

Optimizing Delivery Time in Multi-Objective Vehicle Routing Problems with Time Windows

Abel Garcia-Najera and John A. Bullinaria

School of Computer Science, University of Birmingham
Edgbaston, Birmingham B15 2TT, United Kingdom
{A.G.Najera, J.A.Bullinaria}@cs.bham.ac.uk

Abstract. The Vehicle Routing Problem with Time Windows involves finding the lowest-cost set of routes to deliver goods to customers, which have service time windows, using a homogeneous fleet of vehicles with limited capacity. In this paper, we propose and analyze the performance of an improved multi-objective evolutionary algorithm, that simultaneously minimizes the number of routes, the total travel distance, and the delivery time. Empirical results indicate that the simultaneous minimization of all three objectives leads the algorithm to find similar or better results than any combination of only two objectives. These results, although not the best in all respects, are better in some aspects than all previously published approaches, and fully multi-objective comparisons show clear improvement over the popular NSGA-II algorithm.

1 Introduction

The Vehicle Routing Problem (VRP) is one of the most important and widely studied combinatorial optimization problems, with many real-world applications in distribution and transportation logistics. It has several variants that take into account different constraints. The variant *with Time Windows* (VRPTW) is particularly relevant to practical applications, and considers vehicles with limited capacity and specific delivery time windows. Its objective is to obtain the lowest-cost set of routes to deliver demand to customers. Since the problem was originally formulated as a generalization of the Traveling Salesman Problem, cost has primarily been associated with the number of routes and travel distance, but there are several other types of cost [1].

In particular, companies offering transportation services are often more interested in reducing the overall delivery time (or driver salary cost), than the overall distance traveled (or fuel cost), and there are likely to be trade-offs between them. For the standard VRP, if one assumes a constant vehicle velocity, then counting distances and times are equivalent, but that is not true for the VRPTW because of the time wasted due to arriving before delivery windows. The optimization process needs to produce a set of non-dominated solutions that represent the trade-offs between the objectives, rather than a single solution.

Exact methods can be used to find optimal solutions for small instances of the VRPTW, but the computation time required increases considerably for larger

instances [2]. We are therefore interested in using heuristics to solve this problem, in particular, an Evolutionary Algorithm (EA) that automatically generates a whole population of solutions that cover the full range of trade-offs.

There are many past studies that have solved the VRPTW as a single-objective problem using heuristics. Bräysy and Gendreau [3, 4] provide an excellent survey of them, and Bräysy et al. [5] focus on evolutionary approaches.

Other studies have considered the bi-objective optimization of the VRPTW, using an EA to minimize the number of vehicles and the travel distance. Tan et al. [7] used the dominance rank scheme to assign fitness to individuals, and designed a problem-specific crossover operator and multi-mode mutation operator. They also considered three local search heuristics. Ombuki et al. [8] used a Pareto ranking method to assign fitness and proposed further crossover and mutation operators. Finally, our own earlier study [9] incorporated a similarity measure in a Bi-objective Evolutionary Algorithm (BiEA) to select parents for the recombination process in a way that preserved a higher population diversity [10], and that enabled a better set of solutions to be obtained.

The work presented here is an improvement of the BiEA proposed in the last-mentioned study, minimizing not only the number of routes and travel distance, but also the delivery time. We analyze the results and compare them with those from previous algorithms, and introduce improved comparisons with the popular NSGA-II algorithm [6] using fully multi-objective performance metrics.

The remainder of this paper is organized as follows: The next section describes formally the VRPTW, and Section 3 reviews the two multi-objective performance metrics that are used to compare algorithms. Our proposed EA for solving the VRPTW as a multi-objective problem is described in Section 4. Then Section 5 presents the results from our algorithm, as well as comparisons with others already published. Finally, we give our conclusions in Section 6.

2 The VRP with Time Windows

Formally, the VRPTW is defined as a set $\mathcal{V} = \{0, \dots, N\}$ of vertices. Vertices in subset $\mathcal{V}^* = \mathcal{V} \setminus \{0\} = \{1, \dots, N\}$ are called *customers*. Each customer $i \in \mathcal{V}^*$ is geographically located at coordinates (x_i, y_i) , has a demand of goods $g_i > 0$, a time window $[b_i, e_i]$ during which it has to be supplied, and a service time s_i is required to unload its goods. The special vertex 0, called the *depot*, is positioned at (x_0, y_0) , has time window $[0, e_0 > \max \{e_i : i \in \mathcal{V}^*\}]$, and demand $g_0 = 0$. It is from the depot that the customers are serviced by a homogeneous fleet of vehicles with capacity $Q \geq \max \{g_i : i \in \mathcal{V}^*\}$. The problem is to design a lowest-cost set of K routes $\mathcal{R} = \{r_1, \dots, r_K\}$, such that each route begins and ends at the depot, and each customer is serviced by exactly one vehicle.

