Computers & Industrial Engineering 65 (2013) 286–296

Contents lists available at SciVerse ScienceDirect

Computers & Industrial Engineering

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

A hybrid meta-heuristic for multi-objective vehicle routing problems with time windows ☆

Raúl Baños [a,*], Julio Ortega [a], Consolación Gil [b], Antonio L. Márquez [b], Francisco de Toro [c]

[a] Dpt. Computer Architecture and Technology, CITIC-UGR (Research Centre on Information and Communications Technology), University of Granada, C/Periodista Daniel Saucedo s/n, E-18071 Granada, Spain

[b] Dpt. Computer Architecture and Electronics, University of Almería, Carretera de Sacramento s/n, La Cañada de San Urbano s/n, E-04120 Almeria, Spain

[c] Dpt. Signal Theory, Telematics and Communications, CITIC-UGR (Research Centre on Information and Communications Technology), University of Granada, C/Periodista Daniel, Saucedo s/n, E-18071 Granada, Spain

ARTICLE INFO

Article history:
Received 29 December 2011
Received in revised form 1 December 2012
Accepted 15 January 2013
Available online 14 February 2013

Keywords:
Vehicle routing problems
Time windows
Travelling distance
Load imbalance
Multi-objective optimization
Hybrid meta-heuristics

ABSTRACT

The Capacitated Vehicle Routing Problem with Time Windows is an important combinatorial optimization problem consisting in the determination of the set of routes of minimum distance to deliver goods, using a fleet of identical vehicles with restricted capacity, so that vehicles must visit customers within a time frame. A large number of algorithms have been proposed to solve single-objective formulations of this problem, including meta-heuristic approaches, which provide high quality solutions in reasonable runtimes. Nevertheless, in recent years some authors have analyzed multi-objective variants that consider additional objectives to the distance travelled. This paper considers not only the minimum distance required to deliver goods, but also the workload imbalance in terms of the distances travelled by the used vehicles and their loads. Thus, MMOEASA, a Pareto-based hybrid algorithm that combines evolutionary computation and simulated annealing, is here proposed and analyzed for solving these multi-objective formulations of the VRPTW. The results obtained when solving a subset of Solomon's benchmark problems show the good performance of this hybrid approach.

1. Introduction

The Vehicle Routing Problem (VRP), and its multiple variants, is a core problem in transportation, logistics, and supply chain management. Logistics, and especially the distribution of goods, lies at the heart of business activity because it is often coupled with inventory and production decisions, and the delivery cost accounts for a significant portion of the total logistic costs (Alabas-Uslu & Dengiz, 2011). Bearing in mind that problems in the domain of goods distribution can be viewed as a VRP (Mester & Bräysy, 2007), this problem contributes directly to reducing costs of all logistic systems (Alvarenga, Mateus, & de Tomic, 2007). Logistic managers need to make decisions to improve the design of their logistic systems, including appropriate decisions concerning the strategies to provide customers with their services while satisfying the company's logistic priorities according to the available vehicle fleet. Furthermore, current concerns over global warming, resource depletion, and the social impact of traffic congestion and pollution are driving companies, governments, and researchers to improve the efficiency of logistics and distribution operations (Hosny & Mumford, 2010).

VRPs are combinatorial optimization problems linked with many branches of mathematics, economics, computer science, and operations research. Since the family of vehicle routing problems is included in the category of NP-hard problems (Lenstra & Rinnooy Kan, 1981), they are hard to solve, especially when the number of customers is large (Lee, Lee, & Lin, 2008). The richness and difficulty of these problems has made vehicle routing an area of intense research work, as witnessed by the large number of research papers found in the literature (Eksioglu, Vural, & Reisman, 2009). Often, the number of customers combined with the complexity of real-life data does not permit them to be solved using exact methods, which is why current research concentrates on heuristic algorithms that are capable of finding high quality solutions to real-life problems in limited time. In particular, heuristics and meta-heuristics support managers in decision-making with approximate solutions to complex problems (Gendreau & Potvin, 2010).

There are different variants of VRPs that aim to take into account the constraints and details of the problem, while also including different aspects of its nature, such as its dynamicity, time dependency, stochastic aspects, etc. An important variant of the VRP is the Capacitated Vehicle Routing Problem with (hard) Time

☆ This manuscript was processed by Area Editor Iris F.A. Vis.

* Corresponding author. Tel.: +34 958243228; fax: +34 958248993.
E-mail addresses: rbanos@ugr.es (R. Baños), jortega@ugr.es (J. Ortega), cgilm@ual.es (C. Gil), almarquez@ual.es (A.L. Márquez), ftoro@ugr.es (F. de Toro).

0360-8352/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cie.2013.01.007

Windows (VRPTW). It consists in determining the optimal set of routes of a fleet of identical vehicles with restricted capacity so that all customers, whose demands are known, are serviced exactly once within each time window. These time windows impose that the vehicle must begin the service to the customer within the time window defined by the earliest and latest times allowed by the customer for the start of service (El-Sherbeny, 2010). Routing problems are often set up with the single-objective of minimizing the cost of the solution despite the fact that the majority of the real applications associated with this problem are multi-objective in nature (Jozefowiez, Semet, & Talbi, 2008; Dabia, Talbi, van Woensel, & de Kok, 2013).

This paper presents a new Pareto-based multi-objective approach that uses a multi-start simulated annealing strategy for solving a multi-objective formulation of the VRPTW that aims to minimize the total distance of the vehicles used to service the customers, while also minimizing the imbalance of workloads (distances travelled/goods delivered by the vehicles). This new approach is evaluated in comparison with two well-known multi-objective evolutionary algorithms, NSGA-II (Deb, Agrawal, Pratap, & Meyarivan, 2001) and SPEA2 (Zitzler, Laumanns, & Thiele, 2001). Section 2 describes the Capacitated Vehicle Routing Problem with Time Windows, while also justifying the importance of using multi-objective optimization methods to solve real-life routing problems. Section 3 details the multi-objective evolutionary algorithm proposed here, while the results obtained when solving some test problems are commented upon in Section 4. Finally, Section 5 provides the conclusions to this work.

2. The Capacitated Vehicle Routing Problem with Time Windows

Since Dantzig and Ramser (1959) introduced the VRP, it has been one of the most widely analyzed NP-hard problems. The basic VRP has been extended to include aspects such as characteristics of the network, the fleet, and the customers, making the problem more difficult to be solved (Bochtis & Sørensen, 2009). The typical formulation of the single-objective VRPTW involves the routing of a set of vehicles with identical capacity stationed at a central depot (logistic centre) which operate within certain time windows and are used to visit and fully supply the demands of a set of customers. Routes are designed to start and end at the depot and the total demand met by any route cannot exceed the vehicle capacity. The customers, whose demand can only be supplied once by exactly one vehicle, are located in diverse geographical regions and have predefined requirements of goods and a service time. The depot on the one hand, and each customer on the other, have time windows, implying that the vehicle may arrive before the time window opens but not after it has closed, and the customer cannot be serviced until the time windows open. The distances between customers are measured by Euclidean distances, and the total distance travelled by all the vehicles defines the travelling times. Therefore, the single-objective VRPTW aims to determine which customers are visited by each vehicle and the route each vehicle follows to serve the assigned customers, while the distances travelled by the vehicles are minimized and the capacity and time windows constraints are satisfied. The VRPTW has been widely studied because it remains one of the most difficult problems in combinatorial optimization and has a considerable economical impact on all logistic systems (Alvarenga et al., 2007), especially due to the importance of just-in-time production systems and the increasingly tight coordination of supply chain operations (Figliozzi, 2010).

Some exact methods have been proposed for the VRPTW, including Lagrangian relaxation-based methods, column generation, and dynamic programming (El-Sherbeny, 2010). However, exact methods often perform poorly in intermediate and large problem instances, especially in some VRP variants (Kritikos & Ioannou, 2010). As a result, several heuristic and meta-heuristic methods have been proposed for solving the VRPTW, including multi-start local search (Bräysy, Hasle, & Dullaert, 2004), genetic algorithms (Alvarenga et al., 2007), tabu search (Cordeau, Laporte, & Mercier, 2001), etc., and the results obtained show that these methods obtain acceptable results in reduced runtimes.

