \min(e_i, l_i - d_{0i} - 2\Delta), \min(e_i, l_i - d_{0i} - 2\Delta)] \qquad (12)$$

where, $[e_i, l_i]$ is time windows of customer i, d_{0i} is the travel distance from the depot to customer i, and Δ is the time between two consecutive decision stages.

It should be noted that all the requests or customers with non-positive call-in time are considered as determined requests (i.e., initial set of customers) at time $t = 0$ and the others are considered as real time or dynamic customers. For any real time request i with positive call-in time, it is feasible to dispatch a new vehicle from the depot at the next implementation stage, because the call-in time is no later than $l_i - d_{0i} - 2\Delta$, and so there is always a feasible solution. In real cases depending on the system type, infeasible customers are sent to the next working day and they are considered as determined requests for that day. These customers may call-in to depot at the time the call center closes or there is not enough time to dispatch a new vehicle to service them. Thus, some instances of the Solomon's problems are chosen randomly and solved by the presented model. The results are reported in Table 5. According to this Table, each instance is solved in two different values of time between two consecutive decision stages (Δ). In the first case, the planning horizon is divided to three decision stages, and in the other case it is divided to five decision stages.

According to this Table, the quality of solutions in the dynamic environment is generally lower than solutions in a static environment. Moreover, this quality is strongly dependent on how customers entering and calling to the decision system. For example, it is assumed that the best route is obtained from customers 1–3 as sequence of $1 \to 2 \to 3$. This path in the proposed dynamic environment is formed only when these customers come-in to system earlier and in the mentioned order. However, if the customer 3 calls in to the system earlier than customers 2 and 1, this may be due to customer 3 is served, the best route is never made. Moreover, according to this table, the quality of solutions is dependent on amount of time between two consecutive decision stages (Δ) too. This quality is improved whatever this stage is longer, because the algorithm has more time to solve the partial static model. So, in the systems with a high degree of dynamism, the reaction time

Table 5
Testing results of the Solomon's instances for the multi-objective dynamic VRPFTW.

Pro.	$\Delta = 3$				$\Delta = 5$			
	Distance cost	Vehicle #	Customers' satisfaction	Waiting cost	Distance cost	Vehicle #	Customers' satisfaction	Waiting cost
R103	1833.12	20	36	608.12	1854.32	20	36.5	622.71
R108	1231.8	13	50.7	355.1	1596.4	15	52	374.2
R203	1820	9	46.6	2011	1801.2	9	47	1987
R204	1001.5	6	47.1	1520.01	986.82	6	48	1486.8
RC101	2219.1	21	36.1	788.5	2275.4	22	36.3	743.32
RC105	2010.5	20	42.5	720.5	2096.1	20	43	714.2

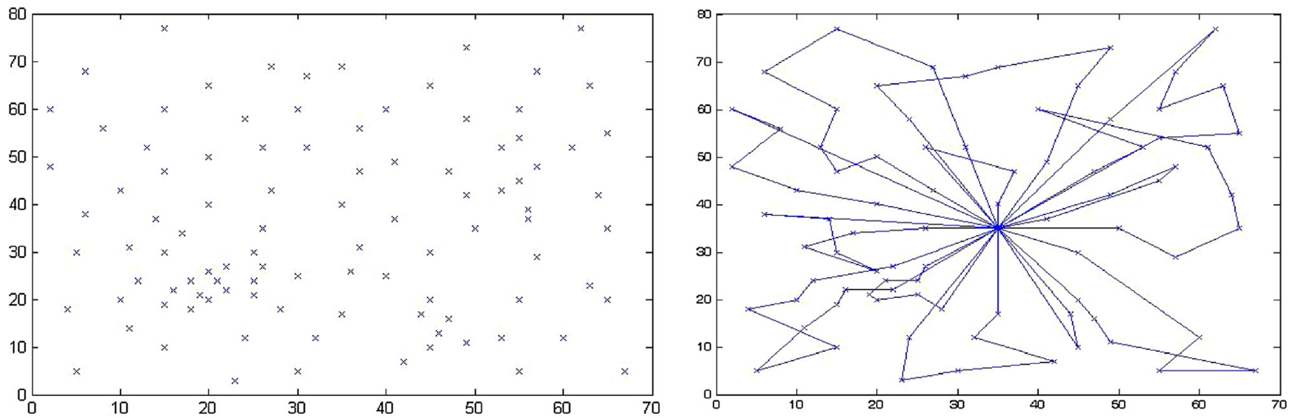

Fig. 23. First static solution to distribute of blood bags.

