Applied Soft Computing 10 (2010) 1096–1107

Contents lists available at ScienceDirect

Applied Soft Computing

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

Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm

Keivan Ghoseiri [a,b,*], Seyed Farid Ghannadpour [a]

[a] School of Railway Engineering, Iran University of Science and Technology, 16846-13114, Iran
[b] Department of Civil and Environmental Engineering, University of Maryland, College Park, MD 20742-3021, USA

ARTICLE INFO

Article history:
Received 29 April 2008
Received in revised form 11 February 2010
Accepted 2 April 2010
Available online 14 April 2010

Keywords:
Vehicle routing problem with time windows (VRPTW)
Goal programming (GP)
Genetic algorithm
Multiple objective optimization
Pareto ranking

ABSTRACT

This paper presents a new model and solution for multi-objective vehicle routing problem with time windows (VRPTW) using goal programming and genetic algorithm that in which decision maker specifies optimistic aspiration levels to the objectives and deviations from those aspirations are minimized. VRPTW involves the routing of a set of vehicles with limited capacity from a central depot to a set of geographically dispersed customers with known demands and predefined time windows. This paper uses a direct interpretation of the VRPTW as a multi-objective problem where both the total required fleet size and total traveling distance are minimized while capacity and time windows constraints are secured. The present work aims at using a goal programming approach for the formulation of the problem and an adapted efficient genetic algorithm to solve it. In the genetic algorithm various heuristics incorporate local exploitation in the evolutionary search and the concept of Pareto optimality for the multi-objective optimization. Moreover part of initial population is initialized randomly and part is initialized using Push Forward Insertion Heuristic and λ-interchange mechanism. The algorithm is applied to solve the benchmark Solomon's 56 VRPTW 100-customer instances. Results show that the suggested approach is quiet effective, as it provides solutions that are competitive with the best known in the literature.

1. Introduction

Vehicle routing problem with time windows (VRPTW) is a variant of vehicle routing problem (VRP) with adding time windows constraints to the model. VRP is one of the most attractive topics in operation research and deals with determination of the least cost routes from a central depot to a set of geographically dispersed customers. Vehicle routing problems (VRPs) are well known combinatorial optimization problems arising in transportation logistic that usually involve scheduling in constrained environments. In transportation management, there is a requirement to provide goods and/or service from a supply point to various geographically dispersed points with significant economic implications. Because of many applications of different kinds of VRP, many researchers have focused to develop solution approaches for these problems. Likewise, useful techniques for the general VRP could be found in [10,15,38,43].

In VRPTW, a set of vehicles with limited capacity is to be routed from a central depot to a set of geographically dispersed customers with known demands and predefined time windows in order that fleet size of vehicles and total traveling distance are minimized and capacity and time windows constraints are not violated. Due to its inherent complexities and usefulness in real life, the VRPTW continues to draw attention from researchers and has become a well-known problem in network optimization, so many authors developed different solution approaches based on exact and heuristics methods. In terms of exact algorithms [41], presented modern branch and cut techniques for routing problems. Also exact approach using Dantzig–Wolfe decomposition for the VRPTW can be found in [26]. Kohl's work [42] is one of the most efficient exact methods for the VRPTW; it succeeded in solving various 100-customer size instances. However, no algorithm has been developed to date that can solve to optimality all VRPTW instances with 100 customers or more. It should be noted that exact methods are more efficient in the situations where the solution space is restricted by narrow time windows; since there are fewer combinations of customers to define feasible routes [35]. Hence, many researchers have investigated the VRPTW using Heuristics and Meta-heuristics approaches.

These approaches seek for approximate solutions in polynomial time instead of exact solutions which would be at intolerably high cost. Various heuristic methods may be found in litera-

* Corresponding author at: Department of Civil and Environmental Engineering, University of Maryland, College Park, MD 20742-3021, USA. Tel.: +1 301 405 1971; fax: +1 301 405 2585.
E-mail addresses: kghoseir@umd.edu (K. Ghoseiri), Ghannadpour@rail.iust.ac.ir (S.F. Ghannadpour).

1568-4946/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2010.04.001

ture for VRPTW in [30,48]. In this area, [52,50] solved VRPTW with simulated annealing, [51,12,21] solved VRPTW with tabu search and [35] applied multiple ant colony system for VRPTW, and [7,18,27,19,31,14,32,49] used genetic algorithm for VRPTW. In applying genetic algorithm, [25] presented two new crossover operators, Merge Cross#1 and Merge Cross#2, and showed that the new operators are superior to traditional crossovers operators. A cluster-first, route-second method using genetic algorithm and local search optimization process was implemented by Thangiah [48]. Comparative studies of the performance of genetic algorithm, tabu search and simulated annealing for the VRPTW is given in [30,48]. Other heuristics that have been applied to the VRPTW include constraint programming and local search [4,44]. Other very good techniques and applications of VRPTW can be found in [13,5,33,20,45,3].

Although literature of the VRPTW is rich in exact and heuristics solution approaches and models dealing with a single point of view, e.g., a comprehensive profit index (or a comprehensive cost index) representing the preferability (or dis-preferability) of the considered actions [23,6], not many could be found that deal with multiple objective vehicle routing problem with time windows especially the ones that solve the problem in a polynomial time and take into consideration the opinions of the decision maker in the decision process.

In the multi-objective area, Tan et al. [31] proposed a hybrid multi-objective evolutionary algorithm (HMOEA) that incorporates various heuristics for local exploitation in the evolutionary search and the concept of Pareto's optimality for solving the multi-objective VRPTW optimization. Unlike the works [22,34] that were designed for parameterized problems, [31] featured with specialized genetic operators and variable-length chromosome representation to accommodate the sequence-oriented optimization in VRPTW and produced very good result on Solomon's 56 benchmark problems. Another similar study in this area is [7] which employed different and efficient operators to produce good results.

Gambardella et al. [35] studied a type of multi-objective implementation of the VRPTW by minimizing a hierarchical objective function, where the first objective minimized the number of vehicles and the second minimized the total travel time. This was achieved by adapting the ant colony system (ACS) [36]. Another research in the area is [8] that applied a hybrid search based on GA and tabu search to the soft VRPTW. While good results were obtained, the approach was two-phased: a GA was first used to set the number of vehicles, and then a local tabu search employed to minimize the total cost of the distance traveled.

This paper studies a bi-objective VRPTW which is modeled by a goal programming approach and implemented with a genetic algorithm. In this study, simultaneous minimization of the number of vehicles and the total traveling distance are considered as the objective functions. The basic idea in this paper is that the decision maker specifies optimistic aspiration levels to the objective functions of the problem and deviations from these aspiration levels are minimized. The problem is modeled in goal programming and a well-established genetic algorithm is developed and implemented to effectively solve the problem.

Goal programming approach (GP) is the most popular of methodologies for dealing with multi-objective programming problems and is a concept of satisfying the objectives, i.e., to seek a solution that comes as near as possible to the goals. For this propose a target level is established by decision maker which represents his/her preferences in relevant to acceptable level of achievement for each objective. Deviational variables are defined in order to represent the deviation amount from the target value. This model is a branch of multiple objective programming, which in turn is a branch of multi-criteria decision analysis (MCDA), also known as multiple-criteria decision making (MCDM). It can be thought of as an extension or generalization of linear programming to handle multiple, normally conflicting objective measures. The initial goal programming formulations ordered the unwanted deviations into a number of priority levels, with the minimization of a deviation in a higher priority level being of infinitely more important than any deviations in lower priority levels. This is known as *lexicographic* or pre-emptive goal programming.

Lexicographic goal programming should be used when there is a clear priority ordering amongst the goals to be achieved. If the decision maker is more interested in direct comparisons of the objectives then *Weighted* or non-pre-emptive goal programming should be used. In this case all the unwanted deviations are multiplied by weights, reflecting their relative importance, and added together as a single sum to form the achievement function. Goal programming was first used by Charnes et al. [1] in 1955, although the actual name first appeared in a 1961 text by Charnes and Cooper [2]. Seminal works by [47,28,9] were followed. Scniederjans [39] gave in a bibliography of a large number of pre-1995 articles relating to goal programming and Jones and Tamiz gave an annotated bibliography of the period 1990–2000 [11].

A related work in the area is the research conducted by Calvete et al. [16] that applied the goal programming on a single objective vehicle routing problem with soft time windows (VRPSTW). To solve the model, an enumeration-followed-by-optimization approach was proposed which first computed feasible routes and then selected the set of best ones. Computational results showed that this approach was adequate for medium-sized delivery problems.

The VRPTW specification requires a minimization of both the number of vehicles and total distance traveled. From a theoretical point of view, this may be impossible to realize, because instances of the VRPTW may have many non-dominated solutions. Some solutions may minimize the number of vehicles at the expense of distance, and others minimize distance while necessarily increasing the vehicle count. If one scans the literature, however, most researchers clearly place priority on minimizing the number of vehicles. Although this might be reasonable in some instances, it is not inherently preferable over minimizing distance. Minimizing the number of vehicles affects vehicle and labor costs, while minimizing distance affects time and fuel resources. Therefore, the VRPTW is intrinsically a Multiple Objective Optimization (MOP) problem in nature, and it recognizes these alternative solutions [7].

