A Framework for Solving Real-Time Multi-objective VRP

Oren E. Nahum(✉) and Yuval Hadas

Department of Management, Bar-Ilan University, Max ve-Anna Webb Street, 5290002 Ramat Gan, Israel
{oren.nahum,yuval.hadas}@biu.ac.il

Abstract. One of the most important logistics problems in the field of transportation and distribution is the Vehicle Routing Problem (VRP). In general, VRP is concerned with the determination of a minimum-cost set of routes for distribution and pickup of goods for a fleet of vehicles, while satisfying given constraints. Today, most VRPs are set up with a single objective function, minimizing costs, ignoring the fact that most problems encountered in logistics are multi-objective in nature (maximizing customers' satisfaction and so on), and that for both deterministic and stochastic VRPs, the solution is based on a pre-determined set of routes. Technological advancements make it possible to operate vehicles using real-time information. Since VRP is a NP-Hard problem, it cannot be solved to optimality using conventional methods; therefore, the paper presents a heuristic framework for solving the problem. In real-time dynamic problems, a solution is given based on known data, as time progresses, new data are added to the problem, and the initial solution has to be re-evaluated in order to suit the new data. This is usually done at pre-defined time intervals. If the time intervals are small enough, thus, at each time interval the amount of information added is limited. Therefore, the new solution will be similar to the previous one. Due to the fact that the result is a solution set, not a single solution, and one solution is to be selected within a short time window, it is necessary to automatically select a single solution. For that, a framework, based on traditional and evolutionary multi-objective optimization algorithms, which incorporate multi-criteria decision making methods, for solving real-time multi-objective vehicle routing problems is presented.

Keywords: Vehicle routing problems · Multi-objective · Multi-criteria decision making · Real-time

1 Introduction

One of the most important logistics problems in the field of transportation and distribution is the Vehicle Routing Problem (VRP) [6, 61, 62]. In general, VRP is concerned with the determination of a minimum-cost set of routes, usually the shortest ones, for distribution and pickup of goods for a fleet of vehicles, while satisfying given constraints. Since the problem was first introduced by Dantzig and Ramser [7] several extensions to the problem, with different types of "cost" and constraints were developed.

J. Żak et al. (eds.), *Advanced Concepts, Methodologies and Technologies for Transportation and Logistics*, Advances in Intelligent Systems and Computing 572, DOI 10.1007/978-3-319-57105-8_5

As of today, most VRPs are 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-objective in nature. In fact, numerous aspects, such as balancing workloads (time, distance, etc.), can be taken into account simply by adding new objectives [27].

Moreover, traditionally, vehicle routing plans are based on deterministic information about demands, vehicle locations and travel times on the roads. Advancement of the technology in communication systems, the geographic information system (GIS) and the intelligent transportation system (ITS), make it possible to operate vehicles using the real-time information about travel times and the vehicles' locations [17]. What is likely to distinguish most VRPs today from equivalent problems in the past, is that information needed to come up with a set of good vehicle routes and schedules is dynamically revealed to the decision maker [48].

While traditional VRPs have been thoroughly studied, limited research has to date been devoted to multi-objective, real-time management of vehicles during the actual execution of the distribution schedule, in order to respond to unforeseen events that often occur and may deteriorate the effectiveness of the predefined and static routing decisions. Furthermore, in cases when traveling time is a crucial factor, ignoring travel time fluctuations (due to various factors, such as peak hour traveling time, accidents, weather conditions, etc.) can result in route plans that can direct the vehicles into congested urban traffic conditions. Considering time-dependent travel times as well as information regarding demands that arise in real time in solving VRPs can reduce the costs of ignoring the changing environment [21].

One point that was neglected, which its importance intensifies in multi-objective, real-time VRP, is the need for a quick and automated selection of a single solution from the non-dominated solution's set. For that, a framework that combines multi-objective VRP together with multi-criteria decision making (MCDM), is presented and assessed.

The rest of the paper is as follows. Section 2 provides a review on both multi-objective and dynamic VRPs as well as multi-criteria decision making methods. Section 3 describes a framework for solving real-time multi-objective VRPs. Section 4 describes results obtained from case study, and, finally, Sect. 5 concludes the paper.

2 Literature Review

The framework presented in this paper is for solving real-time (or dynamic), multi-objective VRPs. As shell be seen later in the paper, the framework incorporates multi-criteria decision methods while solving the problem. Therefore, the three topics, multi-objective VRP, dynamic VRP and multi- criteria decision methods are reviewed next in this chapter.

2.1 Multi-objective VRP

Most VRPs, frequently used to model real cases, are set up with a single objective (minimizing the cost of the solution), although the majority of the problems encountered in industry, particularly in logistics, are multi-objective in nature. According to Jozefowiez et al. [27], multi-objective VRPs are used mainly in three ways:

1. **Extending classic academic problems** – In this case, the problem definition remains unchanged, and new objectives are added. As an example of such an objective, we can consider the following: (1) *Driver workload* – an extension to VRP in which the balance of tour lengths is considered (to increase the fairness of the solution) [43, 60]. (2) *Customer Satisfaction* – an objective added to VRP with time windows [10] in order to improve customer satisfaction with regard to delivery dates [1, 16, 68].
2. **Generalizing Classic Problems** – Another way to use multi-objective optimization is to generalize a problem by adding objectives instead of one or several constraints and/or parameters [58].
3. **Studying real-life cases** - Multi-objective routing problems are also studied for a specific real-life situation, in which decision makers define several clear objectives that they would like to see optimized [2, 11, 19, 31, 66].

The different objectives studied in the literature can be presented and classified according to the component of the problem with which they are associated [27]. The following is a summary of the most common objectives.