The travel between vertices i and j has various associated costs, such as the distance d_{ij} and travel time t_{ij} . For the standard benchmark problems to be considered later, one assumes unit velocity and direct travel, so the times and distances are both simply taken to be the Euclidean distances. For real-world problems, however, the distances d_{ij} are unlikely to be Euclidean and the travel

times t_{ij} are unlikely to be simply related to the distances. The following will take care to accommodate those possibilities.

Suppose $r_k = \langle u(1, k), \dots, u(N_k, k) \rangle$ specifies the sequence of N_k customers supplied in the k -th route, where $u(i, k)$ is the i -th customer to be visited in route k , and $u(0, k) = u(N_k + 1, k) = 0$ is the depot. Then $\mathcal{V}_k^* = \{u(1, k), \dots, u(N_k, k)\}$ is the set of customers in route k , and the total travel distance for that route is

$$D_k = \sum_{i=0}^{N_k} d_{u(i,k)u(i+1,k)} . \quad (1)$$

In addition to the distances, the times are also needed. Let vehicle k leave the depot at time 0, $a(u(i, k))$ denote its arrival time at the i -th customer, and $l(u(i, k))$ be the time it leaves. The arrival time at the i -th customer is then

$$a(u(i, k)) = l(u(i-1, k)) + t_{u(i-1,k)u(i,k)} . \quad (2)$$

Arriving after the end of the customer's time window is simply not allowed. If the vehicle arrives early at the i -th customer's location, it has to wait until the beginning of the customer's time window before it can start unloading, so the departure time $l(u(i, k))$ will be the maximum of the arrival time $a(u(i, k))$ and window opening time $b_{u(i,k)}$, plus the unloading time $s_{u(i,k)}$. Consequently, the waiting time $w(u(i, k))$ involved in serving the i -th customer will be

$$w(u(i, k)) = \max(0, b_{u(i,k)} - a(u(i, k))) . \quad (3)$$

Thus the departure time from the i -th customer in route k can be written as

$$l(u(i, k)) = a(u(i, k)) + w(u(i, k)) + s_{u(i,k)} , \quad (4)$$

and the total time required to complete route r_k is the arrival time at the depot

$$T_k = \sum_{i=0}^{N_k} (t_{u(i,k)u(i+1,k)} + w(u(i+1, k)) + s_{u(i+1,k)}) . \quad (5)$$

There are three VRPTW objective functions f_i that we shall concentrate on minimizing in this paper, namely the number of routes or vehicles (f_1), the total travel distance (f_2), and the total delivery time (f_3), computed as follows:

$$f_1(\mathcal{R}) = |\mathcal{R}| = K , \quad f_2(\mathcal{R}) = \sum_{k=1}^K D_k , \quad f_3(\mathcal{R}) = \sum_{k=1}^K T_k , \quad (6)$$

subject to the demands in each route r_k not exceeding the vehicle capacity, i.e.

$$G_k = \sum_{i \in \mathcal{V}_k^*} g_i \leq Q , \quad (7)$$

and no arrival times after the customer's window ends, i.e.

$$a(u(i, k)) \leq e_{u(i,k)} \quad \forall i = 1, \dots, N_k , \quad \forall k = 1, \dots, K . \quad (8)$$

The simultaneous minimization of all three objectives is not usually possible, so a set of *non-dominated* solutions needs to be obtained, each better than the others on at least one objective. In contrast to single-objective problems, where one can simply compare the best solutions from the various approaches studied, multi-objective problems have to compare whole sets of solutions. For this reason, the definition and use of multi-objective performance metrics is crucial.