The remaining parts of the paper are organized as follows. Section 2 describes the problem and presents the goal programming formulation of the problem. Section 3 introduces the multi-objective genetic search algorithm to solve the problem and Section 4 describes computational experiments carried out to investigate the performance of the proposed GA. Section 5 provides the concluding remarks.

2. Model formulation

The vehicle routing problem with time windows (VRPTW) is given by a special node called depot, a set of customer C to be visited and a directed network connecting the depot and the customers. Also homogeneous fleet of vehicles is available. They are located at the depot, so they must leave from and return to the central depot. It is assumed that there is no limitation on the number of vehicles that can be used, but in order to facilitate the model formulation the maximum possible size of the fleet is denoted by K. The actual number of vehicles will be found after solving the model that it would be equal to the number of routes in the traffic network. Let us assume there are $N+1$ customers, $C=\{0, 1, 2, \ldots, N\}$ and for simplicity, depot is denoted as customer 0.

Each arc in the network corresponds to a connection between two arcs. A route is defined as starting from depot, going through a number of customers and ending at the depot. A distance d_{ij} and travel time t_{ij} are associated with each arc of the network. Every customer in the network must be visited only once by one of the vehicles. Since each vehicle has a limited capacity q_k ($k = \{1, \ldots, K\}$), and each customer has a varying demand m_i, q_k must be greater than or equal to the summation of all demands on the route traveled by that vehicle k. On the other hand, any customer i must be serviced within a predefined time interval $[e_i, l_i]$, limited by an earliest arrival time (e_i) and latest arrival time (l_i). Vehicles arriving later than the latest arrival time are penalized while those arriving earlier than the earliest arrival time incur waiting. Assuming waiting time is permitted at no cost, it is assumed that $e_0 = l_0 = 0$; that is, all routes start at time 0. Vehicles are also supposed to complete their individual routes within the total route time which is essentially the time window of the depot.

The model has two types of decision variables. For each arc (i, j), where $i \neq j$, $i, j \neq 0$, and each vehicle k, the decision variable x_{ijk} is equal to 1 if vehicle k drives from vertex i to vertex j and 0 otherwise. The decision variable a_{ik} denotes the arrival time of each vehicle k at node i. In order to formulate the model, other following notations are defined:

f_{ik} = service time for vehicle k at node i
w_{ik} = waiting time for vehicle k at node i
r_k = maximum route time allowed for vehicle k
z_{0k} = departure time of vehicle k from the central depot

The objective of the VRPTW is to serve all the C customers such that the following objectives are met and the following constraints are satisfied.

Objectives

- Minimize the distance traveled by the vehicles.
- Minimize the total number of vehicles used to serve the customers.

Constraints

- Vehicle capacity constraints are observed.
- Time window constraints are observed.
- Each customer is served exactly once.
- Each vehicle starts its journey from depot and ends at the depot.

Therefore, after establishment of target levels which represent optimistic aspiration levels for each objective, this multi-objective problem is formulated as a goal programming model. Hence, the following goals, in accordance with the above mentioned objectives, are defined:

$$\mathit{Goal}(1): \quad \sum_{i=0}^{N}\sum_{j=0, j\neq i}^{N}\sum_{k=1}^{K} d_{ij}x_{ijk} \leq TL_1 \tag{1}$$

$$\mathit{Goal}(2): \quad \sum_{k=1}^{K}\sum_{j=1}^{N} x_{ijk} \leq TL_2 \quad \text{for} \quad i = 0 \tag{2}$$

$$\mathit{Subject\ to}: \quad \vec{x} \in \Delta \quad (\text{feasible space}) \tag{3}$$

The first and second goals in relations (1) and (2) indicate the aspiration levels of objectives (1) and (2). TL_1 and TL_2 are the target values associated with the desired levels of the objectives (1) and (2) which control the distance traveled by vehicles and the total number of vehicles used to serve the customers.

In order to treat the goals, two mathematical equations (5) and (6) are written and added to the constraints using the non-negative deviational variables (B_i and P_i for $i = 1, 2$) which measure the deviation from the target values. Therefore the objective of the model is to minimize the undesirable deviations P_1 and P_2.

Given the above defined goals, target levels, deviational variables and decision variables, the problem is formulated as follows:

MOV-GP (I):

$$\begin{aligned} &\mathit{Minimize}\, P_1 \\ &\mathit{Minimize}\, P_2 \\ &\mathit{Subject\ to}: \end{aligned} \tag{4}$$

$$\sum_{i=0}^{N}\sum_{j=0, j\neq i}^{N}\sum_{k=1}^{N} d_{ij}x_{ijk} - P_1 + B_1 = TL_1 \tag{5}$$

$$\sum_{k=1}^{K}\sum_{j=1}^{N} x_{ijk} - P_2 + B_2 = TL_2 \quad \text{for}\, i = 0 \tag{6}$$

$$\sum_{j=1, j\neq i}^{N} x_{ijk} = \sum_{j=1, j\neq i}^{N} x_{jik} \leq 1 \quad \text{for}\, i = \{0, \ldots, N\}\, \text{and}\, k = \{1, \ldots, K\} \tag{7}$$

$$\sum_{k=1}^{K}\sum_{j=0, j\neq i}^{N} x_{ijk} = 1 \quad \text{for}\, i = \{1, \ldots, N\} \tag{8}$$

$$\sum_{k=1}^{K}\sum_{i=0, i\neq j}^{N} x_{ijk} = 1 \quad \text{for}\, j = \{1, \ldots, N\} \tag{9}$$

$$\sum_{i=1}^{N} m_i \sum_{j=0, j\neq i}^{N} x_{ijk} \leq q_k \quad \text{for}\, k = \{1, \ldots, K\} \tag{10}$$

$$\sum_{i=0}^{N}\sum_{j=0, j\neq i}^{N} x_{ijk}(t_{ij} + f_{ik} + w_{ik}) \leq r_k \quad \text{for}\, k = \{1, \ldots, K\}. \tag{11}$$

$$z_{0k} = w_{0k} = f_{0k} = 0 \quad \text{for}\, k = \{1, \ldots, K\} \tag{12}$$

$$\sum_{k=1}^{K}\sum_{i=0, i\neq j}^{N} x_{ijk}(a_{ik} + t_{ij} + f_{ik} + w_{ik}) \leq a_{jk} \quad \text{for}\, j = \{i, \ldots, N\} \tag{13}$$

$$e_i \leq (a_{ik} + w_{ik}) \leq l_i \quad \text{for}\, i = \{1, \ldots, N\},\ k = \{1, \ldots, K\} \tag{14}$$

$$x_{ijk} \in \{0, 1\} \quad \text{for}\, i, j = \{1, \ldots, N\} \tag{15}$$

$$a_{ik}, P_1, P_2, B_1, B_2 \geq 0 \quad \text{for}\, i = 1\{1, \ldots, N\},\ k = \{1, \ldots, K\} \tag{16}$$

Formula (4) at MOV-GP (I) formulation minimizes the undesired deviations from the aspiration level of the objectives. If decision maker is more interested in direct comparisons of the objectives then weighted goal programming model should be used, so Eq. (4) changes to the following relation:

$$\mathit{Minimize} \quad \sum_{j=1}^{2} \alpha_j P_j \tag{17}$$

where α_j reflects the relative importance of the objectives and denote the penalties per unit of deviation from each goal j. This paper keeps two objectives separately and by the use of Pareto Ranking approach which is described later, brings the objectives together in the solution technique simultaneously.

Constraints (5) and (6) refer to goals 1 and 2. They indicate the difference between what is aspired and what is accomplished with respect to the objectives. It does not make any sense for both deviational variables to take non-zero values simultaneously.

Constraint (7) secures every route starts and ends at the central depot. Constraints (8) and (9) define that every customer node is visited only once by one vehicle. Constraint (10) is the capacity constraint. Constraint (11) is the maximum travel time constraint. Constraints (12)–(14) define the time windows.

For simplifying the MOV-GP (I) formulation; those constraints involving deviational variables are rewritten as Eqs. (18) and (19) respectively:

$$d_1 = \max\left(0, \sum_{i=0}^{N}\sum_{j=0, j\neq i}^{N}\sum_{k=1}^{K} d_{ij}x_{ijk} - TL_1\right) \tag{18}$$

$$d_2 = \max\left(0, \sum_{k=1}^{K}\sum_{j=1}^{N} x_{ijk} - TL_2\right) \tag{19}$$

Constraint (18) yields that $d_1 = 0$ if $\left(\sum_{i=0}^{N}\sum_{j=0, j\neq i}^{N}\sum_{k=1}^{K} d_{ij}x_{ijk} - TL_1\right) \leq 0$, otherwise $d_1 = \sum_{i=0}^{N}\sum_{j=0, j\neq i}^{N}\sum_{k=1}^{K} d_{ij}x_{ijk} - TL_1$. Likewise, Constraint (19) yields that $d_2 = 0$ if $\left(\sum_{k=1}^{K}\sum_{j=1}^{N} x_{ijk} - TL_2\right) \leq 0$, otherwise $d_2 = \sum_{k=1}^{K}\sum_{j=1}^{N} x_{ijk} - TL_2$.