1. **Objectives related to the tour**: (a) **Cost**: Minimizing the cost of the solutions generated is the most common objective, usually for economic reasons; however, other motivations are possible. For instance, in [45, 46], it is done to avoid damaging the product being transported. (b) **Makespan**: Minimizing the makespan ensures some fairness in solutions [30]. (c) **Balance**: Some objectives are designed to even out disparities between the tours [37].
2. **Objectives related to node/arc activity**: Most of the studies dealing with objectives related to node/arc activity involve time windows. Time windows are usually replaced by an objective that minimizes the number of violated constraints [4], the total customer and/or driver's wait time due to earliness or lateness [3, 9, 22], or both [13, 14].
3. **Objectives related to resources**: A common objective is the minimization of the number of vehicles, as in VRP with time windows (usually treated lexicographically) [41]. Goods-related objectives are used to take the nature of the goods into account (merchandise is perishable and we want to avoid its deterioration [45, 46]).

Over the last several years, many techniques have been proposed for solving multi-objective problems. These strategies can be divided into three general categories:

1. **Scalar methods** - The most popular is weighted linear aggregation. For multi-objective VRPs, weighted linear aggregation has been used with specific heuristics [18], local search algorithms [64], and genetic algorithms [65].
2. **Pareto methods -** Pareto methods use the notion of Pareto dominance directly. Pareto methods are used with evolutionary algorithms, local searches, heuristics, and/or exact methods [64, 65].
3. **Methods that belong to neither the first nor the second category** - These non-scalar and non-Pareto methods are based on genetic algorithms, lexicographic strategies, ant colony mechanisms, or specific heuristics [29].

2.2 Dynamic VRP

In many real-life applications relevant data changes during the execution of transportation processes and schedules have to be updated dynamically. Thanks to recent advances in information and communication technologies, vehicle fleets can now be managed in real-time. In this context, Dynamic or real-time VRPs (DVRPs), are becoming increasingly important [49].

The most common source of dynamism in VRP is the online arrival of customer requests during the operation [49].

Travel time variations have been studied by Haghani and Jung [21]; Potvin et al. [47]; Fleischmann et al. [12]; Hu [23]; Hu et al. [24]; Ichoua et al. [26]. Malandraki and Daskin [35], who used a step function for that purpose, while Gendreau et al. [15]; Liao [34] proposed tabu search algorithms for solving the problem.

Finally, some more recent work considers dynamically revealed demands for a set of known customers [44, 55, 56] and vehicle availability [32, 33, 39], in which case the source of dynamism is the possible breakdown of vehicles.

2.3 Multi-criteria Decision Making

In most cases, when solving a multi-objective optimization problem, the result is a set of non-dominated solution (a set in which there is no solution that is better in all objectives from another solution in the set), from which the decision maker (DM) has to choose his preferred alternative.

Multi-criteria decision-making (MCDM) methods, some of which are listed below, are automated methods for selecting a preferred solution given a set of feasible solutions, while having conflicting criteria [8, 67]. MCDM methods also allow assigning the various solutions to different pre-defined classes and ordering them from best to worst [63].

The Max-Min method, for example, can be used when the DM wants to maximize the achievement in the weakest criterion. On the other hand, the Min-Max method can be used when the DM wants to minimize the maximum opportunity loss. Compromise Programming identifies the solution whose distance from the ideal solution (an artificial solution consists of the upper bound, for maximization, of the criteria set) is minimum. The ELECTRE Method [50] compares two alternatives at a time and attempts to eliminate alternatives that are dominated using the outranking relationship. In the first version of this method, the result is a set of alternatives (called the kernel) that can be presented to the DM for selecting the preferred solution. The second version of this method provides a complete rank ordering of the original set of alternatives. The TOPSIS method [25] assumes that the preferred solution should simultaneously be closest to the ideal solution and farthest from the negative-ideal solution (an artificial solution consists of the lower bound, for maximization, of the criteria set). For every solution, TOPSIS calculates an index that combines both its closeness to the positive-ideal solution and its remoteness from the negative-ideal solution. The alternative that maximizes this index value is the preferred alternative. Multi-attribute utility theory (MAUT) [28] is based upon the assumption that every DM tries to optimize a utility function (not necessarily known at the beginning of the decision process). The global utility of an alternative is assessed using a utility function, composed of various criteria. Each criterion is assigned with a marginal utility score by the DM, which in a second phase, is aggregated to the global utility score. Each alternative is evaluated and ranked using the utility function.

Many MCDM methods require the use of relative importance weights of criteria, which are usually proportional to the relative value of unit changes in criteria value functions. A simple and common method for ranking criteria is the "weights from ranks" method. In this method, the DM ranks each criterion, r_i, in order of increasing relative importance (highest ranked criterion is rank as 1.) Next each the weight of criteria is defined as $\lambda_i = (k + r_i + 1) \Big/ \left(\sum_{j=i}^{k} k + r_j + 1 \right)$, when k is the number of criteria. While this method produces an ordinal scale, it not guarantee the correct type of criterion importance because ranking does not capture the strength of preference information [36].

When a large number of criteria are considered, it may be easier for the DM to provide pairwise ranking instead of complete ranking. As an example of such a method, consider the analytic hierarchy process (AHP) proposed by Saaty [51, 53]. With AHP, the decision problem is first structured as hierarchal levels. At the top level is the goal of the problem while subsequent levels represent criteria, sub-criteria, and so on with the last level representing the decision alternatives. Next, value judgments concerning the alternatives with respect to the next higher level sub-criteria are calculated based on available measurements. If measurements are not available, the calculation is made from pairwise comparison. After the value judgments of alternatives have been computed, composite values are determined by finding the weighted average values across all levels of the hierarchy. The analytic network process (ANP), a generalization of the AHP method that deals with dependencies, is another example of MCDM methodology [52]. ANP allows for more complex interrelationships among the decision levels and attributes than AHP. Two-way arrows represent interdependencies among attributes and attribute levels. The directions of the arrows signify dependence.

Arrows emanate from an attribute to other attributes that may influence it. The relative importance or strength of the impacts on a given element is measured on a ratio scale similar to AHP (using pairwise comparisons and judgment). A priority vector may be determined by directly asking the DM for a numerical weight but there may be less consistency, since part of the process of decomposing the hierarchy is to provide better definitions of higher level attributes. The ANP approach is capable of handling interdependence among elements by obtaining the composite weights through the development of a "supermatrix".