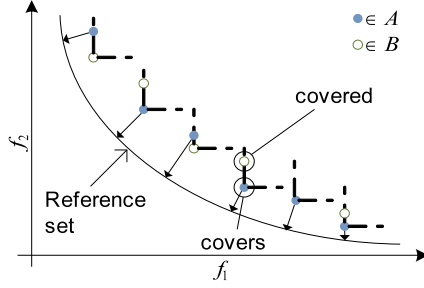
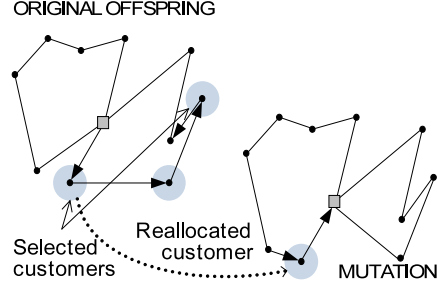
Fig. 1. Representation of M_C and M_D .

Fig. 2. Reallocation mutation operator.

3 Multi-objective performance metrics

Two existing metrics are particularly applicable to the problem at hand.

The coverage metric M_C [11] compares the number of solutions in set \mathcal{B} that are covered by (i.e. dominated by or equal to) solutions in set \mathcal{A} to the cardinality of \mathcal{B} . Thus, $M_C(\mathcal{A}, \mathcal{B})$ maps the ordered pair $(\mathcal{A}, \mathcal{B})$ to the interval $[0,1]$, as the fraction of solutions in set \mathcal{B} that are covered by solutions in set \mathcal{A} :

$$M_C(\mathcal{A}, \mathcal{B}) = \frac{1}{|\mathcal{B}|} |\{\mathbf{b} \in \mathcal{B} : \exists \mathbf{a} \in \mathcal{A}, \mathbf{a} \preceq \mathbf{b}\}|, \quad (9)$$

where the relation $\mathbf{a} \preceq \mathbf{b}$ means that \mathbf{a} is at least as good as \mathbf{b} for all the objectives, and \mathbf{a} is strictly better than \mathbf{b} for at least one objective.

The convergence metric M_D [12] measures the distance between the approximation set \mathcal{A} and a reference set \mathcal{R} . The convergence $M_D(\mathcal{A}, \mathcal{R})$ is defined as

$$M_D(\mathcal{A}, \mathcal{R}) = \frac{1}{|\mathcal{A}|} \sum_{\mathbf{i} \in \mathcal{A}} d_{\mathbf{i}}, \quad (10)$$

using the smallest normalized Euclidean distance from each point $\mathbf{i} \in \mathcal{A}$ to \mathcal{R}

$$d_{\mathbf{i}} = \min_{\mathbf{j} \in \mathcal{R}} \sqrt{\sum_{k=1}^F \left(\frac{f_k(\mathbf{i}) - f_k(\mathbf{j})}{f_k^{\max} - f_k^{\min}} \right)^2}, \quad (11)$$

where f_k^{\max} and f_k^{\min} are the maximum and minimum function values of the k -th objective function in \mathcal{R} .

The idea is that the algorithm with the best performance is the one which provides solutions with the largest coverage of the other non-dominated sets and the minimum distance to the reference set. Figure 1 presents a simple example with $M_C(\mathcal{A}, \mathcal{B}) = 3/5$ better than $M_C(\mathcal{B}, \mathcal{A}) = 2/6$, and the average distance from points in set \mathcal{A} to the reference set smaller than that for the points in \mathcal{B} . So, the algorithm producing \mathcal{A} is deemed better than that producing \mathcal{B} .

4 Multi-objective EA for VRPTW

Our proposed Multi-Objective EA (MOEA) involves selection, cross-over and mutation as in most EAs, and uses a simple list-based encoding. The main

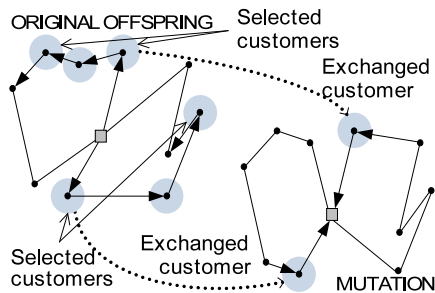


Fig. 3. Exchange mutation operator.