for service to real time requests is very short, and therefore the cost of finding the new solution is increased. However, in the systems with a large number of the determined requests, such as R204, the smaller Δ can help to improve of the obtained solutions. It is because of more re-optimization process on not served customers in consecutive stages. In contrast, in problem R103, which has the fewer number of determined requests, the quality of solution is decreased as the time between two consecutive decision stages is decreased. In this case,due to the greater real time requests in each stage and limited time to solve the model, the quality of solutions is decreased.In the systems with a high degree of dynamism, the reaction time to reply to new requests is very important, and in these systems the time between two consecutive decision stages is very short and there are a large number of decision stages. In these systems because of the importance of response time, the proposed optimization approach can be used without using of improvement heuristics or hill-climbing method. In this situation when Δ is very small, the simple heuristics (e.g., insertion methods) can be used.

6. A case study

This section shows the implementation process of the proposed model and its application in distribution of the needed blood bags of hospitals and clinics from one or more central distribution centers. The similar case was studied before by [45,46] with same basic definitions and now it is tried to develop it by the proposed conception of fuzzy time windows and dynamic requests in a defined structure. In the mentioned distribution plan, some hospitals and clinics that have the daily demands are exist and they are considered as determined requests. Dynamic requests are entered into system under emergency conditions (e.g., unpredictable events, wars and earthquake) in these situations one or more clinics or hospitals may be faced with the reduced blood supply and request their emergency requirements from the central depots. Thus, these dynamic necessities should be served immediately in a minimum reaction time. In this section, the mentioned assumptions of the above case study are considered by the concept of fuzzy time windows, proposed dynamic structure and following assumptions:

- The hard time windows are considered because of serving all requests without delay in service.
- The classical time windows are used for the determined requests of hospitals and clinics, and the fuzzy time windows are considered for the emergency or dynamic requests that should be responded with the minimum reaction time.
- The fuzzy time windows for emergency request are considered as $[e_i, u_i, l_i]$ where the earliest arrival time (e_i) is near to call-in time of customer (i) and the desired time (u_i) is also close to e_i. These assumptions are set for the ability to response to the emergency requests.
- Moreover, the dynamic requests use the narrow fuzzy time windows, which indicate the willingness of these requests in order to receive their services as soon as possible.

A complete randomly problem is considered to indicate the implementation process of the proposed model on the above-mentioned case study. It includes 82 determined clinics and hospitals with classical time windows and 18 real time requests with fuzzy time windows that arrive into the system during the planning horizon of the working day. The planning horizon is set as [0,90] and it is divided to three consecutive decision stages. Fig. 23 illustrates the geographical position of customers and the best firs static solution, which should be implemented in two first decision stages.

For more precisely, this solution is shown in Fig. 24 on the left hand side. This solution should be implemented up to time 60. Within time [0,30], the new customers are checked that may be called in for service. According to this solution, routes 3, 4, 5, 11, 12, and 13, which are specified, are the routes that their vehicles return to the depot until time 60 or they are on route to reach the depot. Therefore, they are permitted to return to the depot and be ready for re-dispatch if it is needed. Moreover, the specified customers in other routes are the customers that are in the service of vehicles or the vehicles are on route and close to them for service up to time 30.

Thus, the second partial static model, which should be solved within [30,60], is formed by checking the system at time 30. This new solution, which should be implemented within [60,90], is shown on the right hand side of Fig. 24.

The routes 1–7 are the previous partial routes formed during the past stages and the other routes are the new routes formed here. For instance, according to the past schedule (left hand side of Fig. 24), the 8th vehicle, which is equivalent to the 5th vehicle in the new solution, is responsible to serve the customer 66 after the customer 70 and before returning to the depot. In this new solution (i.e., right hand side of Fig. 24), the real time customer 86 is added to this route (i.e., route number 5) instead of customer 66 and a new other route with customers 66 and 95 is formed as route number 12. According to this solution, seven new routes should be formed and implemented within [60,90]. Six vehicles among these seven routes can be used from the previous vehicles, which are returned to the depot in the previous stage. Just one new other vehicle is used in a new schedule. Thus, this procedure can be continued for serving other real time requests that may come in to a system during the working day.