3 Posteriori Decision Making Framework for Solving Real-Time Multi-objective Vehicle Routing Problems

In a posteriori framework, a multi-objective algorithm is executed, followed by a decision making algorithm that automatically selects a preferred solution from the solution set.

For a given multi-objective VRP, let's assume that there exists an algorithm for solving the problem. It is then possible to solve the real-time version of this VRP, simple by re-solving the problem as soon as new information is available, using the existing algorithm. However, working in such a way is time consuming, and cannot guarantee that a proper solution will exist when needed. Since in most cases, new information may causes relatively small changes (if it is processed soon enough), it will be ideal if we can update the current solution, so it will reflect the new information, and at the same time, provide an optimal or near optimal solution. Population based algorithms and evolutionary algorithms are well suitable for this task, as the new information can be inserted into the current population, which will be considered as the initial conditions for the new updated problem.

Evolutionary Algorithms belong to the evolutionary computation field of study concerned with computational methods inspired by the process and mechanisms of biological evolution. Evolutionary Algorithms are concerned with investigating computational systems that resemble simplified versions of the processes and mechanisms of evolution, toward achieving the effects of these processes and mechanisms, namely the development of adaptive systems.

This section describes a simple framework, based on evolutionary or population based algorithms, for solving real-time multi-objective VRP. The framework is described using genetic algorithms [38, 57], that are a family of computational models inspired by evolution. These algorithms encode a potential solution to a specific problem on a simple chromosome-like data structure and apply recombination operators to these structures in order to preserve critical information.

An implementation of a genetic algorithm begins with a population of (typically random) chromosomes (solutions). One then evaluates these structures and allocated reproductive opportunities in such a way that these chromosomes which represent a better solution to the target problem are given more chances to 'reproduce' than those chromosomes which are poorer solutions. The 'goodness' of a solution is typically defined with respect to the current population.

Genetic Algorithm processes a number of solutions simultaneously. Hence, in the first step a population having *P* individuals is generated. Next, individual members of the population are evaluated to find the objective function value, which is mapped into a fitness function that computes a fitness value for each member of the population. Three main operators, reproduction, crossover and mutation, are used to create a new population. The purpose of these operators is to create new solutions by selection, combination or alteration of the current solutions that have shown to be good temporary solutions. The new population is further evaluated and tested until termination. Since the problem for which the framework is designed is a multi-objective problem, it is necessary to implement a multi-objective algorithm, such as VEGA [54] and SPEA2 [69], as the base of the framework.

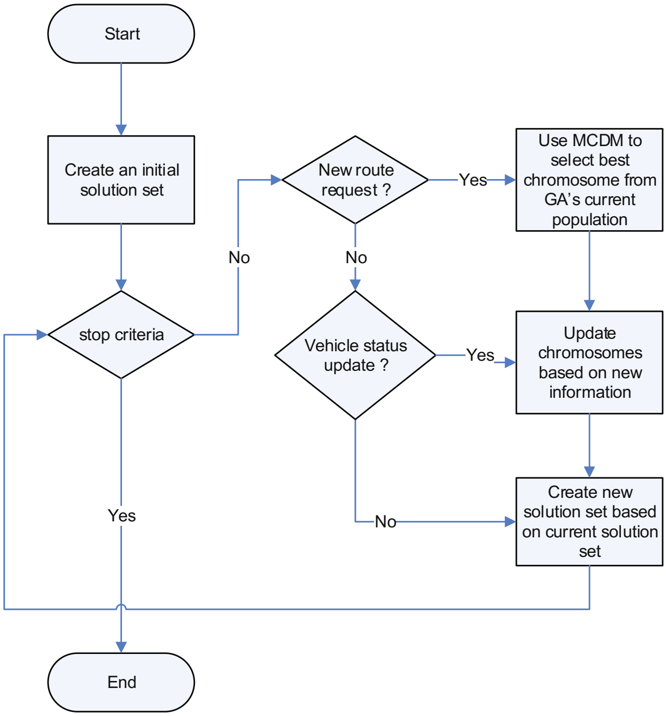

Fig. 1. A'posteriori decision making framework

Figure 1 illustrates the a'posteriori DM framework for solving a real-time multi-objective VRP.

As with any genetic algorithms, the first step of the framework is to generate an initial population (a set of chromosomes or solutions). This population can be created randomly, or using some king of a heuristic. Next, each of the population's chromosomes is evaluated and given a fitness value. The fitness value calculation, is based on the multi-objective genetic algorithm implementation. Next an iterative section is executed until a stopping condition is met. In this framework, the stopping condition is that there are no customers that have to be served and all vehicles are at the depot. The first step of the iterative section is the request route operation. The request route operation selects a single chromosome from the current population, and assigns it routes to the vehicles. The request route operation is executed on two conditions: (1) Current time is defined as departure time. It is then possible that new vehicles have to leave the depot. The request routes operation is used to determine whether new vehicles have to leave the depot and their destinations. (2) A vehicle is at a customer, and the customer has been fully served. The vehicle needs to start its way towards the next customer. A request route operation is performed in order to determine the vehicle's destination, as it can be changed based on current information. Next, all chromosomes are updated based on new information, if any (new routes, customers' demands, travel times, etc.). If a chromosome has been changed, then its fitness must be re-calculated. Next, a new population is generated based on the current population, using crossover and mutation operations. The fitness of each new chromosome in the new population is also calculated.

The request route operation, as mentioned before, selects a single chromosome (or solution) from the current population (set of solution). In most cases, when solving a multi-objective optimization problem, the result is a set of non-dominated solution (a set in which there is no solution that is better in all objectives from another solution in the set), from which the decision maker (DM) has to choose his preferred alternative. This set of non-dominated solution can be obtained using various multi-objective optimization algorithms.