2.1. Mathematical model for VRPTW

The VRPTW can be modeled on an non-directed complete graph $G(V,E)$ where vertices V={1, ..., N} correspond to the depot and the customers, and edges $e \in E\{(i,j): i,j \in V\}$ to the links between them (El-Sherbeny, 2010).

Decision variable

$$X_{ij}^{k} = \begin{cases} 1 & \text{if vehicle } k \text{ travels from node } i \text{ to node } j \\ 0 & \text{otherwise} \end{cases}$$

Parameters

a_j	is the earliest time for customer j to allow the service,
b_j	is the latest time for customer j to allow the service,
C_{ij}	is the cost for travelling from node i to node j (here, C_{ij} is considered as the distance or time required for travelling from node i to node j),
d_j	is the demand at customer j,
K	is the maximum number of vehicles that can be used,
N	is the number of customers plus the depot (the depot is denoted with number 1, and the customers are denoted as 2, ... N),
Q	is the loading capacity of each vehicle.

The VRPTW can be stated as follows:

$$\text{minimize}: \quad TD = \sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j=1}^{N} X_{ij}^{k} C_{ij} \tag{1}$$

$$\text{subject to} \quad X_{ii}^{k} = 0 \quad (\forall i \in \{1,\ldots,N\}, \quad \forall k \in \{1,\ldots,K\}) \tag{2}$$

$$X_{ij}^{k} \in \{0,1\} \quad (\forall i,j \in \{1,\ldots,N\}, \quad \forall k \in \{1,\ldots,K\}) \tag{3}$$

$$\sum_{k=1}^{K}\sum_{i=1}^{N} X_{ij}^{k} = 1 \quad (\forall j \in \{2,\ldots,N\}) \tag{4}$$

$$\sum_{i=1}^{N}\sum_{j=2}^{N} X_{ij}^{k} d_j \leqslant Q \quad (\forall k \in \{1,\ldots,K\}) \tag{5}$$

$$\sum_{k=1}^{K}\sum_{j=2}^{N} X_{1j}^{k} \leqslant K \tag{6}$$

$$\sum_{j=2}^{N} X_{1j}^{k} - \sum_{j=2}^{N} X_{j1}^{k} = 0 \quad (\forall k \in \{1,\ldots,K\}) \tag{7}$$

$$a_j \leqslant s_{kj} \leqslant b_j \quad (\forall i,j \in [1,\ldots,N], \quad \forall k \in \{1,\ldots,K\}) \tag{8}$$

$$s_{ki} + C_{ij} - L\left(1 - X_{ij}^{k}\right) \leqslant s_{kj} \quad (\forall i,j \in \{1,\ldots,N\}, \quad \forall k \in \{1,\ldots,K\}) \tag{9}$$

Eq. (1) is the objective function of the problem. Eq. (2) denotes that a vehicle must travel from one node to a different one. Eq. (3) indicates that X_{ij}^{k} is equal to 1 if vehicle k goes from node i to node j, and is equal to 0 otherwise. Eq. (4) states that a customer is visited once by exactly one vehicle. By specifying the constraint of Eq. (5), it is taken into account that for a given vehicle k, the load that has to be transported to complete the route assigned to such vehicle cannot exceed its capacity Q. Eq. (6) specifies that there are up to K routes going out of the delivery depot. Eq. (7) guarantees that vehicles depart from and return to the depot. Let s_{kj} be the sum of the distances travelled by vehicle k before arriving customer j. Eq. (8) ensures that time windows are observed. Given a large value, L, the inequality represented in Eq. (9) specifies that, if vehicle k is travelling from customer i to customer j, the vehicle cannot arrive at customer j before $s_{ki} + C_{ij}$. As specified by El-Sherbeny (2010), the variable s_{kj} corresponds to the time vehicle k starts to service customer j. If the vehicle k does not service j, s_{kj} is not calculated.

2.2. Multi-objective VRPTW: related work

Though most research papers in the field of VRP concentrate on the optimization of a single objective, the use of multi-objective optimization is attracting increasing research attention because it offers new opportunities for defining problems, thus extending routing problems with several objectives. Some multi-objective VRPs are formulated and solved by using aggregate approaches where all the objectives to be optimized are included in a single function using a combination of mathematical operations. The drawback of such an approach is that the weights are difficult to determine precisely, particularly when there is insufficient information or knowledge concerning the large real-world vehicle routing problems (Tan, Chew, & Lee, 2006). Therefore, methods that provide a range of solutions representing the trade-offs between objectives are crucial for many real-world applications, and Pareto-based strategies (Goldberg, 1989) are well suited for this purpose. Pareto-optimization methods establish relationships between solutions according to Pareto-dominance relations in the following manner: a solution s_1 dominates (or is preferred to) another solution s_2 ($s_1 \succeq s_2$) if and only if s_1 is better in at least one objective, and not worse in the others. Two solutions are called indifferent if neither of them dominates the other. Since all the objectives are equally important, the aim of multi-objective optimization is to find the Pareto-optimal front or a representative sample of it.

Jozefowiez et al. (2008) classify the different objectives of VRPs according to the component of the problem with which they are associated, i.e. the tour, the resources, and the node/arc activity. Regarding the tour, the most common objective is to minimize the costs of the solutions in terms of distance travelled or time required, while other authors aim to minimize the length of the longest tour (makespan), to reduce the disparity (imbalance) in the tour's workloads of the vehicles, where workloads can be expressed as the number of customers visited, the quantity of goods delivered, the time required, or the tour length, etc. (Jozefowiez, Semet, & Talbi, 2007). The main objectives related to resources focus on the use of the vehicles, minimizing the number of vehicles or maximizing their cost-effectiveness in terms of time or capacity, and on goods, minimizing the deterioration of perishable goods or the perceived risk associated with the transportation of hazardous goods. In reference to the node/arc activity, most studies replace the time windows with an objective that minimizes either the number of violated constraints or the total driver's wait time due to earliness or lateness.

Probably, the most widely studied multi-objective formulation is to simultaneously minimize the number of vehicles used and the travelling distance (Ghoseiri & Ghannadpou, 2010). In fact, many papers dealing with the single-objective VRPTW are based on a two-stage approach where the number of routes is minimized in the first stage and the total travelling distance is then minimized in the second one, i.e. minimizing the number of vehicles is considered the primary objective, while for the same number of vehicles, the total distance travelled is often used as the secondary objective (Bräysy et al., 2004; Nagata, Bräysy, & Dullaert, 2010). However, several practical situations suggest directly considering the total travelling distance as the primary objective: the vehicles and drivers often belong to the company and therefore, their cost is almost independent of their use; the travelling distance has a higher economic impact, since fuel consumption is an important variable of the transportation process (Kuo, 2010). Since travelling time depends on the travelling distance, this objective is more accurate when transporting perishable goods; these two objectives may be positively correlated with each other. The latter reason is especially important from the viewpoint of multi-objective optimization, since both objectives may be positively correlated with each other, or may be conflicting, i.e. fewer vehicles employed in service do not necessarily increase the travelling distance (Tan et al., 2006).

Of the few papers dealing with multi-objective formulations of the VRP that consider workload imbalance, most omit the time windows constraints. For instance, Borgulya (2008) proposed a heuristic procedure for the Capacitated Vehicle Routing Problem without Time Windows that aims to find the route for each vehicle in order for all the demands nodes of the network to be visited in the shortest possible time. Jozefowiez, Semet, and Talbi (2009) uses a meta-heuristic method based on an evolutionary algorithm involving classical multi-objective operators to address a bi-objective vehicle routing problem without time windows, in which the total length of routes is minimized as well as their balance, i.e. the difference between the maximum and the minimum route lengths. Recently, Kritikos and Ioannou (2010) have proposed a variant of the VRPTW that targets the balancing of the load carried by each active vehicle named balanced cargo vehicle routing problem with time windows. Since few papers dealing with multi-objective formulations of the VRPTW that consider the workload imbalance are found in the literature, a major research effort is required in this field.