Route 1:	[0 60 59 3 16 44 38 43 50 58 0]
Route 2:	[0 69 4 11 7 73 71 68 0]
Route 3:	[0 23 20 24 0]
Route 4:	[0 28 25 76 65 0]
Route 5:	[0 6 53 13 21 62 63 0]
Route 6:	[0 32 54 48 47 51 55 56 0]
Route 7:	[0 8 45 41 14 10 40 34 36 37 52 0]
Route 8:	[0 15 12 61 64 22 70 66 0]
Route 9:	[0 74 1 79 18 29 82 77 0]
Route 10:	[0 72 9 17 19 80 81 78 75 0]
Route 11:	[0 27 67 0]
Route 12:	[0 2 35 33 26 49 0]
Route 13:	[0 5 39 42 30 31 46 57 0]

Route 1:	[0 60 59 3 16 44 38 43 50 58 0]
Route 2:	[0 69 4 11 7 73 71 68 83 90 0]
Route 3:	[0 32 54 48 47 51 56 94 55 0]
Route 4:	[0 8 45 41 14 10 40 34 36 37 88 0]
Route 5:	[0 15 12 61 64 22 70 86 0]
Route 6:	[0 74 1 79 18 29 82 77 0]
Route 7:	[0 72 9 17 19 80 81 78 75 97 0]
Route 8:	[0 91 84 0]
Route 9:	[0 85 0]
Route 10:	[0 92 0]
Route 11:	[0 52 99 87 0]
Route 12:	[0 95 66 0]
Route 13:	[0 96 93 0]
Route 14:	[0 89 98 100 0]

Fig. 24. Results of the proposed case study.

7. Conclusion

In this paper, a new multi-objective dynamic vehicle routing problem with fuzzy time windows (DVRPFTW) has been presented and solved. In this problem, a set of the real time requests arrives randomly over time and the dispatcher does not have any deterministic or probabilistic information on the location and size of them until they arrive. Moreover, this model has considered the customers' satisfaction in routing of vehicles, and this preference information was represented as a convex fuzzy number. In addition, the DVRPFTW has been considered as a multi-objective problem (MOP), in which the total required fleet size, overall total traveling distance and waiting time imposed on vehicles are minimized and the overall customers' preferences for service are maximized. A solving strategy, based on a genetic algorithm (GA), consisting of three basic modules has been proposed, in which the state of the system including information of vehicles and customers each time has been checked in the management module. The strategy module organized the information reported by the management module and constructed an efficient structure for solving in the subsequent module.

The performance of the proposed approach has been evaluated in different steps on various test problems generalized from the Solomon's VRPTW benchmark problems. In the first step, the performance of the proposed has been checked in static conditions, and then the other assumptions and developments have been added gradually and changes have been examined. Based on the obtained results, the categories of C1 and C2 have had the correlating objectives positively when the first two objectives have been considered. Moreover, the conflicting behaviors have been more in R and RC categories and most of these instances have had the conflicting objectives in a population distribution of them. It should be noted when the degree of dynamism is increased, a mix between real-time insertion of new requests and re-optimization driven by batching strategies should be considered. Moreover, when a dispatcher has been faced with systems with highly degree of dynamic, due to limited time between two consecutive decision stages, just real-time insertion could be used for saving time. Thus, whenever a new request is received, the algorithm should try to find a feasible spot to insert the new customer without re-scheduling the customers who are already in the solution. Following this policy, the quality of solutions is decreased and the dispatcher will instantly be able either to accept or reject new customers calling in for service.

Eventually, an interesting topic for future research can be evaluating the impact of future information on the solution quality. In the proposed approach, there is no future information available, and hence at each decision epoch available information up to that point of time is only used. Now if there is some information available regarding future orders (e.g., probability distribution of call-in times), the new approach by the combination of real-time optimization methods and a-priori optimization-based methods is emerged. Then the question is how one can incorporate such information into the static problem at each decision epoch to get a better overall solution.

In the proposed approach, the decision epochs are assumed to be fixed and known in advance in our current approach. It will be interesting to see what happens if the decision epochs are determined over time based on the arrival process of orders. This is an interesting topic and could affect the efficiency of proposed approach. Another important issue in the proposed dynamic setting is computational time. When the number of demand nodes increases, a larger search space for the genetic algorithm is generated and the computational time is increased. If larger problems must be solved faster we need to improve the efficiency of computation. Further research is needed to accomplish this. So a future goal is to generate larger problem instances, and further evaluate the GA's performance on these problems.