In an automated environment, however, a mechanism for choosing a preferred solution from a set of non-dominated solutions needs to be implemented. In this case, the request route operation uses a MCDM method, such as the ones described in the literature review, in order to rank the solution. Then, based on this ranking, a preferred solution is selected.

4 A Case Study Example

To demonstrate the usage of the posteriori decision making framework, a real-time multi-objective vehicle routing problem has been designed. The full details of the problem, including the description of the various objectives and constraints, the mathematical model and the various multi-objective evolutionary algorithms used for solving it, are outside the scope of this paper, and can be found in [40, 41]. Generally, the problem is defined as a vehicle fleet that has to serve customers of fixed demands from a central depot. Customers must be assigned to vehicles, and the vehicles routed so that a number of objectives are minimized/maximized.

Based on a vast literature review, five objectives were selected: (1) Minimizing the total traveling time [35]; (2) Minimizing the number of vehicles [5]; (3) Maximizing customers' satisfaction [1, 16, 68]; (4) Maximizing drivers' satisfaction [43, 60] and (5) Minimizing the arrival time of the last vehicle.

As a real-time problem, vehicles' routes are adjusted at certain times in a planning period. This adjustment considers new information about the travel times (which is stochastics and depends on the distance between two points and the time of day), current location of vehicles, new demand requests (that can be deleted after being served, or added since they arise after the initial service began) and more. This result in a dynamic change in the demand and traveling time information as time changes, which has to be taken into consideration in order to provide optimized real-time operation of vehicles.

Having several assumptions and limitations, such as a system with dynamic conditions (real-time variation in travel times and real-time service requests); all demands have specified service times and service time intervals; soft time windows for service around the desired service times are considered, and more, Nahum et al. [41] formulated the problem as a mixed integer linear programming problem on a network. However, since the problem is a NP-Hard problem, it cannot be solved to optimality using conventional methods, and therefore, the posteriori decision making framework was used.

The simplicity and generality of the framework makes it possible to use any population based algorithm (as long as it can solve the problem as a static problem) for solving the problem as a real-time problem. For that reason, three evolution algorithms have been developed for solving the problem. The first algorithm is an improved version of the vector evaluated genetic algorithm (VEGA) (which incorporates elitism). The second algorithm is an implementation of the SPEA2 algorithm. And, the third evolutionary algorithm is a combination of the vector evaluated technique and artificial bee colony algorithm. The algorithms were incorporated into the posteriori decision making framework, while the multi-criteria decision making method used by the framework is the TOPSIS method.

The results of the three algorithms were compared using a case study. The case study is based on two transportations networks, each based on real-file information, each with different characteristics. The first network is based on metropolitan Tel-Aviv's urban road network. In this network, there are 45 customers (their locations are based on the stores' locations of "Mega Ba'ir" – a large super-market chain store in Israel) (not including the depot). The second network is based on Israel's interurban road system. In this network, there are 34 customers (not including the depot). The 34 customers are located in major Israeli cities. For both networks, "Google Maps", was used (1) to determine the shortest distance (based on actual network) between every two customers, and (2) to collect traveling time (at different times of the day) for each edge in the network. The traveling time information was later used in order to calculate a log-normal travel-time distribution function for each path [20]. Each customer was also associated with a time window. The time windows were randomly generated according to the following assumptions: (1) The minimum possible time window start time, PSTW, is equal to 8:00 am plus the time it takes to get from the depot to the customer (when leaving the depot at 8:00 am). It is assumed that the distance from the depot to the customer is known and the travel speed is 15 km per hour for the first network and 70 km per hour for the second. (2) The time window start time, STW, is based on possible time window start time and is a random value within the range of PSTW to PSTW + 1.5 (plus one and a half hour). (3) The time window end time, ETW, is based on the time window start time and is a random value within the range of STW + 0.5 to STW + 3. Each customer is also associated with a randomly generated demand, in the range of 10 to 50, similar to the demands used in Solomon's instances [59].

In each test problem, half the customers are considered as customers with unknown demands. These are the customers with the latest time window start time. Each unknown demand is revealed to the simulation at least two hours prior to the beginning of the time window.

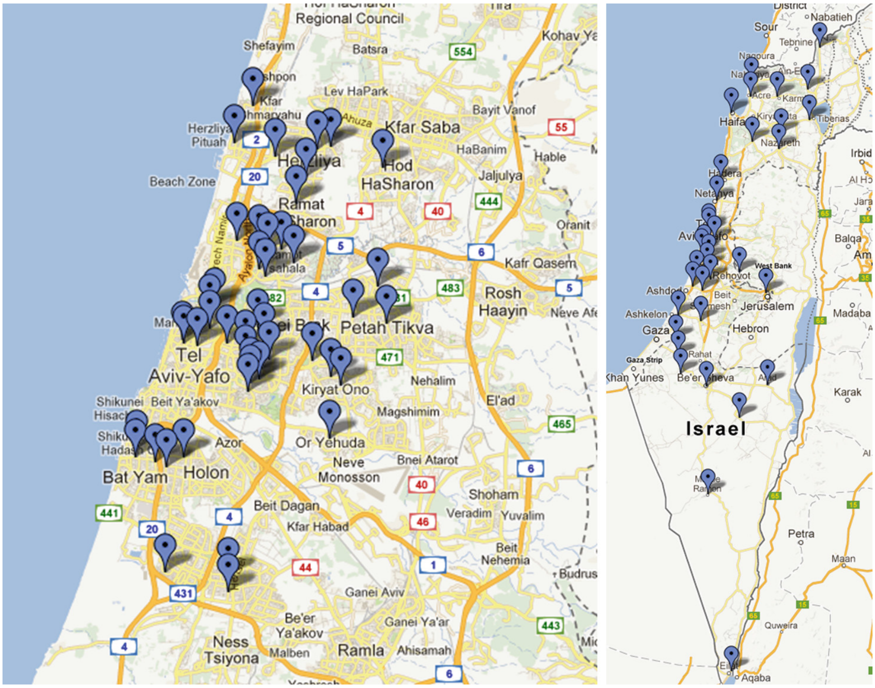

Fig. 2. Urban network (left), having 45 customers in the greater Tel-Aviv metropolitan area, and interurban network (right), having 39 customers in major cities in Israel.