The present paper aims to solve the vehicle routing problem with hard time windows and workload balancing (here called WB-VRPTW), where not only the total distance travelled by the vehicles of the fleet (travelling distance, TD) is minimized, but also the workload imbalance. The workload imbalance is analyzed from two points of view: the imbalance in the distances travelled (distance imbalance, DI) by the vehicles used, and the imbalance in the loads of these vehicles (load imbalance, LI). These objectives are mathematically defined as follows:

$$minimize: \; DI = max\left(\sum_{i=1}^{N}\sum_{j=2}^{N} X_{ij}^{k} C_{ij}\right) - min\left(\sum_{i=1}^{N}\sum_{j=2}^{N} X_{ij}^{k} C_{ij}\right) \quad (\forall k \in K) \quad (10)$$

$$minimize: \; LI = max\left(\sum_{i=1}^{N}\sum_{j=2}^{N} X_{ij}^{k} d_j\right) - min\left(\sum_{i=1}^{N}\sum_{j=2}^{N} X_{ij}^{k} d_j\right) \quad (\forall k \in K) \quad (11)$$

while the constraints described above (Eqs. (2)–(9)) are satisfied.

3. MMOEASA: a multi-start multi-objective evolutionary algorithm with simulated annealing

Hybridization allows enhancing the strengths and compensating the weaknesses of two or more methods with the aim of generating better solutions by combining the key elements of competing methodologies. Some previous studies have shown the advantages of embedding local search techniques into evolutionary approaches for solving vehicle routing problems (Alba & Dorronso, 2006). With the aim of obtaining high quality non-dominated solutions for the multi-objective VRPTW described above, we propose a new approach (MMOEASA) that aims to improve the efficiency of a Pareto-based multi-objective evolutionary algorithm by using simulated annealing as acceptance criterion. Both meta-heuristics have been hybridized as, this way, it could be possible to take advantage of the evolutionary and local search strategies to reach a procedure with a well balanced exploration–exploitation behavior. In fact, the design of hybrid meta-heuristics based on evolutionary and local search techniques for solving multi-objective formulations of vehicle routing problems is still an open question.

Evolutionary Algorithms (EAs) (Holland, 1975) are general purpose heuristic search methods based on the principles of natural selection and evolution that consist of evolving a population of solutions by applying mutation and crossover operators with certain probabilities ($P_{crossover}$, $P_{mutation}$) in combination with a selection mechanism (binary tournament, roulete mechanism, etc.) that selects the individuals best adapted to the environment. On the other hand, Simulated Annealing (SA) (Kirkpatrick, Gelatt, & Vecchi, 1983) is a local search meta-heuristic that optimizes a solution by exposing it to a high initial temperature, T_i, cooling it by means of a cooling rate $T_{cooling}$ until the temperature falls below a given threshold, T_{stop}. Therefore, better neighboring solutions are always accepted, whereas worse solutions are accepted with a certain probability, which is dependent on the current temperature, t. Since variable t is included within the Metropolis function (Metropolis, Rosenbluth, Rosenbluth, Teller, & Teller, 1953), a decrease of t implies a reduction in the probability of accepting movements that worsen the cost function.

MMOEASA works by improving the population of solutions using the mutation and crossover operators like the evolutionary algorithm. However, instead of accepting or rejecting the solutions in accordance to a typical evolutionary selection mechanism, the modified solutions are accepted in accordance with a modified Pareto-dominance criterion which considers the current temperature and the Metropolis function. MMOEASA uses an external archive (*ND*) to store non-dominated solutions found in the search, and works in a multi-start manner, so that when the termination condition is fulfilled, the annealing parameters are re-initialized in all the solutions, which continue improving the solutions obtained until that moment.

Algorithm 1 describes the general structure of our procedure MMOEASA, where the input parameters are the number of solutions in the population, or population size, p; the number of multi-start iterations, *MS*; the termination condition, *TC* (that can be evaluated as number of iterations, total runtime, etc.); the probabilities for applying crossover and mutation operators, respectively $P_{crossover}$ and $P_{mutation}$; and the minimum, maximum, and stop temperatures, respectively T_{min}, T_{max}, *and* T_{stop}. Each solution of the population, I_i, is initialized (line 3 in Algorithm 1) according to the time window based insertion heuristic, TWIH (), described in Section 3.2. Then (lines 4 and 5), an annealing scheme defined by the initial temperature, $I_i \cdot T_i$, and the cooling factor, $I_i{\cdot}T_{cooling}$, is assigned to each solution I_i. In particular, $I_i \cdot T_{cooling}$ is obtained by using the procedure Calculate_cooling ($I_i \cdot T_i$, *TC*) described in Section 3.6. Then, the main loop (lines 7–18) is repeated for each multi-start iteration, that starts by reinitializing the current temperature, $I_i \cdot t$, used by each solution I_i (lines 8 and 9). The inner loop of Algorithm 1 (lines 10–17) evolves the population by applying crossover (line 12) and mutation (line 13) operators described in Section 3.3. The changes in the solutions are accepted or not (line 14) according to the multi-objective Metropolis criterion, MO_Metropolis (), described in Section 3.4. Then, a modified solution is included in the external archive, *ND*, if it is not dominated by any of the solutions previously included in *ND* (lines 15 and 16), i.e. if there is no solution of the external set, ND_j, that improves the modified solution, I_i, in one objective and, at the same time, ND_j is not worse than I_i in the other one. Finally, the temperature is updated by using the cooling rate (line 17) and a new multi-start is considered (line 18). Once the main loop finishes, the non-dominated set, *ND*, is returned (line 19).

Algorithm 1. MMOEASA

1. Input: p, *MS*, *TC*, $P_{crossover}$, $P_{mutation}$, T_{min}, T_{max}, T_{stop};
2. For each $I_i \in P$
3. $I_i \leftarrow$ TWIH ();
4. $I_i \cdot T_i \leftarrow T_{min} + i * ((T_{max} - T_{min})/p)$;
5. $I_i \cdot T_{cooling} \leftarrow$ Calculate_cooling ($I_i \cdot T_i$, *TC*);
6. *current_multi-start* = 1;
7. While (*current_multi-start* $\leqslant$ *MS*)
8.   For each $I_i \in P$
9. $I_i \cdot t \leftarrow I_i \cdot T_i$;
10.   While ($I_i \cdot t \leqslant T_{stop}$, $\forall I_i \in P$)
11.   For each $I_i \in P$
12. $I'_i \leftarrow$ Crossover $(I_i, P_{crossover})$;
13. $I''_i \leftarrow$ Mutation $(I'_i, P_{mutation},$ random $(1, 10))$;
14. $I_i \leftarrow$ MO_Metropolis $(I_i, I''_i, I_i \cdot t)$;
15.   If ($\nexists ND_j \succeq I_i\ \forall ND_j \in ND$)
16. $ND \leftarrow ND \cup I_i$;
17. $I_i \cdot t \leftarrow I_i \cdot t * I_i \cdot T_{cooling}$;
18.   *current_multi-start* = *current_multi-start* + 1;
19. Return (*ND*);

3.1. Representation

MMOEASA works with a population P={I_1, I_2, ..., I_P}, where each solution represents the routes travelled by K vehicles used to deliver goods to the demand nodes (customers). Therefore, each individual, I_i, is represented by a set of chromosomes, C_{ik}, which consists of a variable number of genes, $C_{ik} = \left\{1, G^1_{ik}, G^2_{ik}, \ldots, G^l_{ik}, 1\right\}$ representing the route of the kth vehicle in the ith individual, with $2 \leqslant G^j_{ik} \leqslant N$. For example, $C_{7,5}$ = {1, 12, 35, 3, 84, 57, 1}, indicates that the fifth vehicle of the seventh individual departs from the depot and visits customers 12, 35, 3, 84, and 57, before returning to the depot, which is represented by the identificator 1.