MOV-GP (II) problem is produced from MOV-GP (I) with respect to the aforementioned simplifications:

MOV-GP (II):

$$\textit{Minimize} \quad \langle \sum_{i=0}^{N}\sum_{j=0, j\neq i}^{N}\sum_{k=1}^{K} d_{ij}x_{ijk} - TL_1 \rangle \tag{20}$$

$$\textit{Minimize} \quad \langle \sum_{k=1}^{K}\sum_{j=1}^{N} x_{0jk} - TL_2 \rangle \tag{21}$$

$$\textit{Subject to}: \quad \vec{x} \in \Delta\,(\text{Constraints}\,(7)-(15)) \tag{22}$$

Here the bracket operator $\langle\,\rangle$ returns the value of the operand if the operand is positive, otherwise returns zero. The advantage with the above formulation is that:

- There is no need of any additional constraint for each goal.
- Since the suggested solution approach (that will be described later) does not require objective functions to be differentiable, the aforementioned objective function could be used.

Since there is a way to convert a goal programming problem into an equivalent multi-objective problem, a multi-objective genetic algorithm could be used to solve it. The multi-objective genetic algorithm is described in the following section.

3. Multi-objective genetic search

This section presents an efficient method to solve MOV-GP (II) formulation so that the objectives are met and the constraints are satisfied. The algorithm that is adapted to solve the problem is a genetic algorithm (GA) that is a class of adaptive heuristics based on the drawing concept of evolution—"survival of the fitness" developed by Holland at the University of Michigan in 1975 [24]. A GA starts with a set of chromosomes referred to as initial population. Each chromosome represents a solution to the problem, and the initial population is either randomly generated or generated using some of heuristics. A selection mechanism will then be used to select the prospective parents based on their fitness computed by evaluation function. The selected parent chromosomes will then be recombined via the crossover operator to create potential new population. The next step will be to mutate a small number of newly obtained chromosomes, in order to introduce a level of randomness that will preserve the GA from converging to a local optimum. The GA will then reiterate through this process until a predefined number of generations has been produced, or until there was no improvement in the population, which means that the GA has found a near optimal solution, or until a predefined level of fitness has been reached.

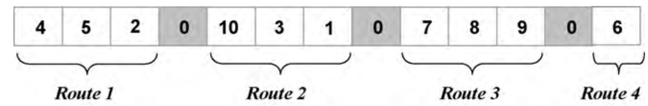

Fig. 1. Typical representation of each chromosome.

In genetic algorithm for evaluation of each chromosome, special fitness function is defined but in MOP application of genetic algorithm the Pareto ranking scheme has often been used [7,33]. The Pareto ranking process tries to rank the solutions to find the non-dominated solutions. Therefore, according to this process each solution gives a rank value in respect of different objective values that shows the quality of the solution in compare to the other solutions. It is easily incorporated into the fitness evaluation process within a genetic algorithm by replacing the raw fitness scores with Pareto ranks. These ranks, to be defined later, stratify the population into preference categories. With it, lower ranks are preferable, and the individuals within rank 1 are the best in the current population. The idea of Pareto ranking is to preserve the independence of individual objectives. This is done by treating the current candidate solutions as stratified sets or ranks of possible solutions. The individuals in each rank set represent solutions that are in some sense incomparable with one another. Pareto ranking will only differentiate individuals that are clearly superior to others in all dimensions of the problem. This contrasts with a pure genetic algorithms attempt to assign a single fitness score to a MOP, perhaps as a weighted sum. Doing so essentially recasts the MOP as a single-objective problem. The difficulty with this is that the weighted sum necessitates the introduction of bias into both search performance and quality of solutions obtained. For many MOP's, finding an effective weighting for the multiple dimensions is difficult and ad hoc, and often results in unsatisfactory performance and solutions.

3.1. Chromosome representation

For solving multi-objective VRPTW with GAs, it is usual to represent each individual by just one chromosome, which is a chain of integers, each of them representing a customer. In this representation each vehicle identifier (gene with index 0) represents in the chromosome a separator between two different routes, and a string of customer identifiers represents the sequence of deliveries that must cover a vehicle during its route. Fig. 1 shows a representation of a possible solution with 10 customers and 4 vehicles that is calculated according to relation (23)

$$\text{Vehicle number at each chromosome} = \text{number of vehicle identifiers} + 1 \tag{23}$$

Each route begins and ends at the depot. If there is a solution that shows two vehicle identifiers in consecutive manner with no customer identifier in between, it would be understood that the route is empty and, therefore, it will not be necessary to use all the vehicles available. This representation allows the number of vehicles to be manipulated and minimized directly for multi-objective optimization in VRPTW. It should be noted that most existing routing approaches consider an individual objective such as cost of traveling distance due to the fact that the number of vehicles is incontrollable in their representations.

3.2. Initial population

An initial population is built such that each individual must at least be a feasible candidate solution, i.e., every route in the initial population must be feasible. In this paper part of population is initialized using heuristics and part is initialized randomly. A fast and simple heuristic procedure to distribute all customers in the vehicles, if used to obtain the part of first individual generation, can reduce significantly the GA time necessary to reach the reasonable local minima. Because of this, the heuristic method proposed by Solomon [40], called *Push Forward Insertion Heuristic* (PFIH), has been frequently used by many researchers with this purpose. For a detailed description of the PFIH method, see [48]. In PFIH method, the relation (24) defines the first customer in each new route. Once the first customer is selected for the current route, the heuristic selects from the set of unrouted customers the one customer which minimizes the total insertion cost between every edge in the current route without violating the time and capacity constraints.

$$c_i = -\alpha d_{0i} + \beta b_i + \gamma\left(\left(\frac{p_i}{360}\right) d_{0i}\right) \tag{24}$$

In relation (24), α is the 0.7 (empirically calculated by Solomon [40]); β the 0.1 (empirically calculated by Solomon [40]); γ the 0.2 (empirically calculated by Solomon [40]); d_{0i} the distance from customer i to the central depot; b_i the upper time and P_i is the polar coordinate angle of the customer i. After the initial feasible solution (S_0) is formed using PFIH, by letting it and its feasible random neighbors $\forall S \in N_\lambda(S_0)$ using λ-interchange described later, a portion of starting population is completed. The rest of the population is generated totally on random basis and starts by inserting customers one by one into an empty route in a random order. Any customer that violates a constraint is deleted from current route. The route is then accepted as part of the solution. A new empty route is added to serve the deleted customer and other remaining customers. This process continues until all customers are routed and a feasible initial population is built. The reason for having this mixed population is that, a population of members entirely from the same neighborhood cannot go too far and hence give up the opportunity to explore other regions.

This paper uses a λ-interchange mechanism that moves customers between routes to generate neighborhood solution for the VRPTW. Given a feasible solution for the VRPTW represented by $S = \{R_1, \ldots, R_p, \ldots, R_q, \ldots, R_k\}$ where R_p is a set of customer served by a vehicle route p. A λ-interchange between a pair of routes R_p and R_q is a replacement of subset $S_1 \subseteq R_p$ of size $\left|S_1\right| \le \lambda$ by another subset $S_2 \subseteq R_q$ of size $\left|S_2\right| \le \lambda$, to get the new route sets R'_p, R'_q and a new neighboring solution $S' = \{R_1, \ldots, R'_p, \ldots, R'_q, \ldots, R_k\}$ where:

$$R'_p = (R_p - S_1) \cup S_2, R'_q = (R_q - S_2) \cup S_1 \tag{25}$$

The neighboring $N_\lambda(S)$ of a given solution S is the set of all neighbors $\{S'\}$ generated by the λ-interchange method for a given λ. In one version of the algorithm called GB (global best), the whole neighborhood is explored and the best move is selected. In another version, FB (first best), the first admissible improving move is selected if exists; otherwise the best admissible move is implemented.

3.3. Pareto ranking

As mentioned earlier, a Pareto ranking scheme is incorporated into a genetic algorithm by replacing the chromosome fitness values with Pareto ranks. These ranks are sequential integer values that represent the layers of stratification in the population obtained via dominance testing. Chromosomes assigned rank 1 are non-dominated, and inductively, those of rank $i+1$ are dominated by all chromosomes of ranks 1 through i. First, the set of non-dominated chromosomes in the population are assigned rank 1. These chromosomes are removed, and the remaining non-dominated chromosomes are assigned rank 2. This is repeated until the entire population is ranked. Evolution then proceeds as usual, using the rank values as fitness scores. Note that Pareto ranks are relative measurements, and there is no concept of "best solution" using a rank score. Fig. 2 illustrates the typical output of the Pareto ranking procedure. At the first step, non-dominated solutions are found over entire population and assigned rank 1. These solutions are the lowest layer of solutions depicted in Fig. 2. Then, these solutions are omitted temporarily and the process is repeated again to find the non-dominated solutions over the remained solution space which it forms the second lowest layer of solutions in Fig. 2. The solutions that found at this stage are assigned rank 2 and this process is repeated until the entire population is ranked.