In order to perform the case study, simulation was used. The simulation is based on two processes running in parallel, the algorithm process and the simulation process, which exchange information between each other.

In simulating a full-day of operation, several assumptions are made: (1) The planning period (the time that the algorithm runs before the first vehicle has to leave the depot) starts at 7:00 am and ends at 8:00 am. (2) Service starts at 8:00 am, when the first vehicle leaves the depot, and ends when the last vehicle returns to the depot. (3) During the planning period, new information about customer demands is not acceptable. (4) The workday is divided into 24 time intervals, each one hour long, starting at 0:00 am. (5) For each edge in the transportation network, the travel time is given using log-normal distribution functions for each time interval. (6) Information about real travel times is known two hours in advance (i.e., for the next two time intervals), and is updated 15 min before the beginning of the hour. (7) Every half an hour on the hour, new vehicles that have to leave the depot, leave the depot on their way to their first customers (this can happen due to new customer demands or due to route splitting). (8) New customer demands are acceptable only if there is at least one vehicle who has not completed its route. (9) If all vehicles are either at the depot or driving to the depot, the algorithm stops working (end of the case study). (10) The capacity of a single vehicle is equal to 200 units, as in Solomon's instances.

The simulation process simulates an entire work today. It does so by handling each of the vehicles, collecting data about travel times and new customers' demands.

The three algorithms were compared using a number of case studies, based on the above mentioned real-world transportation networks (urban and interurban), with two different approaches for prioritizing customers' requests (equal or demand size bases priority) and two different customer satisfaction functions.

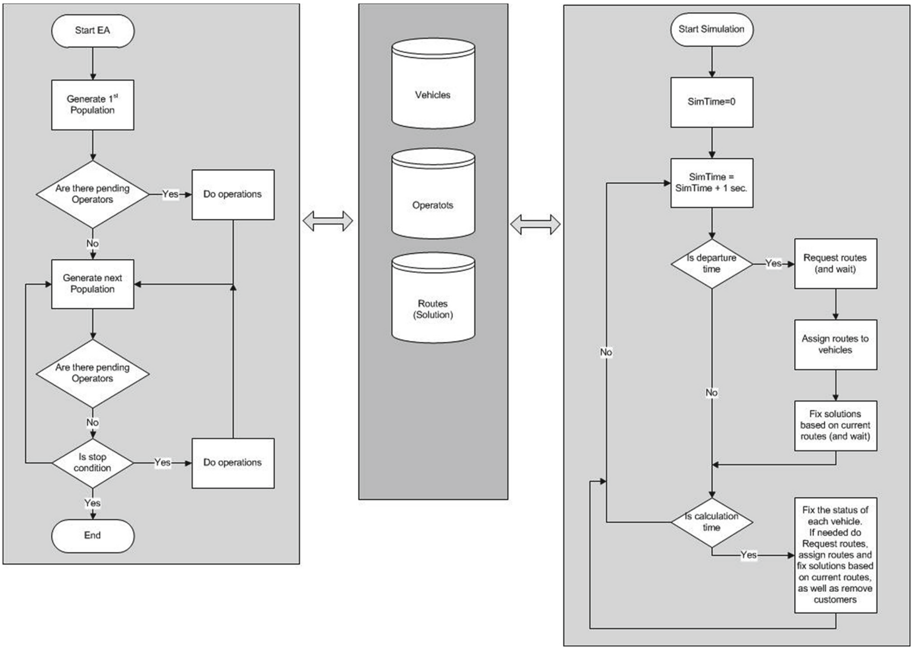

Fig. 3. The relationship between the algorithm process and the simulation process

As an example, during the execution of the SPEA2 algorithm, for the urban network when both travel times and customers' demands are unknown (desired real-world situation) (see [40]), there has been 56 times that a set of routes had to be selected from the current set of solutions provided by the evolutionary algorithm (see Table 1). That means that in average, every four minutes and fifty seconds, the decision maker has to choose from about thirty-four non-dominated solutions. In such a case, the decision maker spends most of his time choosing preferred routes. By incorporating a multi-criteria decision making model into the multi-objective algorithm, the decision maker does not have to carry out the task of selecting preferred routes, which is now performed automatically by the algorithm.

Table 1. Results

Time	Size of pareto front	Routes				
		Total	Unchanged	Changed	New	Removed
07:45:00	44	7	–	–	7	–
08:00:00	54	7	3	4	0	0
08:30:00	84	7	7	0	0	0
08:55:00	89	8	5	2	1	0
08:56:19	92	9	5	3	1	0
09:00:00	42	10	5	4	1	0
09:20:05	87	12	5	5	2	0
09:26:44	10	13	7	5	1	0
09:28:15	35	14	12	1	1	0
09:30:00	19	14	14	0	0	0
09:34:43	72	16	12	2	2	0
09:37:45	79	17	11	5	1	0
09:51:52	55	18	7	10	1	0
09:52:19	37	19	16	2	1	0
09:59:55	57	21	14	5	2	0
10:00:00	11	21	21	0	0	0
10:01:58	60	21	21	0	0	0
10:03:39	87	22	20	1	1	0
10:03:51	16	22	20	1	1	1
10:04:06	17	22	22	0	0	0
10:05:49	15	22	22	0	0	0
10:07:47	20	22	21	1	0	0
10:20:41	64	21	18	3	0	1
10:28:35	72	21	18	3	0	0
10:30:00	43	20	17	3	0	1
10:32:13	74	20	17	3	0	0
10:34:44	48	19	19	0	0	1
10:35:14	78	19	19	0	0	0
10:35:39	45	18	18	0	0	1
10:37:35	52	18	18	0	0	0
10:39:48	48	17	17	0	0	1