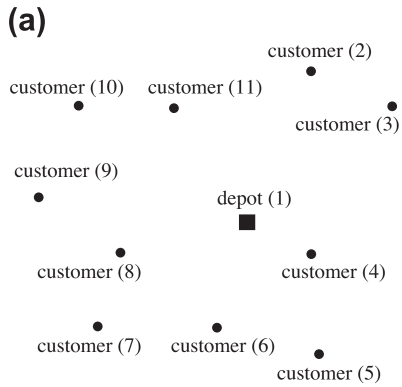
3.2. Initialization

The initial routes are built by using a time window based insertion heuristic (TWIH () in Algorithm 1), which is graphically described in Fig. 1. This heuristic sequentially constructs the routes by first visiting those customers with the earliest time in their time windows, i.e. those available soonest, which are inserted in the vehicles whenever the time windows and capacity constraints are fulfilled. This procedure is very fast (less than one second in all test instances) because it is applied only once in one individual, then copying the routes (solution) in the remaining ones.

3.3. Evolutionary operators

A large number of evolutionary search operators have been proposed to solve the VRP (Tan et al., 2006; Alvarenga et al., 2007; El-Sherbeny, 2010; Garcia-Najera & Bullinaria, 2011). In order to increase the diversification in the search, MMOEASA uses a total of ten often used mutation operators to solve VRPs. These mutation operators are described below:

- Customer random reallocation: a vehicle (route) and a random customer associated with it are chosen; the reallocation of this customer to another position of the same vehicle is tried, and it is accepted if the new solution is feasible.



(b)

NODE	a_j	b_j	visiting order
1	0	--	--
2	3	6	2
3	5	15	3
4	2	10	1
5	10	17	8
6	14	18	9
7	15	19	10
8	9	13	7
9	8	12	6
10	8	16	5
11	7	11	4

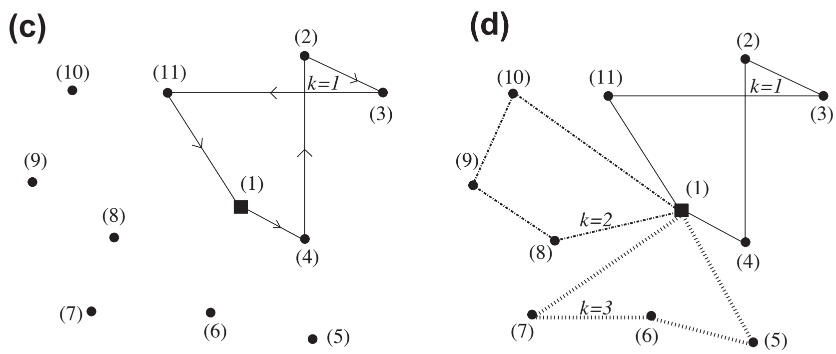


Fig. 1. Time window based insertion heuristic: (a) Spatial location of the depot and customers ($N = 11$). (b) Customers' time windows and visiting order. (c) Customers visited by the first vehicle ($k = 1$). (d) Initial solution obtained by TWIH( ).

- Customer best reallocation: a vehicle and a random customer associated with it are chosen; the reallocation of this customer between all the pairs of subsequent customers of its vehicle is analyzed, then inserting the customer in the feasible position with the highest positive gain (in the multi-objective case, this means reallocating the customer in a position that creates a new solution that dominates the previous one).
- Customer random migration: a vehicle and a random customer associated with it are chosen; the reallocation of this customer to another non-empty vehicle is tried, and is accepted if it is feasible.
- Customer best migration: a vehicle and a random customer associated with it are chosen; the reallocation of this customer between all the pairs of subsequent customers of other non-empty vehicle is analyzed, then inserting the customer in the feasible position with the best fitness (in the multi-objective case this means reallocating the customer in a position that creates a new solution that dominates the previous one).
- Customers random exchange: two vehicles and a random customer associated with each of them are chosen, then exchanging the customers if it is feasible.
- Customers best exchange: two vehicles are chosen, then exchanging the pair of customers with the highest positive gain (in the multi-objective case, this means exchanging a pair of customers that generate a new solution that dominates the previous one).
- Customers exchange with coincident time-window: a vehicle and a random customer associated with it are chosen. The exchange of this customer with that customer from other vehicles who has coincident time windows, i.e. more or less the same start time and end time, is analyzed, accepting the exchange if it is feasible.
- Route partition: a vehicle and a random customer associated with it are chosen. The route covered by that vehicle is then divided in two, so that those customers originally visited earlier to that random customer are visited by the same vehicle, while that random customer and those visited later are now assigned to a non-used vehicle.
- New route: a vehicle and a random customer associated with it are chosen, then assigning that random customer as the first and only customer of one of the non-used vehicles of the fleet.
- Route elimination:a vehicle is randomly chosen and, one after the other, all of the customers are reallocated in feasible positions of other routes.

On the other hand, the crossover operator consists in taking a given number of routes from the first parent, which are copied in the child solution. After that, all the routes of the second parent that are not in conflict with the previously inserted ones are also included in the child solution. The customers not visited by any route (in the child individual) are inserted in the existing routes, starting from the shortest ones. If this insertion is not possible in one of the existing routes without violating the constraints, a new route is created, which is also considered for insertion of other non-assigned customers.

3.4. Acceptance criterion

In the single-objective context, the traditional Metropolis function is used to accept/reject solutions according to the variation of the fitness value of a solution and the current temperature (Metropolis et al., 1953), but in the multi-objective context, it is necessary to consider other objectives. Therefore, the acceptance criterion in the multi-objective case, here called multi-objective Metropolis criterion (MO_Metropolis), is applied as follows: an offspring solution obtained from the application of mutation or crossover operators is directly accepted if it is not dominated by its

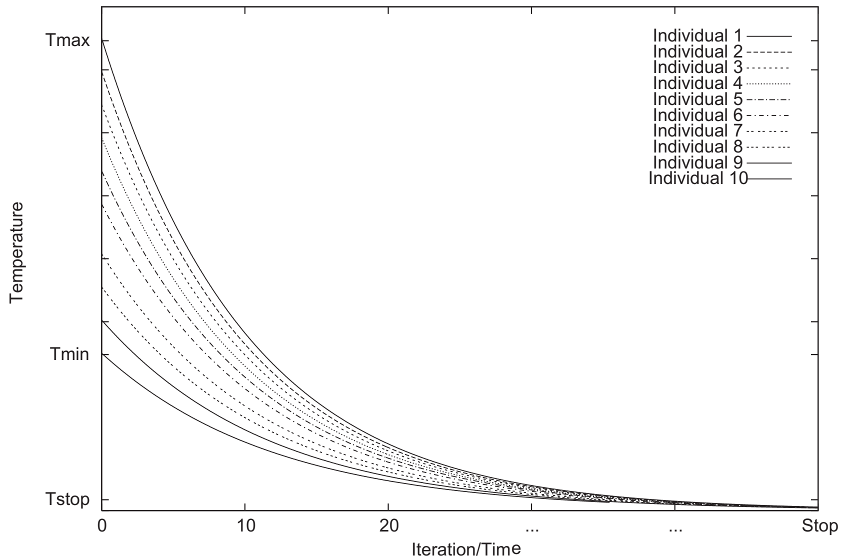


Fig. 2. Sample annealing scheme using different values of Ti and $T_{cooling}$.

parent, while if the parent dominates the child solution, the latter is accepted according to the Metropolis function, where the sum of the fitness value variations in the objectives considered and the current temperature are included as parameters.

3.5. External archive

While the mutation and crossover operators are independent of the number of objectives to be optimized, the selection in the multi-objective case cannot be applied using typical evolutionary search operators (e.g. tournament or roulette-based selection), but rather using Pareto-dominance relations. Our hybrid approach uses an external archive to store non-dominated solutions found in the search. Thus, after applying the mutation or crossover operator to a given individual, the new solution is compared with those individuals of the external archive, so that, if it is not dominated by any of the solutions located in the archive, it is inserted, then removing those which were previously stored in the external archive but which are now dominated by the new solution. When the number of individuals exceeds the size of the external archive, a truncation procedure that guarantees the preservation of boundary solutions is applied.