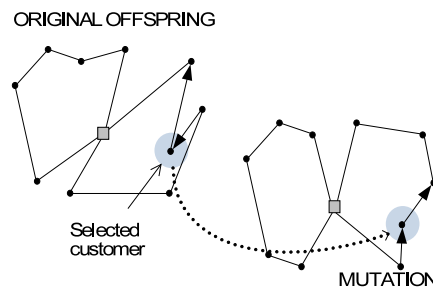


Fig. 4. Reposition mutation operator.

novel characteristic is the implementation of a similarity measure to preserve population diversity. This is based on the *Jaccard's similarity coefficient*, which measures how similar two sets are as the ratio between the number of common elements and the total number of elements. We adapted this metric to the VRPTW for our earlier BiEA [9] by treating each solution as a set of arcs, and used it to calculate the average similarity between each solution in the population and every other solution. The MOEA here differs from the BiEA in its improved mutation stage, and in dealing with more than two objectives.

A dominance depth criteria [6] is used to assign fitness to solutions, by which individuals are grouped into non-dominated fronts and the relative depth of the front defines the fitness. Then in the recombination stage, a tournament selection is used to choose the first parent according to fitness as usual, but the second parent is selected on the basis of lowest similarity measure. Afterwards, the recombination process is designed to preserve routes from both parents.

The mutation involves the use of three basic functions: (i) `selectRoute()` stochastically selects a route according to the largest ratio between the travel distance and the number of customers, (ii) `selectCustomer()` stochastically selects one customer from a specific route according to the longest average length of its inbound and outbound arcs, and (iii) `insertCustomers()` tries to insert a set of customers, where the lowest travel distance is obtained, into any specific route, or all existing routes. The last two functions are used by the operators:

- *Reallocation* - Takes a random sequence of customers from a given route and reallocates them to other existing routes, as illustrated in Figure 2.
- *Exchange* - Swaps a sequence of customers between two routes if that is possible, as illustrated in Figure 3.
- *Reposition* - Selects one customer from a specific route and reinserts it into the same route, as illustrated in Figure 4.

Two routes are first chosen using `selectRoute()`. If they are the same, the *Reallocation* operation is performed, otherwise the *Exchange* operator is. Then `selectRoute()` selects another route and the *Reposition* operator is carried out. Finally, the parent and offspring populations are combined and fitness levels assigned, and the individuals with highest fitness form the next generation.

5 Experimental study

Our study has three purposes: First to determine whether the minimization of delivery time has any effect on the performance of the algorithm on minimizing the number of routes and travel distance. Second, to test how well our improved MOEA performance compares with previous approaches. Finally, to perform a direct fully multi-objective comparison of MOEA with NSGA-II [6].

To carry out controlled experiments, we used the standard benchmark set of Solomon [13] that includes 56 instances of size $N = 100$ categorized as clustered (C1, C2), random (R1, R2), and mixed (RC1, RC2). Tan et al. [7] found that categories C1 and C2 have positively correlating objectives, but the majority of the instances in categories R1, R2, RC1 and RC2 have conflicting objectives.

We ran our algorithm and NSGA-II 30 times for each problem instance. The parameters of our algorithm were set to values that have proved to work well in preliminary testing: population size = 100, number of generations = 500, tournament size = 2, crossover rate = 1.0, and mutation rate = 0.1.

5.1 Minimization of the delivery time

The first aim was to analyze the performance of our MOEA with different objective settings, to test the effect of the additional delivery time objective. The number of routes (R), travel distance (D) and delivery time (T) were first set to be minimized in pairs, giving three objective settings (RD, RT and DT), and then all three of them were minimized together (RDT).

To use the coverage metric, for each setting and instance, the outcome set of non-dominated solutions after each of the 30 repetitions was recorded. Then, for each given instance and ordered pair of settings X and Y, $M_C(X_i, Y_j), \forall i, j = 1, \dots, 30$ were computed for the three-objective space, that is 900 M_C values. The average of these M_C values over the instances in each category are presented in Table 1, and in brackets are the numbers of instances for which the result is significantly better than the reverse case (determined by a *t-test* at 0.05 significance level). For categories C1 and C2, despite all four settings having a high coverage of each other, DT and RDT have a higher coverage of RD and RT. Otherwise, the coverage of RT, DT and RDT by RD is low ($\leq 14\%$), and the coverage of RD, DT and RDT by RT is nearly zero. The most interesting cases are settings DT and RDT, as their coverage of RD and RT is much higher. Between them, the coverage of DT by RDT is significantly larger than the inverse case in more instances than the inverse case, despite similar average coverages.