(*continued*)

Table 1. (*continued*)

Time	Size of pareto front	Routes				
		Total	Unchanged	Changed	New	Removed
10:40:02	6	18	16	1	1	0
10:47:15	10	17	17	0	0	1
10:48:43	7	16	16	0	0	1
10:48:58	6	15	15	0	0	1
10:49:45	8	15	15	0	0	0
10:54:18	7	14	14	0	0	1
10:54:34	4	13	13	0	0	1
10:55:37	6	12	12	0	0	1
11:00:00	5	11	11	0	0	1
11:01:13	5	11	11	0	0	0
11:03:00	7	10	10	0	0	1
11:05:30	8	10	10	0	0	0
11:07:15	6	9	9	0	0	1
11:13:05	10	8	8	0	0	1
11:16:10	7	7	7	0	0	1
11:22:40	5	7	7	0	0	0
11:24:15	7	6	6	0	0	1
11:30:00	9	5	5	0	0	1
11:42:26	120	5	5	0	0	0
11:45:37	24	4	4	0	0	1
11:57:37	6	3	3	0	0	1
11:57:47	2	2	2	0	0	1
12:00:00	1	1	1	0	0	1
12:06:05	1	1	0	0	0	0

Table 2 is an example of some of the results obtained using the framework, and it shows the results for the fifth strategy (a situation in which both travel times and customers' demands are unknown, a desired real-world situation) for the urban network, using the three evolutionary algorithms with two different customer satisfaction functions (in the first, all customers are equally important, and in the second, the importance of a customer is relative to its demand). For each objective function, the best value is colored in red. From the results, it is clear the best values are either obtained using the Improved VEGA algorithm or the SPEA2 algorithm.

As mentioned, three algorithms were compared using a number of case studies. An analysis of the results obtained using the various algorithms shows that in 70% of the cases, best solutions were obtained using the improved VEGA algorithm. In 75% of the cases, best solutions were obtained using the SPEA2 algorithm. As for the VE-ABC algorithm, it provided the best solutions in 70% of the cases[1]. Furthermore, when

[1] A (best) solution obtained using more than one algorithm, is counted separately for each algorithm.

Table 2. Comparison of the 5th strategy used in all three algorithms

Customer's priority	Objective function	Algorithm		
		Imp. VEGA	SPEA2	VE-ABC
Equal	1	92.834	**82.654**	96.584
	2	**17.021**	18.293	19.226
	3	0.929	**0.454**	2.907
	4	**0.95**	35.596	46.852
	5	21.461	**19.375**	20.881
Demand based	1	**93.123**	98.894	100.742
	2	**18.567**	19.367	18.718
	3	5.356	**0.659**	0.785
	4	23.922	**0.152**	0.463
	5	**20.654**	20.656	20.831

comparing the results of two strategies: non-prioritized customers versus prioritized customers (based on the demand), interesting results are obtained. All the algorithms provide better results for the prioritized strategy on 60% of the cases, with the remaining cases equal for both strategies.

The results show that all three algorithms provide better solutions when each customer is assigned a different priority. The results also show that when the VEGA algorithm is used, it can provide solutions equal in quality to those obtained from more sophisticated and more recent algorithms. This is important, since the VEGA algorithm offers several advantages: its simplicity of implementation; its running speed compared with other algorithms (and as a result, the number of iterations in a given time period); and its capacity for modifications.

5 Summary

This paper presents a framework for solving real-time multi-objective VRP. The framework is based on evolutionary algorithms, which are well suitable for solving this kind of problems, since the previous solution can be considered as an initial solution for the updated problem, while there is no need to start the calculation of the new routes from the beginning. When a driver has to drive to a new customer, the current solution of the algorithm is used in order to define the driver's new destination. Since the result of the algorithm is a set of non-dominated solutions, as in the case of multi-objective, a multi-decision method is used for automatically choose the preferred alternative.

The advantage of the framework is illustrated using a case study. An example based on the case study, shows how frequent a route has to be chosen, and that the number of non-dominated solution from which the route has to be chosen is relatively high. In such a case, the decision maker spends most of his time choosing preferred routes. Incorporating a multi-criteria decision making model into a multi-objective algorithm, automates the process of selecting a preferred route, such that the decision maker handle other tasks.

Using a case study, which was solved using various evolutionary algorithm incorporated into the framework it was shown, based on real information, that all algorithms provide better solutions when each customer is assigned a different priority. Moreover, for the case study it was found that all algorithms provide relatively the same results. This means, that for real world conditions, we can use relatively simple algorithms and still get results similar to state of the art algorithms.