3.6. Multi-temperature annealing scheme

The use of different annealing parameters in the same run accommodates the random effect of using a fixed annealing scheme (Baños, Gil, Ortega, & Montoya, 2004). Therefore, an interval of initial temperatures, $[Ti_{min}, Ti_{max}]$, and a termination criterion, TC (number of evaluations, total runtime, etc.), are established so solution P_1 starts in Ti_{min}, solution P_p starts in Ti_{max}, and the others are equally distributed along this interval. Then, $T_{cooling}$ is computed for each solution in function of its initial temperature and the stop condition. Fig. 2 shows an example of this strategy using ten different solutions.

4. Empirical analysis

4.1. Test problems

Solomon's problems (Solomon, 1987) have been extensively used for benchmarking different algorithms for VRPTW over the years since they represent relatively well different kinds of routing scenarios (Tan et al., 2006). Solomon's benchmark problems include a total of 56 instances, divided into six groups of between 8 and 12 problem instances. Each benchmark instance includes the fleet (maximum number of vehicles and their capacities) and customers characteristics (geometric coordinates, demand, time-windows and service time). Each of the 56 benchmark instances contains a total of 100 customers and one central depot. The depot service time and demand are zero, the vehicle fleet is homogeneous and the capacity is defined according to the data set. The distance between two customers is calculated from their Cartesian coordinates using Euclidean distances, which are also used as time units. Each customer has a given time window which represents the time interval to arrive at that customer. The capacity of the vehicles and the customer's demands are defined in each class of instance. These six categories are named C1, C2, R1, R2, RC1, and RC2, where the alphabetic characters indicate the spatial customer distribution and the number indicates the time windows constraints. In reference to the spatial customer location, problem category R has all customers located randomly, problem category C has all customers clustered in groups, and problem category RC has a mixture of random and clustered customers. Therefore, customers are located closer to each other in problem category C than in R, implying the latter category a longer total travelling distance, and making it more difficult to be solved, since a small change in the sequence of visited customers of a given route may result in larger differences in terms of the total routing distance. The problems in category 1 (C1, R1, and RC1) have smaller time windows than those in category 2 (C2, R2, and RC2), implying that some candidate solutions are more likely to become unfeasible after a small change in the sequence of visited customers in problems of category 1 than in those of category 2. Furthermore, the time window is smaller at the depot for the instances of group 1 than for those of group 2, which means that a smaller number of customers can be served by one vehicle in benchmarks belonging to the group 1. In each group of problems, the customers geographical distribution, demand, and service time do not change, while they differ in the percentage of customers with time windows and the time windows intervals.

4.2. Performance measures

It is well-known that to evaluate the performance of optimization algorithms, two main criteria have to be taken into account:

computational cost and the quality of the result. While the computational cost is often measured in terms of runtime or number of evaluations of the fitness function, the quality is more difficult to be determined, especially in the multi-objective context. Several authors have proposed a large number of measures to compute the performance of multi-objective algorithms, which can be classified into two main classes: metrics for diversity and metrics for convergence. In terms of diversity, some measures, including the so called spacing metrics, have been proposed to measure how well distributed the non-dominated solutions are in the objective space. In terms of convergence, the goal of Pareto-based multi-objective optimization is to obtain a set of non-dominated solutions as close as possible to the Pareto-optimal front. In some optimization problems, the solutions are known beforehand and, therefore, it is possible to compute their true Pareto front, but unfortunately, in other complex optimization problems, such as NP-hard problems, the true Pareto-front is unknown. Since the VRP is included in the NP-hard category, it is therefore almost impossible to exactly describe what a good approximation is in terms of a number of criteria such as closeness to the Pareto set. An interesting way to overcome this drawback is to construct a quasi-optimal Pareto front by executing one or several algorithms over a very large number of iterations. In this paper, the three evaluated methods have been executed for 3 h for each Solomon's problem. After that, for each benchmark, the three non-dominated fronts returned by the algorithms have been combined, discarding the dominated solutions and maintaining the non-dominated ones. Therefore, this composed non-dominated front is later used as a reference set to evaluate the performance of the methods analyzed here.

Since both the convergence and the diversity are important (Deb, 2002; Coello, Lamont, & Van Veldhuizen, 2007), the quality of the sets of non-dominated solutions obtained here is evaluated using three metrics: the set coverage and the hyper-volume, that are based on ideas taken from the metrics proposed by Zitzler and Thiele (1999) and have been recently used for multi-objective VRPTW (Garcia-Najera & Bullinaria, 2011), and the spacing metric proposed by Deb et al. (2001). Since the objectives to be optimized have different scales (e.g. the travelling distance values are larger than the imbalance ones), the values returned by the three metrics have been normalized in the interval [0, 1]. A brief description of these metrics is now provided.

- Hyper-volume (*HV*): Let $A = (x_1, x_2, \ldots, x_n)$ be a set of non-dominated solutions. The function *HV*(A) returns the area of the objective space bounded by a reference point, x_{ref}, dominated by at least one of the members of *A*. Therefore, the bigger *HV*(A) is, the better *A* is. The reference point considered in our experiments is twice the fitness in each objective corresponding to the initial solution obtained by the time window based insertion heuristic, i.e. $x_{ref} = (2{*}TD(I_i),\ 2{*}TI/LI(I_i))$. For instance, if the solutions of Fig. 3a are considered, it is observed that HV(A) > HV(B), as it can be seen in Fig. 3b and c.
- Set coverage (*SetC*): Let A and B be two sets of non-dominated solutions. The function *SetC* maps the ordered pair (A, B) to a value *z* within the interval [0, 1] that represents the percentage of solutions of B dominated by at least one solution of A. Therefore, the bigger *SetC*(A, B) is, the better A is with respect to B. It is noted that *SetC*(A, B) = *z* does not imply that *SetC*(B, A) = 1 − *z*. For instance, if the solutions of Fig. 3a are considered, *SetC*(A, B) = 0.666, and *SetC*(B, A) = 0.125, i.e. 66.6% of solutions of non-dominated front B are dominated by at least one solution of A, while only 12.5% of solutions of A are dominated by at least one solution of B.
- Spacing metric (SM): This metric measures how well distributed the non-dominated solutions are in the objective space (Deb et al., 2001). The smaller the value of the spacing metric, the better are the solutions spread along the obtained front. Fig. 3b and c denotes that A is better distributed than B, i.e. SM(A) < SM(B).

4.3. Parameter settings

As commented above, MMOEASA has been compared with NSGA-II and SPEA2. The NSGA-II algorithm (Deb et al., 2001) is a multi-objective evolutionary algorithm that uses two populations in the search process, and applies mechanisms to include elitism pressure and preserving the diversity. The elitism is controlled by a non-dominated sorting procedure that creates a non-dominance based hierarchy among solutions. The diversity of the population is maintained according to a crowding comparison operator that estimates the density of the individuals in the solution space. NSGA-II has been recently used for solving multi-objective formulations of the VRPTW (Scheffermann, Bender, & Cardeneo, 2009). On the other hand, SPEA2 (Zitzler et al., 2001) is an improved version of SPEA (Zitzler & Thiele, 1999) that incorporates a fine-grained fitness assignment strategy, which takes into account, for each individual, the number of individuals that dominate it, and the number of individuals dominated by it. It applies a nearest neighbor density estimation technique that guides the search more efficiently by avoiding crowding on the front, while also using an external archive to store promising solutions found in the search, always preserving the boundary solutions.