To apply the convergence metric, for each instance and objective setting, the overall non-dominated solutions were extracted from the 30 non-dominated sets, and composite non-dominated reference sets \mathcal{R} in the three-objective space were generated. Then, for each setting X, $M_D(X_i, \mathcal{R})$ was computed for all $i = 1, \dots, 30$. The averages of these M_D values over the instances in each category are presented in Figure 5. It can be seen that, in general, solutions from settings RD and RT are the farthest from the reference set, while those from DT and RDT are the nearest.

Table 1. Coverage metric values, averaged over instance categories, for solutions obtained with MOEA settings RD, RT, DT and RDT. In brackets are the numbers of instances for which the result is significantly better than the reverse case.

Obj.	Covers	C1	C2	R1	R2	RC1	RC2
RD	RT	0.87 (6)	0.64 (3)	0.04 (6)	0.01 (1)	0.05 (4)	0.02 (2)
	DT	0.82 (1)	0.72 (0)	0.08 (1)	0.11 (4)	0.14 (0)	0.08 (0)
	RDT	0.82 (1)	0.72 (1)	0.08 (1)	0.11 (3)	0.13 (0)	0.08 (1)
RT	RD	0.68 (0)	0.55 (1)	0.01 (2)	0.00 (0)	0.01 (2)	0.01 (0)
	DT	0.68 (0)	0.63 (0)	0.03 (0)	0.04 (0)	0.06 (0)	0.02 (0)
	RDT	0.68 (0)	0.62 (0)	0.04 (0)	0.03 (0)	0.07 (0)	0.03 (0)
DT	RD	0.91 (2)	0.90 (3)	0.31 (11)	0.14 (5)	0.36 (8)	0.21 (6)
	RT	0.97 (5)	0.92 (4)	0.49 (12)	0.42 (11)	0.46 (8)	0.48 (8)
	RDT	0.91 (2)	0.88 (2)	0.43 (4)	0.40 (4)	0.42 (2)	0.42 (3)
RDT	RD	0.89 (3)	0.91 (3)	0.32 (11)	0.16 (6)	0.38 (8)	0.23 (7)
	RT	0.97 (6)	0.93 (3)	0.52 (12)	0.44 (11)	0.49 (8)	0.44 (8)
	DT	0.89 (2)	0.89 (1)	0.44 (5)	0.43 (5)	0.46 (6)	0.41 (3)

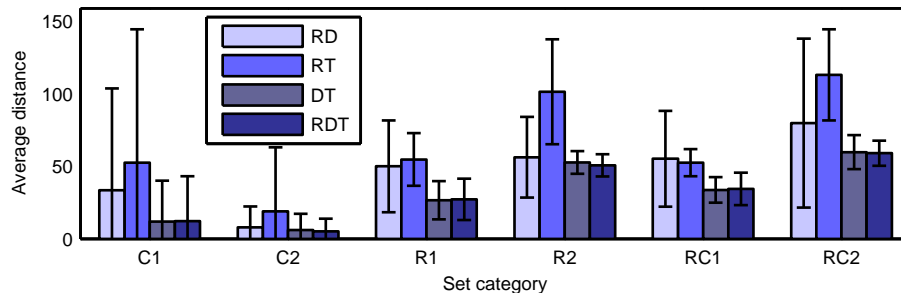


Fig. 5. Convergence metric values, averaged over instance categories, for solutions obtained with MOEA settings RD, RT, DT and RDT.

Even though DT and RDT produce similar results for the convergence metric, we can conclude that, considering both metrics, setting the MOEA to minimize all three objectives does lead to better non-dominated solutions. Consequently, the solutions from the RDT case will be used for all the following analyses.

5.2 Comparison with previous studies

Although previous studies have considered the VRPTW as a multi-objective problem, they have not presented their results in a multi-objective manner, and have only shown averages of their best results. So the averages of our best results are now compared with those previous studies, before performing proper multi-objective comparisons with NSGA-II. Table 2 presents the average number of routes and travel distances from MOEA, and those from three previous multi-objective studies. For each instance, all the solutions in the resulting non-dominated set are taken from all repetitions, and the average for each objective

Table 2. Numbers of routes (upper figures) and travel distances (lower figures), averaged over categories, for the best solutions found in past studies and by our MOEA.