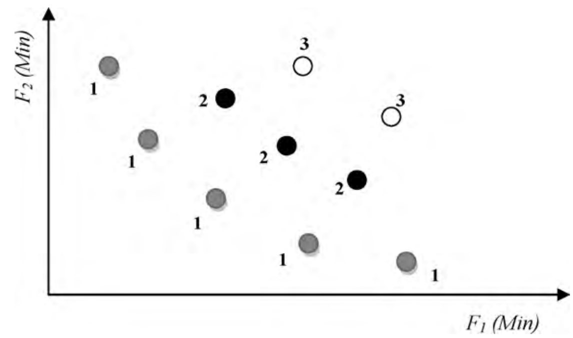

Fig. 2. Typical output of pareto ranking scheme.

The two objectives are the number of vehicles and the total cost of overall traveling distance. They define two independent dimensions in a multi-objective fitness space. Thus, each candidate VRPTW solution in the population has associated with a vector $\bar{v} = (\bar{n}, \bar{c})$ where $\bar{n}$ is the objective value for number of vehicles yielded by the solution, and $\bar{c}$ is the objective value for total cost according to relation (26)

$$\begin{aligned} &Goal: G_N(\vec{x}) \le TL_1 \to \bar{n} = \langle G_N(\vec{x}) - TL_1 \rangle \\ &Goal: G_C(\vec{x}) \le TL_2 \to \bar{c} = \langle G_C(\vec{x}) - TL_2 \rangle \end{aligned} \tag{26}$$

In relation (26), $G_N(\vec{x})$ and $G_C(\vec{x})$ are the number of vehicles and the total cost of chromosome $\vec{x}$ that are presented earlier as Eqs. (1) and (2) at MOV-GP formulation and *TL1* and *TL2* are the target level for number of vehicles and total cost determined by decision maker. These two dimensions are retained as independent values, to be eventually used by the Pareto ranking procedure. Pareto ranking is applied to the $(\bar{n}, \bar{c})$ vectors of the population, essentially creating for the population a set of integral ranks ≥ 1. These ranks are then used by the GA as fitness values for generating the next population. Note that the ranks themselves do not convey the quality of solutions, nor whether an optimal solution has been discovered. Each population, including the randomized initial population, is guaranteed to have a rank 1 set. This is not a disadvantage for general instances of the VRPTW anyway, since there is no efficient means of knowing whether a candidate VRPTW solution is truly optimal.

3.4. Tournament selection

In tournament selection two identical (through differently ordered) copies of the population are kept. In every generation, adjacent chromosomes are compared in one copy of the population pair by pair, and the chromosome with a lower rank is selected, and then this procedure is repeated with the second copy of the

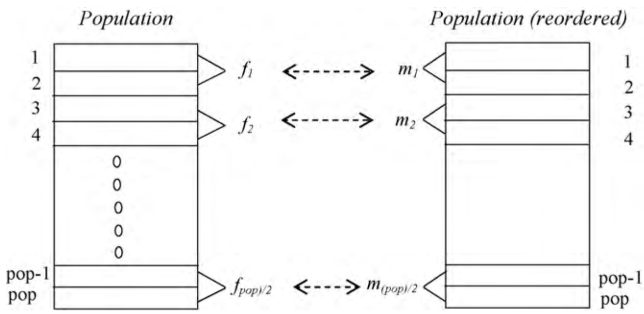

Fig. 3. Tournament selection procedure.

population to select the other half of the selected population. This procedure is illustrated in Fig. 3.

As shown in Fig. 3, two identical copies of the population with size "pop" are maintained at every generation and ranked arbitrary. For each population, adjacent chromosomes are compared and the solutions with lower rank qualify to be potential parent. After comparing all pairs in two populations, *pop/2* "fathers" namely $f_1, \ldots, f_{(pop)/2}$ and *pop/2* "mothers" namely $m_1, \ldots, m_{(pop)/2}$ are created and each f_i and m_i are mated subsequently. In this procedure the superior chromosomes are given priority in mating but average entities have chance of being selected too. This procedure first was properly applied by [30] and this article tunes the procedure according to the assumptions and concept of non-dominated solutions.

3.5. Best cost-best route crossover (BCBRC)

One of the unique and important aspects of the techniques involving genetic algorithms is the important role that recombination (traditionally, in the form of crossover operator) plays. Since standard genetic operators may generate infeasible solutions for VRPTW, the specialized genetic operators should be designed and describe later.

Classical one-point crossover may produce infeasible route sequence because of the duplication and omission of vertices after reproduction. Ishibashi et al. [17] proposed a two-point ordered crossover that randomly selects two crossing points from parents and decides which segment should be inherited to the offspring. In [8] the authors carried experiments where they established two standard crossover operators: Uniform Order Crossover (UOX) [53] and Partially Mapped Crossover (PMX) [53]. In [8] the authors then introduced Route Crossover (RC) which is an improvement of the UOX. Experimental details showed that the RC outperforms UOX and PMX.Ombuki et al. [7] employed a problem-specific crossover best cost-route crossover (BCRC) which aimed at minimizing the number of vehicles and cost simultaneously while checking feasibility constraints. In that operation, two routes from the parents are selected randomly, then all the genes in a selective route of a parent are removed from another parent and eventually the crossover reinserts each removed gene in best position of the parent. Tan et al. [31] proposed a simple crossover operator for HMOEA that allows the good sequence of routes or genes in a chromosome to be shared with other individuals in the population. The operation was designed such that infeasibility after the change could be eradicated easily.

This paper employs the best cost-best rout crossover (BCBRC) which selects a best route from each parent. BCBRC is very similar to BCRC [7] with minor differences. The best route is chosen according to the criteria of averaged cost over nodes. Next, for a given parent, the customers in the chosen route from the opposite parent are removed. Since each chromosome should contain the entire customer numbers, the next step is to locate the best possible locations for the removed customers in the corresponding children. This operator minimizes the number of vehicles and cost simultaneously while checking feasibility constraints.

3.6. Sequenced based mutation (SBM)

Given two children solution produced from crossover phase, this operator first selects randomly a link to break a route on each of the solutions and then make an exchange on the routes before and after the break points to produce new children solutions. Fig. 4 illustrates how SBM applies to create a new child solution. SBM selects randomly break point 1 on child solution 1 and break point 2 on child solution 2, and then make a connection between the route of customers served before the break point 1 and the route of customer serviced after the break point 2. As a result, *new child solution 1* is created that is made up of a newly created route R^*_1, removal of route R_1 and keeping route R_2 as it was. Likewise, the second new chromosome could be created by reversing the role of the children, i.e., the customers that are serviced before the break point 2 on the route of child-solution 2 are linked to the customers that are serviced after the break point 1 on the route of child-solution 1 and a *new child solution 2* is created.

In a feasible solution, customers with early time windows are typically scheduled at the beginning of a route. Conversely, customers with late time windows are typically scheduled at the end of the route. Hence, by linking the first customers on a route of child-solution 1 to the last customers on a route of child-solution 2, the time window constraints are likely to be satisfied.

Since some customers are duplicated or unrouted in the process, a repair operator is applied to the new chromosome to generate a new feasible solution. For example, in Fig. 4, two customers in *new child solution 1* are located on two different routes (customers *g* and *h* are located on two routes R^*_1 and R_2), and two other customers are unrouted (customers d and e). This operator deals with the infeasible solution in the following way:

- If a customer appears twice in the new route, one of the two copies is removed from the route. If a customer appears once in the new route, and once in an old route, the customer is removed from the old route.
- If a customer is unrouted, then this customer is inserted at the feasible insertion place that minimizes the additional cost and satisfies capacity and time window constraints. Obviously, there is no guarantee that there is a feasible insertion place for each of the customers. If this situation occurs, the new solution is discarded, and an old child-solution is restored.

Hence, customers *g* and *h* are removed from R_2 in *new child solution 1* and customers *d* and *e* are inserted in the best location. Fig. 5 shows how to apply repair operator on a new solution.

3.7. Hill-climbing

Hill-climbing is a scheme for randomly selecting a portion of the population and then improving those solutions by a few iterations of removal and reinsertion. The hill-climbing approach can contribute to the intensification of the optimization results, which is usually regarded as a complement to evolutionary operators that mainly focus on global exploration.

Two famous local heuristics namely one-interchange (FB) and shortest path heuristic are incorporated in this paper to search for better routing solutions in the VRPTW. There is no preference made among the local heuristics and one of them is randomly executed at the end of each generation for a portion of the population to search for better local routing solutions.