All the methods use a population of 40 individuals ($|P| = 40$), and the archive is also of the same size ($|ND| = 40$). Each of the individuals of the population is randomly initialized using one of the time window based insertion heuristic described above. In all iterations, mutation, crossover, and selection operators are applied. The probability of applying mutation and crossover is 25%. If mutation is applied, each of the 10 mutation variants is applied with the

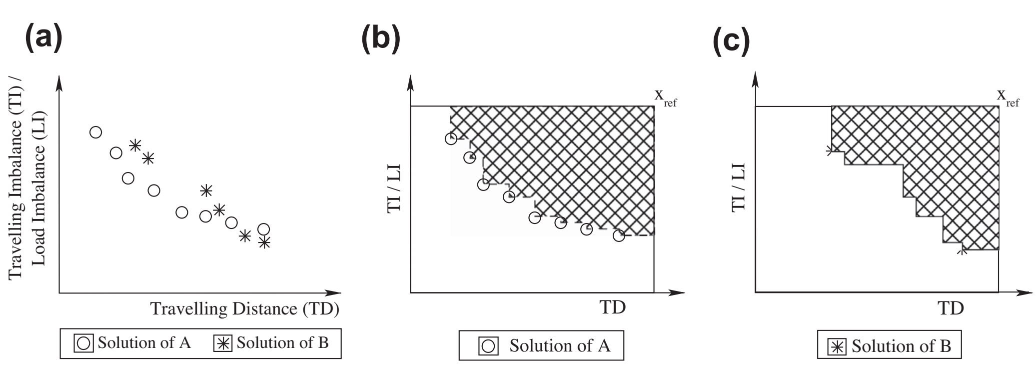


Fig. 3. Graphical explanation of the metrics used over two non-dominated sets.

same probability (10%). To provide reliable statistics, each version of the algorithm was run 25 times for each benchmark instance, thus obtaining 25 non-dominated fronts for each algorithm and benchmark. The median front in terms of hyper-volume metric was then selected as the representative front. The performance of the multi-objective evolutionary algorithms here analyzed is tested on two of the problem instances of each of the six groups that form the 56 Solomon's benchmarks with 100 customers: C103, C108, C203, C208, R103, R108, R203, R208, RC103, RC108, RC203, and RC208, which are representative of all C/R/RC-types. These tests were run on an Intel Core processor with 2.27 GHz and 8 GB of RAM memory. When comparing different algorithms, it is possible to determine that one technique is better than another if it obtains a better performance given the same amount of computational cost. Therefore, the computational times are used here to compare the algorithms.

4.4. Results and discussion

As commented above, the bi-objective formulation of the WB-VRPTW can be stated as minimizing the travelling distance (TD) and one of the workload imbalance criteria: the distance imbalance (DI) or load imbalance (LI). The first analyzed aspect is the quality of the solutions obtained by MMOEASA, NSGA-II, and SPEA2 when solving the multi-objective formulation that minimizes the total travelling distance and the distance imbalance among vehicles (TD + DI). Table 1 shows the best and median hyper-volume of the 25 independent runs obtained by each method when solving Solomon's instances of type R (R103, R108, R203, and R208) considering as stop criterion a total runtime of 20, 40, and 80 s. The best values are marked with bold fonts. Two main conclusions are obtained from these results: the quality of the solutions tends to improve when the total runtime grows; the median hyper-volume obtained by MMOEASA outperforms SPEA2 and NSGA-II in almost all cases.

Table 2 shows the median values obtained by all the methods categorized according to the multi-objective formulation (TD + DI)/(TD + LI), the execution time and Solomon's type of problem. It is observed that the normalized hyper-volume obtained by MMOEASA is higher in all groups of benchmarks. It is noteworthy to remark that the improvement with respect to SPEA2 and NSGA-II is especially notable in test problems with larger time windows (category 2), but MMOEASA also outperforms NSGA-II and SPEA2 in problems with smaller time windows (category 1). The improvement is even more marked in those executions with large runtimes.

The hyper-volume metric is able to include in a single scalar value both the closeness of the solutions to the Pareto-optimal set and, to some extent, the spread of the solutions across the objective space (While, Hingston, Barone, & Huband, 2006). Nevertheless, the set coverage is also used to reinforce the conclusions. Table 3 summarizes the set coverage relationships among the three multi-objective algorithms here compared using the subset of 12 Solomon's benchmark instances enumarated abovee. The median front, in terms of hyper-volume, of the previous executions with 80 s is used in this comparison. In reference to the bi-objective formulation that aims to minimize the travelling distance and the distance imbalance, on average 88.5% and 77.7% of the non-dominated solutions obtained by NSGA-II and SPEA2 are dominated by solutions of MMOEASA, respectively, while these methods are able to dominate only 1.7% and 16.6% of the solutions of MMOEASA. On the other hand, it is also observed that SPEA2 dominates a greater percentage of solutions of NSGA-II (87.6%) than viceversa (1.5%). On average, MMOEASA covers 83.1% of the solutions of the other methods, while the latter only dominate 9.2% of those found by the former. Similar conclusions can be obtained from the results obtained in the bi-objective formulation that aims to minimize the travelling distance and the load imbalance, where, on average, 87.2% and 66.1% of the non-dominated solutions obtained by NSGA-II and SPEA2 are dominated by solutions of MMOEASA, while these methods are able to dominate only 4.5% and 30.4% of the solutions of MMOEASA, respectively. Therefore, these results reinforce the conclusions obtained from the hyper-volume metric regarding the good performance of MMOEASA.

With respect to Deb's spacing metric, Table 4 summarizes the results obtained by all the algorithms when the runtime is 80 s. Thus, the average distribution of the non-dominated solutions obtained by MMOEASA in the first multi-objective formulation is 0.0026, i.e. 0.26% of the maximum normalized distance between the reference point and the origin of coordinates. In reference to the second multi-objective formulation, the spacing values corresponding to the solutions obtained by MMOEASA and SPEA2 are also better than those obtained by NSGA-II.

Table 1
HV when minimizing travelling distance (TD) and distance imbalance (DI).

	20 s		40 s		80 s	
	HV_{best}	HV_{median}	HV_{best}	HV_{median}	HV_{best}	HV_{median}
R103						
NSGA-II	92.71	86.96	93.34	90.33	95.01	90.73
SPEA2	89.50	86.87	94.26	90.69	98.84	95.13
MMOEASA	91.22	**87.10**	94.26	**91.33**	98.40	**95.44**
R108						
NSGA-II	93.62	87.78	96.32	92.11	97.86	94.10
SPEA2	92.59	**88.38**	95.33	91.59	98.04	95.27
MMOEASA	92.45	88.26	96.02	**92.48**	98.41	**96.80**
R203						
NSGA-II	88.40	81.53	89.73	81.64	91.74	81.69
SPEA2	90.75	86.60	95.56	91.69	98.44	96.13
MMOEASA	94.70	**91.70**	97.59	**96.20**	99.55	**98.21**
R208						
NSGA-II	92.79	82.43	91.22	83.39	92.08	83.51
SPEA2	91.01	88.92	94.79	92.57	98.34	96.26
MMOEASA	95.12	**92.58**	98.15	**96.08**	99.09	**98.61**

Table 2
Median hyper-volume obtained by all the methods in both formulations.

	Type R	Type C	Type RC	Type 1	Type 2	AVG
TD + DI 20 s						
NSGA-II	84.68	83.46	83.61	85.44	82.39	83.92
SPEA2	87.69	85.66	85.12	85.70	86.61	86.16
MMOEASA	**89.91**	**87.43**	**86.88**	**85.96**	**90.19**	**88.07**
TD + LI 20 s						
NSGA-II	86.79	87.14	85.44	84.13	88.78	86.45
SPEA2	85.92	88.54	85.12	84.65	88.40	86.53
MMOEASA	**88.82**	**92.08**	**88.87**	**86.59**	**93.25**	**89.92**
TD + DI 40 s						
NSGA-II	86.87	84.95	85.25	88.35	83.03	85.69
SPEA2	91.64	89.62	89.18	89.94	90.35	90.14
MMOEASA	**94.02**	**92.44**	**91.20**	**90.90**	**94.21**	**92.55**
TD + LI 40 s						
NSGA-II	90.54	89.26	88.40	87.74	91.06	89.40
SPEA2	90.15	92.21	89.80	89.21	92.23	90.72
MMOEASA	**93.66**	**95.74**	**93.50**	**92.04**	**96.56**	**94.30**
TD + DI 80 s						
NSGA-II	87.51	85.86	86.48	89.54	83.70	86.62
SPEA2	95.70	93.93	93.64	94.24	94.60	94.42
MMOEASA	**97.27**	**95.37**	**95.05**	**94.62**	**97.17**	**95.90**
TD + LI 80 s						
NSGA-II	92.21	91.47	89.80	89.96	92.37	91.16
SPEA2	94.91	**97.08**	94.36	94.80	96.10	95.45
MMOEASA	**96.52**	96.85	**95.71**	**95.16**	**97.58**	**96.37**

Table 3
Set coverage obtained by all the methods (80 s).