Algorithm	C1	C2	R1	R2	RC1	RC2	Accum.
Tan et al. [7]	10.00	3.00	12.92	3.55	12.38	4.25	441.00
	828.91	590.81	1187.35	951.74	1355.37	1068.26	56293.06
Ombuki et al. [8]	10.00	3.00	13.17	4.55	13.00	5.63	471.00
	828.48	590.60	1204.48	893.03	1384.95	1025.31	55740.33
BiEA [9]	10.00	3.00	12.50	3.18	12.38	4.00	430.00
	830.64	589.86	1191.22	926.97	1349.81	1080.11	56125.35
MOEA	10.00	3.00	12.83	3.82	12.63	4.38	446.00
	828.38	589.86	1191.30	916.32	1349.24	1060.80	55829.68
% diff. R	0.00	0.00	2.67	20.00	2.02	9.38	3.72
% diff. D	0.00	0.00	0.33	2.61	0.00	3.46	0.16

is computed. Then, these are averaged over each category. For each algorithm and category is shown the average number of routes (upper figure) and the average travel distance (lower figure). The last column (Accum.) presents the total average number of routes and travel distance for all 56 instances. The last two rows show the percentage difference between the results from MOEA and those from the method that obtained the lowest value for each objective.

For categories C1 and C2, that do not have conflicting objectives, our MOEA achieved similar results to the previously published studies. The lowest number of routes for the other categories, as well as the accumulated, was obtained by our BiEA [9], but MOEA found solutions with lower travel distances for categories R2, RC1 and RC2, and accumulated. Solutions from Tan et al. [7] have the lowest travel distance for category R1, where results from MOEA are 0.33% higher, and Ombuki et al. [8] obtained the shortest distances for categories R2 and RC2, and accumulated, where results from MOEA are 2.61%, 3.46%, and 0.16% higher, but have smaller numbers of routes. Finally, travel distances from MOEA for category RC1 are the shortest. These results show that, overall, MOEA’s performance is comparable to the previously published algorithms.

5.3 Comparison with NSGA-II

Simple averages as in Table 2 are often misleading, so our results are now analyzed using the multi-objective coverage and convergence performance metrics to compare the non-dominated solutions from MOEA with those from NSGA-II [6]. For a fair comparison, the NSGA-II implementation used the same solution representation, with the same crossover and mutation operators, as MOEA. The difference lies in the fact that MOEA makes use of solution similarity, while NSGA-II utilizes the *crowding distance* which does not involve any routing information at all. The coverages $M_C(\text{MOEA}_i, \text{NSGA-II}_j)$ and $M_C(\text{NSGA-II}_j, \text{MOEA}_i)$ were determined as described above, and then the convergences $M_D(\text{MOEA}_i, \mathcal{R})$ and $M_D(\text{NSGA-II}_i, \mathcal{R})$ were computed using a combined reference set \mathcal{R} of solutions

Table 3. Coverage (upper figure) and convergence (lower figure), averaged over instance categories, for solutions obtained with NSGA-II and MOEA. In brackets are the numbers of instances which are significantly better than the other approach.

Algorithm	C1	C2	R1	R2	RC1	RC2
NSGA-II	0.81 (0)	0.87 (2)	0.14 (0)	0.37 (4)	0.13 (0)	0.35 (0)
	25.06 (0)	4.88 (0)	51.56 (0)	49.81 (0)	72.21 (0)	67.17 (0)
MOEA	0.93 (4)	0.88 (2)	0.78 (12)	0.41 (5)	0.80 (8)	0.45 (7)
	12.20 (3)	5.07 (0)	27.89 (12)	50.67 (0)	33.91 (8)	65.86 (1)

obtained from both algorithms. Table 3 shows the average M_C (upper figure) and M_D (lower figure) over the instances in each category, and the number of instances significantly better than the other algorithm (in brackets).

Both algorithms show similar coverage for categories C1 and C2, but MOEA has significantly higher coverage of NSGA-II in more instances. For C1, solutions from MOEA are closer to \mathcal{R} on average, and for three instances are significantly better than NSGA-II. For C2, solutions from NSGA-II are closer to \mathcal{R} on average, but no instances have significant differences. For all instances in categories R1 and RC1, solutions in the non-dominated sets found by MOEA have a significantly higher coverage of those obtained by NSGA-II, and are also significantly nearer to \mathcal{R} . For category R2, MOEA has a significantly higher coverage in five instances, and NSGA-II is significantly higher in four, while both show similar distances to \mathcal{R} . Finally, for category RC2, MOEA has a significantly higher coverage in seven instances, despite both algorithms having similar average distance to \mathcal{R} . Overall then, it can be concluded that our new MOEA, with its similarity-based second parent selection, performs significantly better than NSGA-II, and this is particularly clear for categories R1 and RC1.