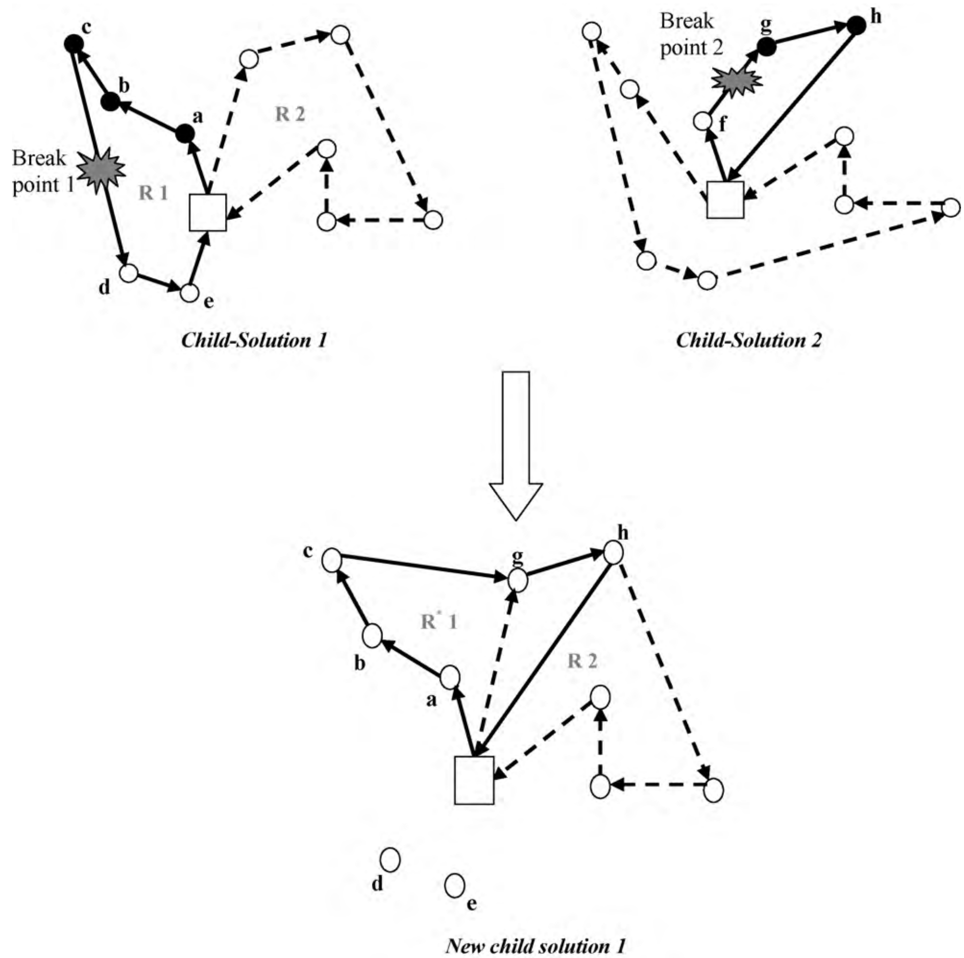

Fig. 4. SBM operator on child-solution 1.

The shortest path heuristic [46] that also used efficiently by Tan et al. [31] is modified from the 'shortest path first' method. It attempts to rearrange the order of nodes in a particular route such that the node with the shortest distance is given priority. In order to rearrange the order of nodes, the first node is chosen based on its distance from the depot and the second node is chosen based on its distance from the first customer node. The process repeats until all nodes in the original route are re-routed. The original route will be restored if the new route obtained is infeasible.

3.8. Elitism

The elitism strategy keeps a small number of good individuals and replaces the worst individuals in the next generation without going through the usual genetic operations.

4. Experimental results and comparisons

This section describes computational experiments carried out to investigate the performance of the proposed GA. The algorithm was coded in MATLAB 7 and run on a PC with 1.6 GHz CPU and 512 MB memory. The experimental results use the standard Solomon's VRPTW benchmark problem instances available at [54].

The Solomon's problems consist of 56 data sets, which have been extensively used for benchmarking different heuristics in literature over the years. The problems vary in 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 customers' details are given in the sequence of customer index, location in x and y coordinates, the demand for load, the ready time, due date and the service time required. All the test problems consist of 100 customers, which are generally adopted as the problem size for performance comparisons in VRPTW. The traveling time between customers is equal to the corresponding Euclidean distance. Solomon's data is clustered into six classes; C1, C2, R1, R2, RC1 and RC2. Problems in the C category means the problem is clustered, that is, 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

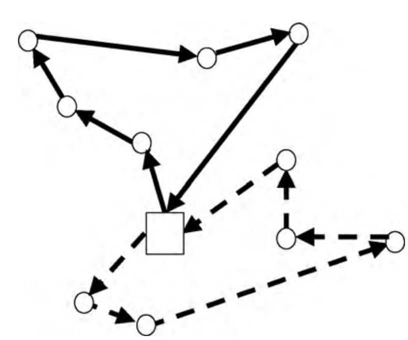

Fig. 5. Repair operator on a new child solution.

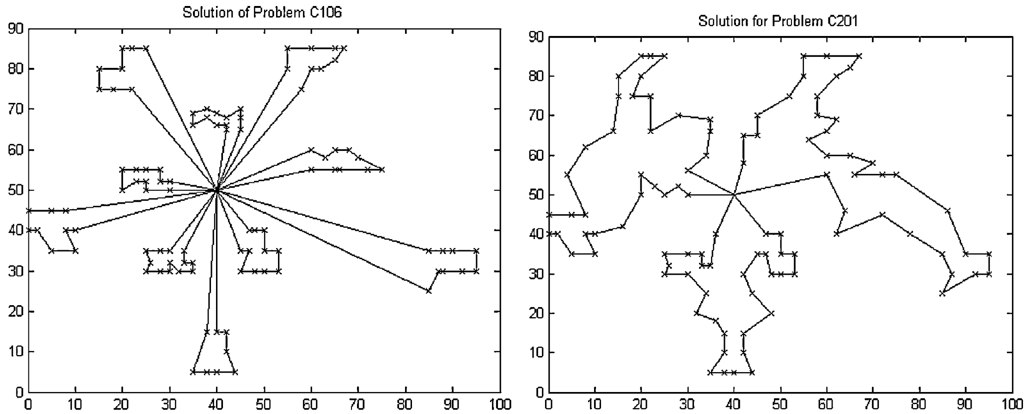

Fig. 6. Typical output for problems C106 (left figure) and C201 (right figure).

both C and R. 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 problems have wider time windows hence many customers can be served by vehicles. Unless otherwise stated, the results presented below are based on the following parameters:

- Population size = 100.
- Generation number = 700.
- Crossover rate = 0.80.
- Mutation rate = 0.20.
- Number of chromosomes undergoing hill-climbing phase = 10.
- Number of chromosomes undergoing recovery phase = 4.
- Repetition for experiments = 10.

As mentioned earlier, this paper considers VRPTW model for solving the Solomon's instances with two objectives including the number of vehicles and the total traveling cost that need to be optimized concurrently. It must be mentioned that, these objectives are quantitatively measurable, but the relationship between these two objectives is unknown until the problem has been solved. So this section wants to analyze this behavior between these two values. These two objectives may be positively correlated with each other. In other word the higher routing cost may be incurred if more vehicles are involved. But these two values may be conflicting to each other. For example, the routing cost of a solution is reduced as the number of vehicles is increased. This behavior is difficulty determined by the classical approach (weighted sum approach or using single objective) and it is easily analyzed by the suggested approach.

Table 1 presents a summary of results and compares the findings with the best known solutions that are reported in the literature. Distance costs are measured by average Euclidian distance. The column labeled *Best Known* gives the best known published solutions so far; column *suggestive approach* gives the results that are produced in 10 runs, when the VRPTW was interpreted as a multi-objective optimization problem. In Table 1, the suggested approach is divided into two columns representing the non-dominated solutions and average results over 10 runs. TL_1 and TL_2, target values for the goals, are supposed to be the best known values published in the literature and whenever the algorithm finds equal or better solutions, those solutions are reported. Bolded numbers in Table 1 indicate the solutions obtained by the suggested method which are either the same or better than the best known solutions reported in the literature (when considering either number of vehicles or cost).

Fig. 6, shows a typical output for test problems C106 and C201 and Figs. 7 and 8 show the output for problems RC102 and R108.

In the suggested approach, when using the Pareto ranking, DM's preference relation among the objectives, namely number of vehicles and overall traveling distance, are taken into consideration to choose among the solutions. Based on Table 1, 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 instances in classes R and RC reduce the distance significantly, but at the expense of adding extra vehicles. In general, as it mentioned earlier, the relationship between these two objectives (*Distance Cost* and *Number of Vehicles*) in a routing problem is unknown until the problem is solved. These two objectives may be positively correlated with each other, or they may be conflicting to each other. This section tries to determine and analyze this behavior. According to Table 1 that shows the results of the suggested GA approach on the Solomon's 56 data sets, all instances in the categories of C1 and C2 have positively correlating objectives. In other words, the routing cost of a solution is increased as the number of vehicles is increased. For instance, Figs. 9 and 10 represent the correlation between distance cost and number of vehicles for problems C107 and C204 yielded by the suggested GA.