References

1. Afshar-Bakeshloo, M., Mehrabi, A., Safari, H., Maleki, M., Jolai, F.: A green vehicle routing problem with customer satisfaction criteria. J. Ind. Eng. Int. **12**(4), 529–544 (2016)
2. Anbuudayasankar, S.P., Ganesh, K., Lenny Koh, S.C., Ducq, Y.: Modified savings heuristics and genetic algorithm for bi-objective vehicle routing problem with forced backhauls. Expert Syst. Appl. **39**(3), 2296–2305 (2012)
3. Baran, B., Schaerer, M.: A multiobjective ant colony system for vehicle routing problem with time windows. In: Paper presented at the 21st IASTED International Conference on Applied Informatics, Innsbruck, Austria, 10–13 February 2003
4. Barkaoui, M., Berger, J., Boukhtouta, A.: Customer satisfaction in dynamic vehicle routing problem with time windows. Appl. Soft Comput. **35**, 423–432 (2015)
5. Corberan, A., Fernandez, E., Laguna, M., Marti, R.: Heuristic solutions to the problem of routing school buses with multiple objectives. J. Oper. Res. Soc. **53**(4), 427–435 (2002)
6. Cordeau, J.-F., Laporte, G., Savelsbergh, M.W., Vigo, D.: Vehicle routing. In: Handbooks in Operations Research and Management Science, vol. 14, pp. 367–428 (2007)
7. Dantzig, G.B., Ramser, J.H.: The truck dispatching problem. Manage. Sci. **6**(1), 80–91 (1959)
8. Ehrgott, M.: Multicriteria Optimization. Springer, Heidelberg (2005)
9. El-Sherbeny, N.: Resolution of a vehicle routing problem with multi-objective simulated annealing method. Ph.D. Dissertation. Faculte Polytechnique de Mons (2001)
10. El-Sherbeny, N.A.: Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods. J. King Saud Univ. Sci. **22**(3), 123–131 (2010)
11. Faccio, M., Persona, A., Zanin, G.: Waste collection multi objective model with real time traceability data. Waste Manag. **31**(12), 2391–2405 (2011)
12. Fleischmann, B., Gnutzmann, S., Sandvoß, E.: Dynamic vehicle routing based on online traffic information. Transp. Sci. **38**(4), 420–433 (2004)
13. Geiger, M.J.: Genetic algorithms for multiple objective vehicle routing. In: Paper Presented at the MIC 2001-4th Metaheuristics International Conference, Porto, Portugal (2001)
14. Geiger, M.J.: Genetic algorithms for multiple objective vehicle routing (2008). Arxiv preprint arXiv:0809.0416
15. Gendreau, M., Guertin, F., Potvin, J.-Y., Taillard, É.: Parallel tabu search for real-time vehicle routing and dispatching. Transp. Sci. **33**(4), 381–390 (1999)
16. Ghannadpour, S.F., Noori, S., Tavakkoli-Moghaddam, R., Ghoseiri, K.: A multi-objective dynamic vehicle routing problem with fuzzy time windows: Model, solution and application. Appl. Soft Comput. **14**, 504–527 (2014)
17. Ghiani, G., Guerriero, F., Laporte, G., Musmanno, R.: Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies. Eur. J. Oper. Res. **151**(1), 1–11 (2003)

18. Gong, Y.-J., Zhang, J., Liu, O., Huang, R.-Z., Chung, H.S.-H., Shi, Y.-H.: Optimizing the vehicle routing problem with time windows: A discrete particle swarm optimization approach. IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) **42**(2), 254–267 (2012)
19. Gupta, R., Singh, B., Pandey, D.: Multi-objective fuzzy vehicle routing problem: A case study. Int. J. Contemp. Math. Sciences **5**(29), 1439–1454 (2010)
20. Hadas, Y., Ceder, A.: Improving bus passenger transfers on road segments through online operational tactics. Transp. Res. Rec. **2072**(2072), 101–109 (2008)
21. Haghani, A., Jung, S.: A dynamic vehicle routing problem with time-dependent travel times. Comput. Oper. Res. **32**(11), 2959–2986 (2005)
22. Hong, S.-C., Park, Y.-B.: A heuristic for bi-objective vehicle routing with time window constraints. Int. J. Prod. Econ. **62**(3), 249–258 (1999)
23. Hu, T.Y.: Evaluation framework for dynamic vehicle routing strategies under real-time information. Artif. Intell. Intell. Transp. Syst. **1774**(1774), 115–122 (2001)
24. Hu, T.Y., Liao, T.Y., Lu, Y.C.: Study of solution approach for dynamic vehicle routing problems with real-time information. Transp. Netw. Model. **1857**, 102–108 (2003)
25. Hwang, C.L., Yoon, K.: Multiple Attribute Decision Making: Methods and Applications: A State-of-the-Art Survey, vol. 13. Springer, New York (1981)
26. Ichoua, S., Gendreau, M., Potvin, J.Y.: Vehicle routing with time-dependent travel times. Eur. J. Oper. Res. **144**, 379–396 (2003)
27. Jozefowiez, N., Semet, F., Talbi, E.-G.: Multi-objective vehicle routing problems. Eur. J. Oper. Res. **189**(2), 293–309 (2008)
28. Keeney, R., Raiffa, H.: Decisions with multiple objectives: preferences and value tradeoffs. Interfaces **7**(4), 115–117 (1977)
29. Kovacs, A.A., Parragh, S.N., Hartl, R.F.: The multi-objective generalized consistent vehicle routing problem. Eur. J. Oper. Res. **247**(2), 441–458 (2015)
30. Lacomme, P., Prins, C., Prodhon, C., Ren, L.: A multi-start split based path relinking (MSSPR) approach for the vehicle routing problem with route balancing. Eng. Appl. Artif. Intell. **38**, 237–251 (2015)
31. Lacomme, P., Prins, C., Sevaux, M.: A genetic algorithm for a bi-objective capacitated arc routing problem. Comput. Oper. Res. **33**(12), 3473–3493 (2006)
32. Li, J.Q., Mirchandani, P.B., Borenstein, D.: A Lagrangian heuristic for the real-time vehicle rescheduling problem. Transp. Res. Part E: Logistics Transportation Rev. **45**(3), 419–433 (2009)
33. Li, J.Q., Mirchandani, P.B., Borenstein, D.: Real-time vehicle rerouting problems with time windows. Eur. J. Oper. Res. **194**(3), 711–727 (2009)
34. Liao, T.Y.: A tabu search algorithm for dynamic vehicle routing problems under real-time information. J. Transp. Res. Board **1882**, 140–149 (2004)
35. Malandraki, C., Daskin, M.S.: Time dependent vehicle routing problems: Formulations, properties and heuristic algorithms. Transp. Sci. **26**(3), 185–200 (1992)
36. Masud, A.S.M., Ravindran, A.R.: Multiple criteria decision making. In: Ravindran, A.R. (ed.) Operations Research and Management Science. Handbook: Taylor & Francis Group, LLC (2008)
37. Matl, P., Hartl, R.F., Vidal, T.: Equity Objectives in Vehicle Routing: A Survey and Analysis (2016). arXiv preprint arXiv:1605.08565
38. Mitchell, M.: An Introduction to Genetic Algorithms. MIT Press, Cambridge (1996)
39. Mu, Q., Fu, Z., Lysgaard, J., Eglese, R.W.: Disruption management of the vehicle routing problem with vehicle breakdown. J. Oper. Res. Soc. **62**(4), 742–749 (2010)
40. Nahum, O.E.: The Real-Time Multi-Objective Vehicle Routing Problem. (Ph.D. Thesis), Bar-Ilan University, Ramat-Gan, Israel (2013)