	NSGA-II		SPEA2		MMOEASA		AVG	
	TD + DI	TD + LI	TD + DI	TD + LI	TD + DI	TD + LI	TD + DI	TD + LI
NSGA-II			0.015	0.016	0.017	0.045	0.016	0.031
SPEA2	0.876	0.927			0.166	0.304	0.355	0.616
MMOEASA	0.885	0.872	0.777	0.661			**0.831**	**0.767**
AVG	0.881	0.900	0.396	0.339	**0.092**	**0.175**		

Table 4
Summary of results using Deb's spacing metric (80 s).

	Type R	Type C	Type RC	Type 1	Type 2	AVG
TD + DI						
NSGA-II	0.0063	0.0057	0.0087	0.0085	0.0053	0.0069
SPEA2	0.0028	0.0027	0.0040	0.0043	0.0020	0.0032
MMOEASA	**0.0024**	**0.0023**	**0.0032**	**0.0039**	**0.0013**	**0.0026**
TD + LI						
NSGA-II	0.0172	0.0298	0.0131	0.0250	0.0151	0.0201
SPEA2	**0.0103**	**0.0235**	0.0097	**0.0172**	0.0119	0.0145
MMOEASA	0.0106	0.0241	**0.0085**	0.0193	**0.0095**	**0.0144**

Fig. 4 displays the fronts of non-dominated solutions obtained by all the methods in the optimization of benchmark instances R108 and RC203. Fig. 4a shows that MMOEASA and SPEA2 are closer to the (unknown) Pareto-optimal front than NSGA-II, while Fig. 4b shows how the non-dominated solutions obtained by MMOEASA clearly improve those obtained by SPEA2 and NSGA-II. Fig. 5 shows the results obtained by all the methods in the optimization of benchmark instances c103 and rc108 when solving the second multi-objective formulation. Fig. 5a shows that MMOEASA obtains a good non-dominated set, not only in terms of dominance, but also in terms of distribution of the solutions. Similar conclusions are obtained from Fig. 5b, where all the non-dominated solutions found by NSGA-II and SPEA2 are dominated by those obtained by MMOEASA. Let us notice that, graphically, the solutions obtained in this multi-objective formulation seem to be better distributed, as in Solomon's benchmarks the customers' demand is an integer value and, therefore, the load imbalance has also discrete values, while in the case of minimizing the distance imbalance of the vehicles, the Euclidean distances return real values, which means that the Pareto fronts do not seem to be so well distributed.

Finally, it is noticed that the good results obtained by MMOEASA in the multi-objective WB-VRPTW reinforce the previous conclusions of a recent study (Baños, Ortega, Gil, Fernández, & De Toro, 2011) that showed the good performance of a multi-start evolutionary algorithm with simulated annealing for solving the single-objective formulation of the VRPTW that only minimizes the travelling distance. Even though we realize that the multi-objective algorithms are not specifically designed for solving the single-objective case, Figs. 4 and 5 show that MMOEASA obtains solutions with a travelling distance close to the best known solutions (Nagata et al., 2010).

5. Conclusions

Over the past decades, the design of methods to solve vehicle routing problems has been an area of research that has attracted much attention, due to its influence in transportation, logistics, and supply chain management. Often, the number of customers combined with the complexity of real-life data does not permit the problem to be solved exactly, and heuristic methods must therefore be applied. These include heuristic approaches, which provide approximate solutions in runtimes which are acceptable for transportation and logistic managers. This paper presents and analyzes a new Pareto-based hybrid meta-heuristic, MMOEASA, for solving multi-objective formulations of the VRPTW, here called WB-VRPTW, which considers not only the minimum distance required to deliver goods, but also the workload imbalance, in terms

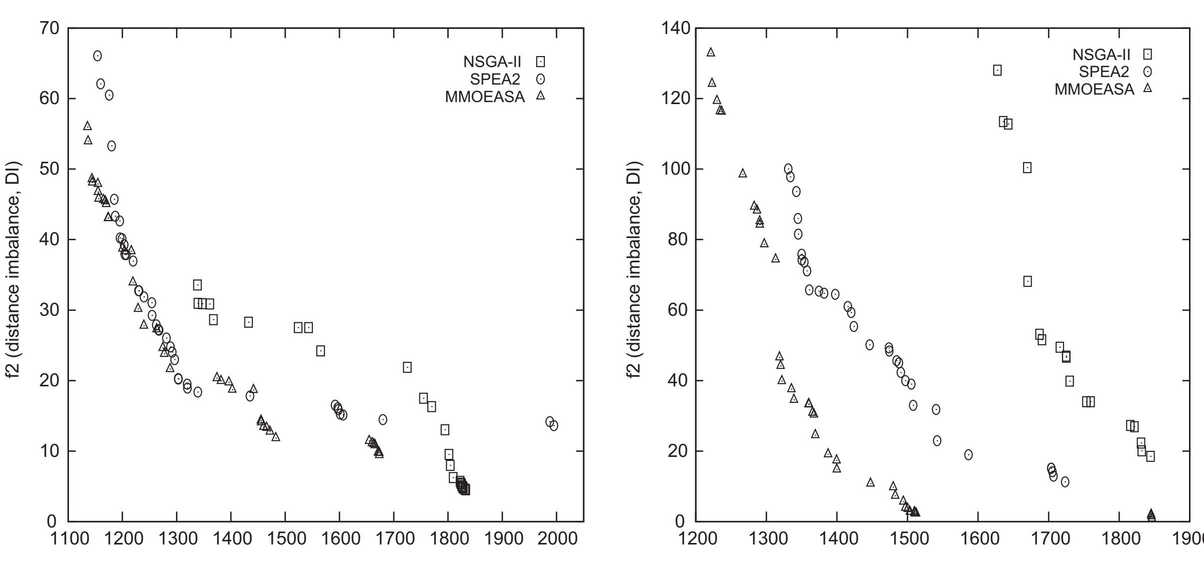


Fig. 4. Fronts of non-dominated solutions obtained in R108 and RC203 (minimizing travelling distance and distance imbalance).

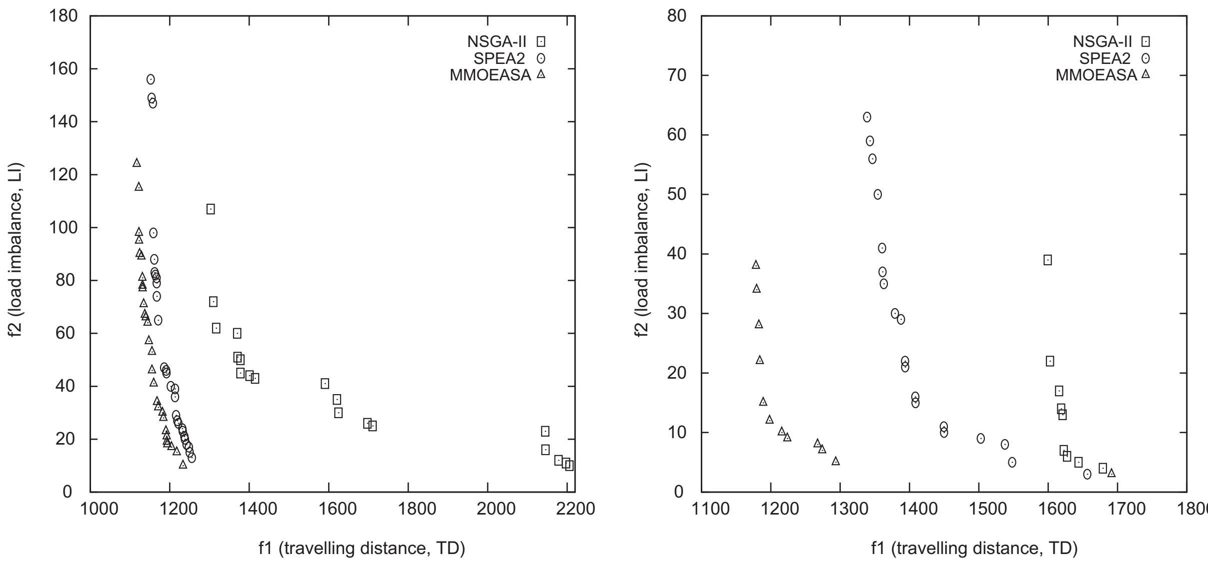


Fig. 5. Fronts of non-dominated solutions obtained in R108 and RC203 (minimizing travelling distance and load imbalance).