On the opposite side, the GA algorithm suggests that there are conflicting objectives for some instances in R1, R2, RC1 and RC2 categories, i.e., for these instance problems, distance cost reduces as the number of vehicles is increased. This conflicting behavior for some instances in classes R and RC is reflected in Table 1. For instance in problem RC105, with increasing the number of vehicles by 1, the distance cost of solution is reduced by 1.37% (from 1611.5 to 1589.4). Note that these alternate Pareto solutions are clearly comparable and decision maker (DM) can decide which solution is more preferable based on his/her preferences.

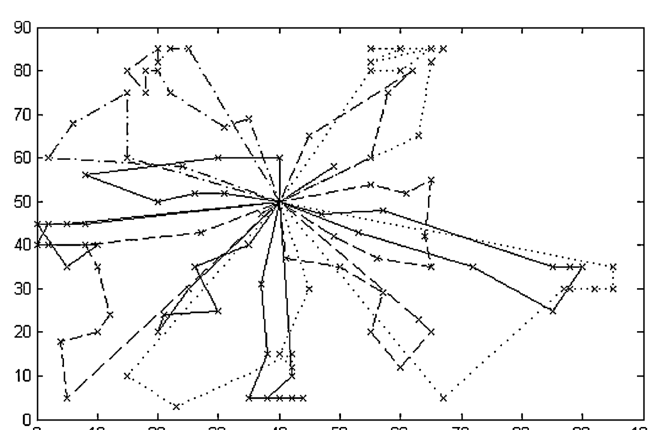

Fig. 7. Typical output for problem RC102.

Table 1
Testing result on benchmark Solomon's 56 VRPTW 100-customer instances.

Instance data	Best known			Suggestive approach			
	Vehicle # (TL_1)	Distance cost (TL_2)	Ref.	Non-dominated solutions		Average of solutions in 10 runs	
				Vehicle #	Distance cost	Vehicle #	Distance cost
C101	10	828.94	[7]	**10**	**828.94**	10	828.94
C102	10	828.94	[7]	**10**	**828.94**	10	839.41
C103	10	828.06	[51]	**10**	**828.06**	10	849.17
C104	10	824.78	[7]	**10**	**824.78**	10	845.56
C105	10	828.94	[7]	**10**	**828.94**	10	828.94
C106	10	828.94	[7]	**10**	**828.94**	10	828.94
C107	10	828.94	[7]	**10**	**828.94**	10	828.94
C108	10	828.94	[7]	**10**	**828.94**	10	839.16
C109	10	828.94	[7]	**10**	**828.94**	10	828.94
C201	3	591.56	[29]	**3**	**591.56**	3	591.56
C202	3	591.56	[29]	**3**	**591.56**	3	593.24
C203	3	591.17	[51]	**3**	**591.17**	3	614.15
C204	3	590.60	[29]	3	599.96	3	603.94
C205	3	588.16	[31]	3	588.88	3	590.74
C206	3	588.49	[29]	3	588.88	3	592.42
C207	3	588.29	[51]	3	591.56	3	593.24
C208	3	588.32	[51]	**3**	**588.32**	3	597.70
R101	19	1650.80	[51]	19	1677.0	19.4	1673.9
				20	1651.1		
R102	17	1434	[37]	18	1511.8	18.7	1510.6
				19	1494.7		
R103	14	1237.05	[7]	14	1287.0	14.3	1291.1
				15	1264.2		
R104	10	974.24	[31]	**10**	**974.24**	11.2	1043.3
R105	14	1377.11	[44]	15	1424.6	15.6	1409.6
				16	1382.5		
R106	12	1252.03	[51]	13	1270.3	14	1315.9
R107	11	1100.52	[7]	11	1108.8	11.8	1134.8
R108	10	960.26	[7]	10	971.91	10.3	1014.3
R109	12	1169.85	[7]	12	1212.3	13	1220.1
				14	1206.7		
R110	11	1112.21	[7]	12	1156.5	12.2	1160.7
R111	10	1096.72	[7]	11	1111.9	11.9	1149.1
R112	10	976.99	[7]	10	1036.9	10.5	1051.7
				11	1011.5		
R201	5	1206.42	[31]	**4**	1351.4	4	1358.7
R202	4	1091.21	[31]	**4**	**1091.22**	4	1173.1
R203	4	935.04	[31]	3	1041.0	4.8	1022.3
				5	995.8		
				6	978.5		
R204	3	789.72	[31]	3	1130.1	5.4	839.82
				4	927.7		
				5	831.8		
				6	826.2		
R205	3	994.42	[7]	4	1087.8	3.4	1188.5
				3	1422.3		
R206	3	833	[48]	3	940.12	3	1004.0
R207	3	814.78	[51]	3	904.90	3	907.9
R208	2	726.823	[35]	3	774.18	3	778.25
R209	3	855	[48]	4	1008.0	4	1009.9
R210	3	954.12	[18]	**3**	**938.58**	3.2	1020.3
R211	4	761.10	[7]	4	1101.5	3.6	1191.0
				3	1310.4		
RC101	15	1636.92	[7]	15	1690.6	15.3	1693.2
				16	1678.9		
RC102	13	1470.26	[31]	14	1509.4	14.5	1521.0
				15	1493.2		
RC103	11	1261.67	[44]	12	1331.8	12.2	1357.4
RC104	10	1135.48	[44]	11	1177.2	11	1213.5
RC105	16	1590.25	[7]	**15**	**1611.5**	15.9	1610.5
				16	**1589.4**		
RC106	11	1427.13	[21]	13	1437.6	13.5	1437.1
				14	**1425.3**		
RC107	11	1230.48	[44]	**11**	**1222.1**	12.2	1287.9
RC108	10	1142.66	[12]	11	1156.5	11.3	1197.9
RC201	6	1134.91	[31]	4	1423.7	4	1457.0
RC202	4	1181.99	[7]	4	1369.8	4	1381.9
RC203	4	1026.61	[31]	4	1060.0	4.9	1196.7
				6	**1020.1**		
RC204	3	798.46	[35]	3	901.46	3	926.74
RC205	4	1300.25	[31]	4	1410.3	4	1411.3
RC206	3	1153.93	[7]	4	1194.8	4	1195.5
RC207	4	1040.67	[31]	**4**	**1040.6**	4	1070.3
RC208	4	785.93	[7]	3	898.50	3.7	905.07

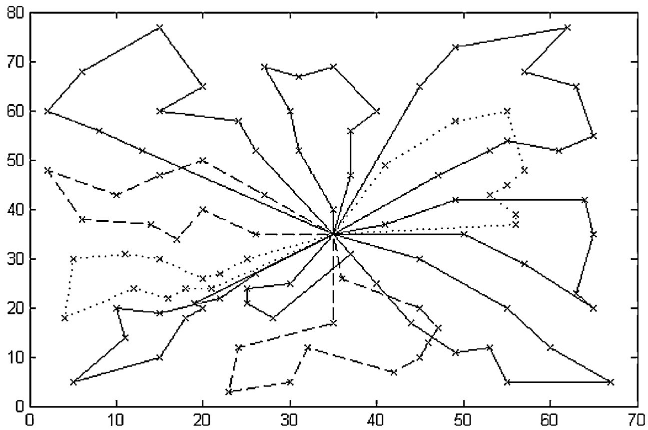

Fig. 8. Typical output for problem R108.

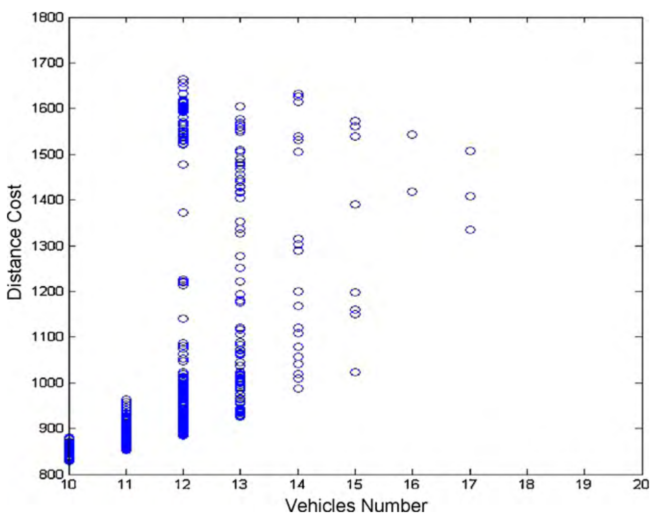

Fig. 9. Comparison of population distribution for problem C107.

Likewise, the GA suggests the conflicting behavior for problem R204where step by step increase in the number of vehicles (from 3 vehicles to 6 vehicles), the routing cost is steadily decreased from 1130.1 to 826.2 which indicates the conflicting behavior between the two objectives. This behavior based on the performance of the suggested GA, is shown in Fig. 11.