41. Nahum, O.E., Hadas, Y., Spiegel, U.: Multi-objective vehicle routing problems with time windows: A vector evaluated artificial bee colony approach. Int. J. Comput. Inf. Technol. **3** (1), 41–47 (2014)
42. Nahum, O.E., Hadas, Y., Spiegel, U., Cohen, R.: The real-time multi-objective vehicle routing problem - case study: Information availability and the quality of the result. In: Paper presented at the Transportation Research Board (TRB) 93rd Annual Meeting, Washington DC, USA (2014)
43. Norouzi, N., Tavakkoli-Moghaddam, R., Ghazanfari, M., Alinaghian, M., Salamatbakhsh, A.: A new multi-objective competitive open vehicle routing problem solved by particle swarm optimization. Netw. Spat. Econ. **12**(4), 609–633 (2012)
44. Novoa, C.M., Storer, R.: An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. Eur. J. Oper. Res. **196**(2), 509–515 (2009)
45. Park, Y.B., Koelling, C.P.: A solution of vehicle routing problems in a multiple objective environment. Eng. Costs Prod. Econ. **10**(2), 121–132 (1986)
46. Park, Y.B., Koelling, C.P.: An interactive computerized algorithm for multicriteria vehicle routing problems. Comput. Ind. Eng. **16**(4), 477–490 (1989)
47. Potvin, J.Y., Xu, Y., Benyahia, I.: Vehicle routing and scheduling with dynamic travel times. Comput. Oper. Res. **33**(4), 1129–1137 (2006)
48. Psaraftis, H.N.: Dynamic vehicle routing: Status and prospects. Ann. Oper. Res. **61**(1), 143–164 (1995)
49. Ritzinger, U., Puchinger, J., Hartl, R.F.: A survey on dynamic and stochastic vehicle routing problems. Int. J. Prod. Res. **54**(1), 215–231 (2016)
50. Roy, B.: The outranking approach and the foundations of ELECTRE methods. Theor. Decis. **31**(1), 49–73 (1991)
51. Saaty, T.L.: A scaling method for priorities in hierarchical structures. J. Math. Psychol. **15** (3), 234–281 (1977)
52. Saaty, T.L.: Analytic network process. Encyclopedia of Operations Research and Management Science, pp. 28–35. Springer, Heidelberg (2001)
53. Saaty, T.L.: Decision making with the analytic hierarchy process. Int. J. Serv. Sci. **1**(1), 83–98 (2008)
54. Schaffer, J.D.: Multi-objective optimization with vector evaluated genetic algorithms. In: Paper presented at the 1st International Conference on Genetic Algorithms, Carnegie-Mellon University, Pittsburgh, USA (1985)
55. Secomandi, N.: Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands. Comput. Oper. Res. **27**(11), 1201–1225 (2000)
56. Secomandi, N., Margot, F.: Reoptimization approaches for the vehicle-routing problem with stochastic demands. Oper. Res. **57**(1), 214–230 (2009)
57. Sivanandam, S.N., Deepa, S.N.: Introduction to Genetic Algorithms. Springer, Heidelberg (2007)
58. Sivaramkumar, V., Thansekhar, M., Saravanan, R., Amali, S.M.J.: Multi-objective vehicle routing problem with time windows: improving customer satisfaction by considering gap time. In: Proc. Inst. Mech. Eng. Part B: J. Eng. Manuf. 1–6 (2015)
59. Solomon, M.M.: Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper. Res. **35**(2), 254–265 (1987)
60. Tavakkoli-Moghaddam, R., Alinaghian, M., Salamat-Bakhsh, A., Norouzi, N.: A hybrid meta-heuristic algorithm for the vehicle routing problem with stochastic travel times considering the driver's satisfaction. J. Ind. Eng. Int. **8**(1), 1–6 (2012)
61. Toth, P., Vigo, D.: The Vehicle Routing Problem. SIAM, Philadelphia (2001)
62. Toth, P., Vigo, D.: Vehicle Routing: Problems, Methods and Applications. SIAM, Philadelphia (2014)

63. Vincke, P.: Multicriteria Decision-Aid. Wiley, New York (1992)
64. Wang, C., Zhao, F., Mu, D., Sutherland, J.W.: Simulated annealing for a vehicle routing problem with simultaneous pickup-delivery and time windows. In: Prabhu, V., Taisch, M., Kiritsis, D. (eds.) APMS 2013. IAICT, vol. 415, pp. 170–177. Springer, Heidelberg (2013). doi:10.1007/978-3-642-41263-9_21
65. Wang, H.-F., Chen, Y.-Y.: A genetic algorithm for the simultaneous delivery and pickup problems with time window. Comput. Ind. Eng. **62**(1), 84–95 (2012)
66. Wen, M., Cordeau, J.-F., Laporte, G., Larsen, J.: The dynamic multi-period vehicle routing problem. Comput. Oper. Res. **37**(9), 1615–1623 (2010)
67. Żak, J., Kruszyński, M.: Application of AHP and ELECTRE III/IV methods to multiple level, multiple criteria evaluation of urban transportation projects. Transp. Res. Procedia **10**, 820–830 (2015)
68. Zhang, J.L., Wang, W.L., Zhao, Y.W., Cattani, C.: Multiobjective quantum evolutionary algorithm for the vehicle routing problem with customer satisfaction. Math. Probl. Eng. **2012**, 1–19 (2012)
69. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength pareto evolutionary algorithm. In: Paper Presented at the Eurogen (2001)