6 Conclusions

We have proposed and analyzed the performance of our new Multi-Objective EA (MOEA) for solving the multi-objective VRPTW, which is an improvement of our earlier Bi-objective EA (BiEA) [9] that introduced similarity-based selection to enhance solution diversity. This involved improved mutation operators, improved analysis using fully multi-objective performance metrics, and performance testing for a third objective, namely the delivery time (T), in addition to the previously studied number of routes (R) and travel distance (D).

We tested four different objective settings: first minimizing pairs of objectives (RD, RT and DT), and then all three at once (RDT). The coverage and convergence performance metrics were used to evaluate the algorithm, showing that settings DT and RDT have a higher coverage of RD and RT, and their solutions are closer to a composite reference set, indicating that the minimization of the delivery time improves the algorithm’s performance. Moreover, RDT covers DT in more instances, which implies that even better results can be obtained by considering the minimization of all three objectives at the same time.

Averages of the non-dominated sets found by our MOEA with setting RDT were compared with previous studies, and, although not better in all respects, MOEA is better in some, and competitive on average. Significantly, using fully multi-objective coverage and convergence metrics to compare the MOEA against the well-known evolutionary multi-objective optimizer NSGA-II showed that the solutions found by MOEA are better for almost all instance categories.

Given the promising performance of our algorithm, we are now looking at the possibility of optimizing even more objectives (such as route balance [1]), and pursuing the comparison of our results with other multi-criterion optimization methods and further multi-objective performance metrics.

7 Acknowledgments

The first author acknowledges support from CONACYT (Scholarship 229168).

References

1. Jozefowicz, N., Semet, F., Talbi, E.G.: Multi-objective vehicle routing problems. *Eur. J. Oper. Res.* **189**(2) 293–309 (2008)
2. Desrochers, M., Desrosiers, J., Solomon, M.: A new optimization algorithm for the vehicle routing problem with time windows. *Oper. Res.* **40**(2) 342–354 (1992)
3. Bräysy, O., Gendreau, M.: Vehicle routing problem with time windows, part I: Route construction and local search algorithms. *Transport Sci.* **39**(1) 104–118 (2005)
4. Bräysy, O., Gendreau, M.: Vehicle routing problem with time windows, part II: Metaheuristics. *Transport. Sci.* **39**(1) 119–139 (2005)
5. Bräysy, O., Dullaert, W., Gendreau, M.: Evolutionary algorithms for the vehicle routing problem with time windows. *J. Heuristics* **10**(6) 587–611 (2004)
6. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE T. on Evolut. Comput.* **6**(2) 182–197 (2002)
7. Tan, K.C., Chew, Y.H., Lee, L.H.: A hybrid multiobjective evolutionary algorithm for solving vehicle routing problem with time windows. *Comput. Optim. and Appl.* **34**(1) 115–151 (2006)
8. Ombuki, B., Ross, B.J., Hanshar, F.: Multi-objective genetic algorithms for vehicle routing problem with time windows. *Appd. Intel.* **24**(1) 17–30 (2006)
9. Garcia-Najera, A., Bullinaria, J.A.: Bi-objective optimization for the vehicle routing problem with time windows: Using route similarity to enhance performance. In Ehr Gott, M., Fonseca, C., Gandibleux, X., Hao, J.K., Sevaux, M. (eds.) *EMO 2009. LNCS*, vol. 5467, pp. 275–289. Springer (2009)
10. Garcia-Najera, A.: Preserving population diversity for the multi-objective vehicle routing problem with time windows. In: *GECCO (Companion) 2009*. pp. 2689–2692. ACM (2009)
11. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Comput.* **8**(2) 173–195 (2000)
12. Deb, K., Jain, S.: Running performance metrics for evolutionary multi-objective optimization. *KanGAL report 2002004*, Indian Institute of Technology (2002)
13. Solomon, M.M.: Algorithms for the vehicle routing and scheduling problems with time windows constraints. *Oper. Res.* **35**(2) 254–265 (1987)