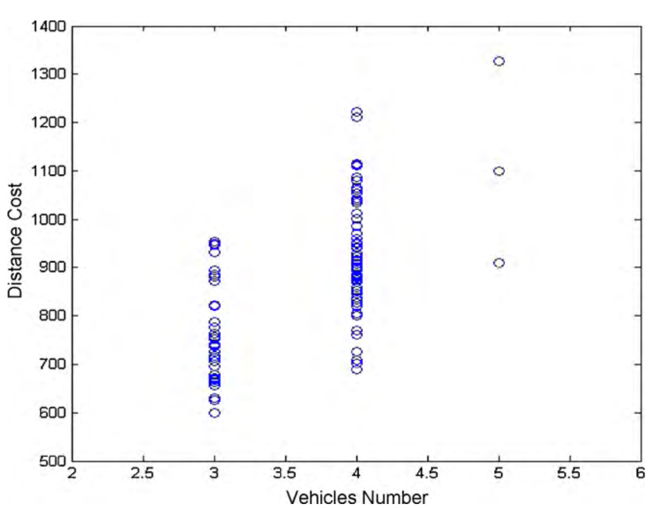

Fig. 10. Comparison of population distribution for problem C204.

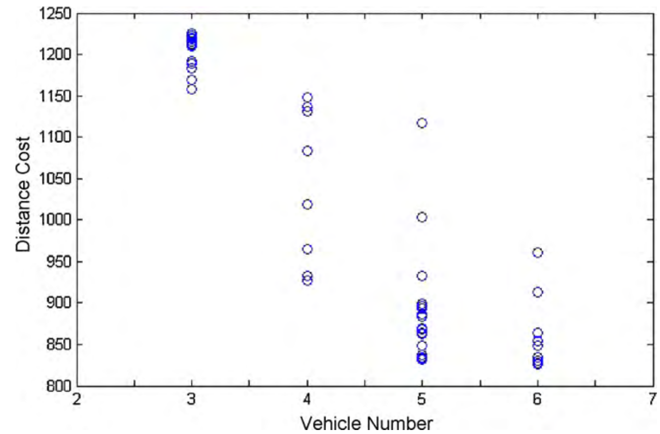

Fig. 11. Conflicting behavior for problem R204.

Moreover, in most of instances according to Table 1, just a single Pareto solution is reported that is the best solution generated by the suggested GA method.

The GA algorithm implies that while some instances in classes R and RC have conflicting objectives, other instances show no conflict between the objectives and the routing cost of a solution increases as the number of vehicles is increased. For instance, Fig. 12 shows the positively correlating objectives for instance problem RC108.

Furthermore, the GA method suggests that the conflicting behaviors are commonly seen in classes R1 and R2.

While conventional single-objective vehicle routing approaches are unable to explore the conflicting behavior of objectives, the GA algorithm is adequate to easily discover the behavior for a bi-objectives vehicle routing problem. Moreover, the proposed procedure obviates the need to experiment with weights as required in a weighted-sum approach because poorly chosen weights results in unsatisfactory solutions.

Eventually, this paper tried to emphasize on the number of vehicles as the separate objective besides the total cost traveling. Because there is an associated cost to having more vehicles and considering to it in different cases is supposed to be important. So in some cases when vehicle and its associated costs namely man-power costs, fuel consumptions costs, etc. are negligible, the routing plan can be used with the conventional approach and using single objective procedure. But when the costs of vehicle counts and minimum traveling distance for reducing the energy consumption are important, the MOP approach can be effective and generates a set of equally valid VRPTW solutions as non-dominated solutions. These solutions represent a range of possible answers, with differ-

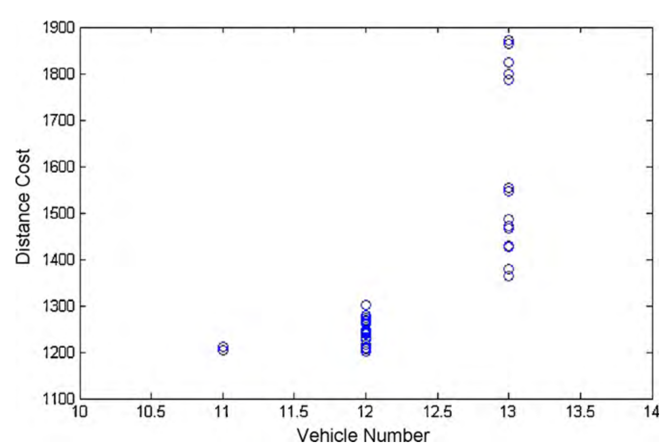

Fig. 12. Comparison of population distribution for problem RC108.

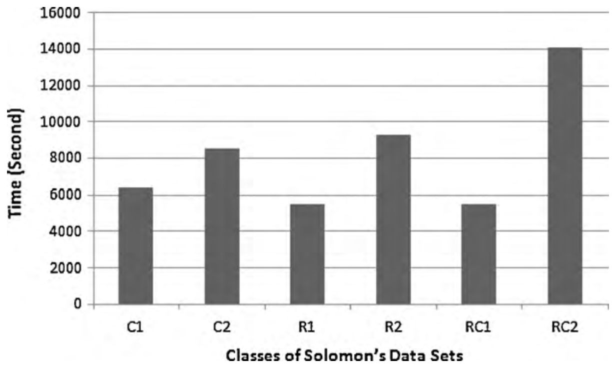

Fig. 13. Average computation time for each category.

ent numbers of vehicles and costs which the decision maker (DM) can decide which kind of solution is preferable.

Another considerable advantage for the suggested method is its computational time. The average computation time (in second) for each class of Solomon's data sets is shown in Fig. 13. According to this figure, the classes C2, R2 and RC2 requires a longer computation time in compare to the other categories which is on account of the wider time windows for these classes that allow a more flexible arrangement in the routing construction process.

According to Table 1 and Fig. 13, the results obtained by the suggested model are quite good as compared to the best published results and the average GA performance is sufficient in respect of time and quality.

5. Conclusion

This paper suggested a new model and solution for multi-objective vehicle routing problem with time windows (VRPTW) using goal programming and genetic algorithm. This paper considered the VRPTW as a multi-objective problem that in which fleet size of vehicles and total traveling distance are minimized while capacity and time window constraints are not violated. This paper formulated Multi-Objective VRPTW mathematically as MOV-GP (I) formulation and then reformulated it as MOV-GP (II) formulation with the approach of goal programming. In this idea the decision maker specified optimistic aspiration levels to the objective functions of the problem and deviations from these aspiration levels were minimized. Then an efficient multi-objective genetic algorithm was suggested for solving MOV-GP (II) formulation that incorporates various heuristics for local exploitation in the evolutionary search and the concept of Pareto's optimality for the multi-objective optimization. The proposed genetic algorithm used a string of customer identifiers which represented the sequence of deliveries that must cover a vehicle during its route and each vehicle identifier represented a separator between two different routes in the chromosome.

Part of initial population was initialized using Push Forward Insertion Heuristic (PFIH) which has frequently been used by many researchers and part was initialized randomly. A λ-interchange mechanism was used to interchange customers between routes and generate neighborhood solution as well as a Pareto ranking scheme was incorporated into a genetic algorithm by replacing the chromosome fitness values with Pareto ranks.

In this paper, the best cost-best rout crossover (BCBRC) was employed on the parent solutions selected by a tournament approach. This operator minimized the number of vehicles and cost simultaneously while checking feasibility constraints. A special mutation operator namely sequenced based mutation (SBM) was applied to explore a larger area of solution space as well. Meanwhile, two famous local heuristics namely one-interchange (FB) and shortest path heuristic were incorporated in this paper to search for better routing solutions and elitism strategy was used to keep a few number of good individuals. At the end, the algorithm was applied to solve the benchmark Solomon's 56 VRPTW 100-customer instances. According to the produced results, the suggested approach was quite sufficient as compared to the best published results and the average GA performance was adequate.