References

[1] R. Dondo, J. Cerda, A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows, Eur. J. Oper. Res. 176 (2007) 1478–1507.
[2] P. Kilby, P. Prosser, P. Shaw, Dynamic VRPs: s study of scenarios, in: Technical Report APES-0-1998, University of Strathclyde, Glasgow, 1998.
[3] A.S. Pepin, G. Desaulniers, A. Herts, D. Huisman, A comparison of five heuristics for the multiple depot vehicle scheduling problem, J. Sheduling 12 (2009) 17–30.
[4] S. Yu, Ch. Ding, K. Zhu, A hybrid GA-TS algorithm for open vehicle routing optimization of coal mines material, Exp. Syst. Appl. 38 (2011) 10568–10573.
[5] B. Ombuki, B. Ross, F. Hanshar, Multi-objective genetic algorithm for vehicle routing problem with time windows, Appl. Intell. 24 (2006) 17–30.
[6] K.C. Tan, Y.H. Chew, L.H. Lee, A hybrid multiobjective evolutionary algorithm for solving vehicle routing problem with time windows, Comput. Optim. Appl. 34 (2006) 115–151.
[7] K. Ghoseiri, S.F. Ghannadpour, Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm, Appl. Soft Comput. 4 (2010) 1096–1107.
[8] B. Eksioglu, A.V. Vural, A. Reisman, The vehicle routing problem: a taxonomic review, Comput. Ind. Eng. 57 (2009) 1472–1483.
[9] F. Li, B. Golden, E. Wasil, Very large scale vehicle routing: new test problems algorithms and results, Comput. Oper. Res. 32 (2005) 1165–1179.
[10] J. Tanga, Z. Pan, R.Y.K. Fung, H. Lau, Vehicle routing problem with fuzzy time windows, Fuzzy Set. Syst. 160 (2009) 683–695.
[11] A. Larsen, The Dynamic Vehicle Routing Problem, Ph.D. thesis IMM-PHD-2000-73. Department of Informatics and Mathematical Modeling, Technical University of Denmark, Denmark, 2000.
[12] G. Godfrey, W.B. Powell, An adaptive dynamic programming algorithm for dynamic fleet management, I: Single period travel times, Transport. Sci. 36 (2002) 21–39.
[13] R.W. Bent, P. Van Hentenryck, Scenario-based planning for partially dynamic vehicle routing with stochastic customers, Oper. Res. 52 (2004) 977–987.