of distances travelled and loads transported by the vehicles. This hybrid approach combines evolutionary computation and simulated annealing in a multi-start manner with the aim of providing a set of non-dominated solutions which can be later analyzed by the decision maker according to particular criteria. The performance of this hybrid approach is compared with two popular multi-objective evolutionary algorithms, NSGA-II and SPEA2, when solving a some of the Solomon's benchmark instances. The main contribution of this work is that the hybrid approach achieves highly competitive non-dominated fronts in terms of hyper-volume, set coverage, and spacing metrics, outperforming those obtained by NSGA-II and SPEA2. Furthermore, this research may prove useful to those researchers interested in solving multi-objective formulations of the vehicle routing problem and its variants, thus increasing the application of this problem to the real world.

Acknowledgements

The authors thank the anonymous reviewers for their useful suggestions and comments. This work was supported by the Spanish Ministry of Science and Innovation under contracts TIN2008-01117 and TIN2012-32039. R. Baños also acknowledges the support of a Juan de la Cierva postdoctoral fellowship.

References

Alabas-Uslu, C., & Dengiz, B. (2011). A self-adaptive local search algorithm for the classical vehicle routing problem. *Expert Systems with Applications, 38*(7), 8990–8998.

Alba, E., & Dorronso, B. (2006). Computing nine new best-so-far solutions for capacitated VRP with a cellular genetic algorithm. *Information Processing Letters, 98*(6), 225–230.

Alvarenga, G. B., Mateus, G. R., & de Tomic, G. (2007). A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows. *Computers & Operations Research, 34*(6), 1561–1584.

Baños, R., Ortega, J., Gil, C., Fernández, A., & De Toro, F. (2011). A multi-start hybrid algorithm for vehicle routing problems with time windows. *Online Conference on Soft Computing in Industrial Applications (WSC 16).*

Baños, R., Gil, C., Ortega, J., & Montoya, F. G. (2004). A parallel multilevel metaheuristic for graph partitioning. *Journal of heuristics, 10*(3), 315–336.

Bochtis, D. D., & Sørensen, C. G. (2009). The vehicle routing problem in field logistics: part I. *Biosystems Engineering, 104*(4), 447–457.

Borgulya, I. (2008). An algorithm for the capacitated vehicle routing problem with route balancing. *Central European Journal of Operations Research, 16*(4), 331–343.

Bräysy, O., Hasle, G., & Dullaert, W. (2004). A multi-start local search algorithm for the vehicle routing problem with time windows. *European Journal of Operational Research, 159*(3), 586–605.

Coello, C. A., Lamont, G. B., & Van Veldhuizen, D. A. (2007). *Evolutionary algorithms for solving multi-objective problems. Genetic and evolutionary computation series*. Springer.

Cordeau, J-F., Laporte, G., & Mercier, A. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society, 52*(8), 928936.

Dabia, S., Talbi, E-G., van Woensel, T., & de Kok, A. G. (2013). Approximating multi-objective scheduling problems. *Computers & Operations Research*. http://dx.doi.org/10.1016/j.cor.2012.12.001.

Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management Science, 6*(1), 80–91.

Deb, K. (2002). *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons.

Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2001). *A fast elitist non-dominated sorting genetic algorithm for multiobjective optimization: NSGA-II. Lecture notes in computer science 1917* (pp. 849–858). Springer.

Eksioglu, B., Vural, A. V., & Reisman, A. (2009). The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering, 57*(4), 1472–1483.

El-Sherbeny, N. A. (2010). Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods. *Journal of King Saud University (Science), 22*(3), 123–131.

Figliozzi, M. A. (2010). An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows. *Transportation Research Part C, 18*(5), 668–679.

Garcia-Najera, A., & Bullinaria, J. A. (2011). An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows. *Computers & Operations Research, 38*(1), 287–300.

Gendreau, M., & Potvin, J-Y. (Eds.). (2010). *Handbook of metaheuristics. International series in operations research & management science* (vol. 156, pp. erlag). Rome: Springer.

Ghoseiri, K., & Ghannadpou, S. F. (2010). Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm. *Applied Soft Computing, 10*(4), 1096–1107.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning*. New York: Addison Wesley.

Holland, J. (1975). *Adaptation in natural and artificial systems*. MIT Press.

Hosny, M. I., & Mumford, C. L. (2010). The single vehicle pickup and delivery problem with time windows: Intelligent operators for heuristic and metaheuristic algorithms. *Journal of Heuristics, 16*(3), 417–439.

Jozefowiez, N., Semet, F., & Talbi, E-G. (2007). Target aiming Pareto search and its application to the vehicle routing problem with route balancing. *Journal of Heuristics, 13*(5), 455–469.

Jozefowiez, N., Semet, F., & Talbi, E-G. (2008). Multi-objective vehicle routing problems. *European Journal of Operational Research, 189*(2), 293–309.

Jozefowiez, N., Semet, F., & Talbi, E-G. (2009). An evolutionary algorithm for the vehicle routing problem with route balancing. *European Journal of Operational Research, 195*(3), 761–769.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*(4598), 671–680.

Kritikos, M. N., & Ioannou, G. (2010). The balanced cargo vehicle routing problem with time windows. *International Journal of Production Economics, 123*(1), 42–51.

Kuo, Y. (2010). Using simulated annealing to minimize fuel consumption for the time dependent vehicle routing problem. *Computers & Industrial Engineering, 59*(1), 157–165.

Lee, C-Y., Lee, Z-J., & Lin, S-W. (2008). An enhanced ant colony optimization (EACO) applied to capacitated vehicle routing problem. *Applied Intelligence, 32*, 88–95.

Lenstra, J. K., & Rinnooy Kan, A. H. G. R. (1981). Complexity of vehicle routing and scheduling problems. *Networks, 11*, 221–227.

Mester, D., & Bräysy, O. (2007). Active-guided evolution strategies for large-scale capacitated vehicle routing problem. *Computers & Operations Research, 34*(10), 2964–2975.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A., & Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics, 21*(6), 1087–1092.

Nagata, Y., Bräysy, O., & Dullaert, W. (2010). A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. *Computers & Operations Research, 37*(4), 724–737.

Scheffermann, R., Bender, M., & Cardeneo, A. (2009). Robust solutions for vehicle routing problems via evolutionary multiobjective optimization. In *Proceedings of IEEE congress on evolutionary computation (CEC'09)*, Trondheim, Norway (pp. 1605–1612).

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research, 35*(2), 254–265.

Tan, K. C., Chew, Y. H., & Lee, L. H. (2006). A hybrid multiobjective evolutionary algorithm for solving vehicle routing problem with time windows. *Computational Optimization and Applications, 34*(1), 115–151.

While, L., Hingston, P., Barone, L., & Huband, S. (2006). A faster algorithm for calculating hypervolume. *IEEE Transactions on Evolutionary Computation, 10*(1), 29–38.

Zitzler, E., Laumanns, M., & Thiele, L. (2001). SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In *Proceedings of evolutionary methods for design, optimisation, and control*, Barcelona, Spain (pp. 95–100).

Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation, 3*(4), 257–271.