References

[1] A. Charnes, W.W. Cooper, R. Ferguson, Optimal estimation of executive compensation by linear programming, Management Science 1 (1955) 138–151.
[2] A. Charnes, W.W. Cooper, Management Models and Industrial Applications of Linear Programming, Wiley, New York, 1961.
[3] B. Crevier, J.F. Cordeau, G. Laporte, The multi-depot vehicle routing problem with inter-depot routes, European Journal of Operational Research 176 (2007) 756–773.
[4] B. De Backer, V. Furnon, P. Shaw, P. Kilby, P. Prosser, Solving vehicle routing problems using constraint programming and metaheuristics, Journal of Heuristics 6 (2000) 501–523.
[5] B.I. Kim, S. Kim, S. Sahoo, Waste collection vehicle routing problem with time windows, Computers & Operations Research 33 (2006) 3624–3642.
[6] B.L. Golden, A.A. Assad, Vehicle Routing: Methods and Studies, North Holland, Amsterdam, 1988.
[7] B. Ombuki, B. Ross, F. Hanshar, Multi-objective genetic algorithm for vehicle routing problem with time windows, Applied Intelligence 24 (2006) 17–30.
[8] B. Ombuki, M. Nakamura, O. Maeda, A hybrid search based on genetic algorithms and tabu search for vehicle routing, in: A.B. Banff, H. Leung (Eds.), Proceedings of the 6th IASTED International Conference On Artificial Intelligence and Soft Computing (ASC 2002), ACTA Press, 2002, pp. 176–181.
[9] C. Romero, Handbook of Critical Issues in Goal Programming, Pergamon Press, Oxford, 1991.
[10] D. Pisinger, S. Ropke, A General Heuristic for Vehicle Routing Problem, Department of Computer Science, University of Copenhagen, 2005.
[11] D.F. Jones, M. Tamiz, Goal programming in the period 1990–2000, in: M. Ehrgott, X. Gandibleux (Eds.), in Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys, Kluwer, 2002, pp. 129–170.
[12] E.D. Taillard, P. Badeau, M. Gendreau, F. Gueritin, J.-Y. Potvi, A tabu search heuristic for the vehicle routing problem with soft time windows, Transportation Science 31 (1997) 170–186.
[13] F. Li, B. Golden, E. Wasil, very large scale vehicle routing: new test problems algorithms and results, Computers & Operations Research 32 (2005) 1165–1179.
[14] G.B Alvarenga, G.R. Mateus, G. de Tomi, A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows, Computers & Operation Research 34 (2007) 1561–1584.
[15] G. Laporte, F. Semet, Classical Heuristics for the Vehicle Routing Problem, Les Cahiers du GERAD, G98-54, Group for Research in Decision Analysis, Montreal, Canada, 1999.
[16] H.I. Calvete, C. Galé, M.J. Oliveros, B.S. Valverde, A goal programming approach to vehicle routing problems with soft time windows, European Journal of Operational Research 117 (2007) 1720–1733.
[17] H. Ishibashi, H. Aguirre, K. Tanaka, T. Sugimura, Multi-objective optimization with improved genetic algorithm, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC), Nashville, 2000, pp. 3852–3857.
[18] J. Berger, M. Barkaoui, O. Bräysy, A Parallel Hybrid Genetic Algorithm for the Vehicle Routing Problem with Time Windows, Working paper, Defense Research Establishment Valcartier, Canada, 2001.
[19] J. Berger, M. Barkaoui, A hybrid genetic algorithm for the capacitated vehicle routing problem, in: E. Cantú-Paz (Ed.), GECCO03. LNCS 2723, Springer-Verlag, Chicago, 2003.
[20] J. Cerda, R. Dondo, A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows, European Journal of Operational Research 176 (2007) 1478–1507.
[21] J.F. Cordeau, G. Larporte, A. Mercier, A unified tabu search heuristic for vehicle routing problems with time windows, Journal of the Operational Research Society 52 (2001) 928–936.
[22] J.D. Knowles, D.W. Corne, Approximating the nondominated front using Pareto archived evolutionary strategy, Evolutionary Computation 8 (2000) 149–172.
[23] J. Desrosier, Y. Dumas, M. Solomon, F. Soumis, in: M. Ball (Ed.), Time Constraint Routing and Scheduling, Handbooks in Operations Research and Management Science 8: Network Routing, Elsevier Science Publisher, Amsterdam, 1995, pp. 35–139.
[24] J.H. Holland, Adaptation in Natural and Artificial System, The University of Michigan Press, Ann Arbor, Michigan, 1975.
[25] J.L. Blanton Jr., R.L. Wainwright, Multiple vehicles routing with time and capacity constraints using genetic algorithms, in: Proceedings of the Fifth International Conference on Genetic Algorithms, 1993, pp. 452–459.
[26] J. Larsen, Parallelization of the vehicle routing problem with time windows, Ph.D. thesis, IMM-PHS-1999-62, Department of Mathematical Modelling, Technical University of Denmark, Lynghy, Denmark, 1999.

[27] J. Louis, X. Yin, Z.Y. Yuan, Multiple Vehicle Routing Problem With Time Windows Using Genetic Algorithm, Genetic Adaptive System LAB, Department of Computer Science, University of Nevada, Reno, 1999.

[28] J.P. Ignizio, Goal Programming and Extensions, Lexington Books, Lexington, MA, 1976.

[29] J.Y. Potvin, S. Bengio, The vehicle routing problem with time windows. Part II. Genetic search, INFORMS Journal of Computing 8 (1996) 165–172.

[30] K.C. Tan, L.H. Lee, K.Q. Zhu, K. Qu, Heuristic methods for vehicle routing problem with time windows, Artificial Intelligence in Engineering 15 (2001) 281–295.

[31] K.C. Tan, Y.H. Chew, L.H. Lee, A hybrid multiobjective evolutionary algorithm for solving vehicle routing problem with time windows, Computational Optimization and Applications 34 (2006) 115–151.

[32] K.C. Tan, L.L. Hay, O. Ke., A hybrid genetic algorithm for solving vehicle routing problems with time window constraints, Asia-Pacific Journal of Operational Research 18 (2001) 121–130.

[33] K.C. Tan, C.Y. Cheong, C.K. Goh, Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation, European Journal of Operational Research 177 (2007) 139–813.

[34] K.C. Tan, T.H. Lee, E.F. Khor, Evolutionary algorithm with dynamic population size and local exploration for multiobjective optimization, IEEE Transactions on Evolutionary Computation 5 (2001) 565–588.

[35] L.M. Gambardella, E. Taillard, G. Agazzi, MACS-VRPT.W. A multiple ant colony system for vehicle routing problems with time windows, in: David Corne, Marco Dorigo, Fred Glover (Eds.), New Ideas in Optimization, McGraw-Hill, London, 1999, pp. 63–76.

[36] M. Dorigo, L.M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, IEEE Transactions on Evolutionary Computation 1 (1997) 53–66.

[37] M. Desrochers, J. Desrosiers, M. Solomon, A new optimization algorithm for the vehicle routing problem with time windows, Operational Research 40 (1992) 342–354.

[38] M. Gendreau, G. Laporte, J.Y. Potvin, Metaheuristics for the vehicle routing problem, Technical Report CRT-963, Centre de Recherche sur les Transports, Université de Montréal, 1999.

[39] M.J. Scniederjans, Goal Programming Methodology and Applications, Kluwer publishers, Boston, 1995.

[40] M.M. Solomon, Algorithms for the vehicle routing and scheduling problems with time window constraints, Operations Research 35 (1987) 254–265.

[41] N. Achutan, L. Caccettal, S. Hill, An improved branch-and-cut algorithm for the capacitated vehicle routing problem, Transportation Science 37 (2003) 153–169.

[42] N. Kohl, Exact methods for time constrained routing and related scheduling problems, Ph.D. Thesis, Department of Mathematical Modeling, Technical University of Denmark, 1995.

[43] O. Bräysy, M. Gendreau, Metaheuristics for the Vehicle Routing Problem with Time Windows, SINTEF Applied Mathematics Research Council of Norway, 2001.

[44] P. Shaw, Using constraint programming and local search methods to solve vehicle routing problems, in: M. Maher, J.-F. Puget (Eds.), in Proceedings of the Fourth International Conference on Principles and Practice of Constraint Programming, Springer-Verlag, 1998, pp. 417–431.

[45] S. Irnich, B. Funke, T. Grunert, Sequential search and its application to vehicle routing problems, Computer & Operations Research 33 (2006) 2405–2429.

[46] S. Lin, Computer solutions for traveling salesman problem, Bell System Technical Journal 44 (1965) 2245–2269.

[47] S.M. Lee, Goal Programming for Decision Analysis, Auerback, Philadelphia, 1972.

[48] S.R. Thangiah, A hybrid genetic algorithms, simulated annealing and tabu search heuristic for vehicle routing problems with time windows, in: L. Chambers (Ed.), Practical Handbook of Genetic Algorithms Complex Structures, vol. 3, CRC Press, 1999, pp. 347–381.

[49] S. Thangiah, Vehicle routing with time windows using genetic algorithms, in: Applications Handbook of Genetic Algorithms: New Frontiers, vol. II, CRC Press, Boca Raton, 1995, pp. 253–277.

[50] W.C. Chiang, Russell, Simulated annealing metaheuristic for the vehicle routing problem with time windows, Annals of Operations Research 63 (1996) 3–27.

[51] Y. Rochat, E.D. Taillard, Probabilistic diversification and intensification in local search for vehicle routing, Journal of Heuristics 1 (1995) 147–167.

[52] Z.J. Czech, P. Czarnas, Parallel simulated annealing for the vehicle routing problem with time windows, in: Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing, Canary Islands, Spain, 2002, pp. 376–383.

[53] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, 3rd ed., Springer-Verlag, New York, 1998.

[54] http://web.cba.neu.edu/∼msolomon/problems.htm.