[14] A. Larsen, O.B.G. Madsen, M.M. Solomon, The a-priori dynamic traveling salesman problem with time windows, Transport. Sci. 38 (2004) 459–572.
[15] H. Lei, G. Laporte, B. Guo, The capacitated vehicle routing problem with stochastic demands and time windows, Comput. Oper. Res. 38 (2011) 1775–1783.
[16] A. Juan, J. Faulin, S. Garsman, D. Riera, J. Marull, C. Mendez, Using safety stocks and simulation to solve the vehicle routing problem with stochastic demands, Transport. Res. C: Emerg. Technol. 19 (2011) 751–765.
[17] G. Berbeglia, J. Cordeau, G. Laporte, Dynamic pickup and delivery problems, Eur. J. Oper. Res. 202 (2009) 8–15.
[18] A. Núñez, B.D. Schutter, D. Sáez, C.E. Cortés, Hierarchical multiobjective model predictive control applied to a dynamic pickup and delivery problem, in: Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems (ITSC 2010), 2010, pp. 1553–1558, Madeira Island, Portugal.
[19] C.E. Cortés, D. Sáez, A. Núñez, D. Muñoz-Carpintero, Hybrid adaptive predictive control for a dynamic pick-up and delivery problem, Transport. Sci. 43 (2009) 27–42.
[20] D. Sáez, C.E. Cortés, A. Núñez, Hybrid adaptive predictive control for the multi-vehicle dynamic pickup and delivery problem based on genetic algorithms and fuzzy clustering, Comput. Oper. Res. 35 (2008) 3412–3438.
[21] M. Gendreau, F. Guertin, J-Y. Potvin, E. Taillard, Parallel tabu search for real-time vehicle routing and dispatching, Transport. Sci. 33 (1999) 381–390.
[22] Z.L. Chen, H. Xu, Dynamic column generation for dynamic vehicle routing with time windows, Transport. Sci. 40 (2006) 74–88.
[23] M.R. Khouadjia, B. Sarasola, E. Alba, L. Jourdan, E.G. Talbi, A comparative study between dynamic adapted PSO and VNS for the vehicle routing problem with dynamic requests, Appl. Soft Comput. 12 (2012) 1426–1439.
[24] T.C. Du, E.Y. Li, D. Chou, Dynamic vehicle routing for online B2C delivery, Omega 33 (2005) 33–45.
[25] S. Lorini, J-Y. Potvin, N. Zufferey, Online vehicle routing and scheduling with dynamic travel times, Comput. Oper. Res. 38 (2011) 1086–1090.
[26] M.M. Solomon, Algorithms for the vehicle routing and scheduling problems with time window constraints, Oper. Res. 35 (1987) 254–265.
[27] J. Jemai, Kh. Mellouli, A neural-tabu search heuristic for the real time vehicle routing problems, J. Math. Model. Algor. 7 (2008) 161–176.
[28] A. Haghani, S. Jung, A dynamic vehicle routing problem with time-dependent travel times, Comput. Oper. Res. 32 (2005) 2959–2986.
[29] E. Taniguchi, H. Shimamoto, Intelligent transportation system based dynamic vehicle routing and scheduling with variable travel times, Transport. Res. C: Emerg. Technol. 12 (2004) 235–250.
[30] J-Y. Potvin, Y. Xu, I. Benyahia, Vehicle routing and scheduling with dynamic travel time, Comput. Oper. Res. 33 (2006) 1129–1137.
[31] R. Montemanni, L.M. Gambardella, A.E. Rizzoli, A.V. Donati, A new algorithm for a dynamic vehicle routing problem based on ant colony system, J. Combin. Optim. 10 (2005) 327–343.
[32] S. Wohlgemuth, R. Oloruntoba, U. Clausen, Dynamic vehicle routing with anticipation in disaster relief, Soc. Econ. Plann. Sci. 46 (2012) 261–271.
[33] V. Pillac, C. Gueret, A.L. Medaglia, An event-driven optimization framework for dynamic vehicle routing, Decis. Support Syst. 54 (2012) 414–423.
[34] K.C. Tan, C.Y. Cheong, C.K. Goh, Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation, Eur. J. Oper. Res. 177 (2007) 139–813.
[35] K. Ghoseiri, S.F. Ghannadpour, A hybrid genetic algorithm for multi-depot homogeneous locomotive assignment with time windows, Appl. Soft Comput. 10 (2010) 53–65.
[36] L.M. Gambardella, E. Taillard, G. Agazzi, MACS-VRPTW. A multiple ant colony system for vehicle routing problems with time windows, in: D. Corne, M. Dorigo, F. Glover (Eds.), New Ideas in Optimization, 1999, pp. 63–76, London.
[37] A. Garcia-Najera, J.A. Bullinaria, An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows, Comput. Oper. Res. 38 (2011) 287–300.
[38] R. Chevrier, A. Liefooghe, L. Jourdan, C. Dhaenens, Solving a dial-a-ride problem with a hybrid evolutionary multi-objective approach: application to demand responsive transport, Appl. Soft Comput. 12 (2012) 1247–1258.
[39] Q. Jun, J. Wang, B.J. Zheng, A hybrid multi-objective algorithm for dynamic vehicle routing problems, Lect. Notes Comput. Sci. 5103 (2008) 674–681.
[40] H. Tang, M. Hu, Dynamic vehicle routing problem with multiple objectives: solution framework and computational experiments, J. Transport. Res. Board 1923/2005 (2006) 199–207.
[41] M. Camara, J. Ortega, F.D. Toro, Performance measures for dynamic multi-objective optimization, Lect. Notes Comput. Sci. 5517 (2009) 760–767.
[42] M. Farina, K. Deb, P. Amato, Dynamic multiobjective optimization problems: test cases, approximation, and applications, IEEE Trans. Evol. Comput. 8 (2004) 425–442.
[43] M. Hapke, P. Wosloek, Handling Imprecision and Flexible Constraints in Vehicle Routing Problems: Fuzzy Approach, Technical Report PA-005/2003, Poznan University of Technology, Poznań, 2003.
[44] A.E. Eiben, S.K. Smit, parameter tuning for configuring and analyzing evolutionary algorithms, Swarm Evol. Comput. 1 (2011) 19–31.
[45] G. Andreatta, G. Lulli, A multi-period TSP with stochastic regular and urgent demands, Eur. J. Oper. Res. 185 (2008) 122–132.
[46] H.H. Sheng, J.C. Wang, H.H. Hanng, D.C. Yen, Fuzzy measure on vehicle routing problem of hospital materials, Exp. Syst. Appl. 30 (2006) 367–377.
[47] C. Erbao, L. Mingyong, The open vehicle routing problem with fuzzy demands, Exp. Syst. Appl. 37 (2010) 2405–2411.