A Memetic Algorithm for the Capacitated Vehicle Routing Problem with Time Windows

Oscar M. González*, Carlos Segura*, S. Ivvan Valdez Peña*, and Coromoto León†

*Centro de Investigación en Matemáticas, A.C. (CIMAT), Área de Computación,
Jalisco S/N, Col. Valenciana, Guanajuato, 36023, México
‡Departamento de Ingeniería Informática y de Sistemas
Universidad de La Laguna
San Cristóbal de La Laguna, Spain
Email: oscar.gonzalez@cimat.mx, carlos.segura@cimat.mx, ivvan@cimat.mx, cleon@ull.edu.es

***Abstract*—Vehicle Routing Problem (VRP) is a widely known NP-Hard combinatorial optimization problem. This paper presents a proposal of a memetic algorithm (MA) with simulated annealing (SA) as trajectory-based method for solving the Capacitated Vehicle Routing Problem with Time Windows (CVRPTW). A novel crossover operator, the Single Breaking-point Sequence Based Crossover (SBSBX), is introduced and compared with a widely used operator, the Sequence-based Crossover (SBX). One of the principles behind the design of SBSBX is to reduce the disruptive behavior of SBX, with the aim of providing additional intensification. Initial studies show that the different crossover operators heavily impact the preservation of diversity in the population. Thus, two different parent-selection operators that induce different selection pressure are applied: random selection and binary tournament. The proposal is validated using the well-known Solomon's benchmark. The experimental validation shows that in some of the tested methods premature convergence is an important issue, whereas in other cases convergence is not attained. Overall, the combination of SBSBX and random selection attains the most promising results. In fact, a new best-known solution could be generated for one commonly used instance.**

I. Introduction

Computing the optimal route in a network is a challenge in industry, economic and scientific domains. The effective computing of optimal routes impacts to transportation, distribution, and logistic planning of products [1]. The Vehicle Routing Problem (VRP) [2] is a general term that is used to denote a set of problems that deal with the finding of routes in a graph. The common aim is to minimize the cost of those set of routes while satisfying a set of constraints. The VRP graph has a special node that is called the depot of the store, whereas the other nodes are the customers. A set of routes defines the vehicle trips. All routes start at the depot, then, they visit a set of customers and return to the depot. Each customer is visited once by any route. In the Capacitated Vehicle Routing Problem (CVRP) each vehicle has a maximum capacity and the customers have a demand which must be satisfied. The Capacitated Vehicle Routing Problem with Time Windows (CVRPTW) [3] is the CVRP, in which, additionally, each customer has a time window to be served. This time window is the time when the customer is available and the vehicle must serve the client within the start and end of the time window.

Exact methods [4] require of an exponential quantity of computational resources or time, thus they might not be admissible methods when the problem size is greater than some dozens of clients. Heuristic and metaheuristic methods [5] have proven to be adequate alternatives since they can find close approximations to the global optimum, although, most of the times, the quality of the solution is related to the stopping criterion. Evolutionary Algorithms (EAs) [6] and Simulated Annealing (SA) [7] are two of the metaheuristics that have attained some of the most promising results. However, there is yet a relatively large difference between currently computed best-known solutions and the best-known lower bounds [8], meaning that there is possibly some room for improvement. In fact, in some instances with special features, some mathematical methods have found the exact solutions, but metaheuristics have not been able to reproduce such results.

The goal of this paper is to improve further the state of the art in the application of metaheuristics to the VRP. Particularly, we started from the hypothesis that one of the drawbacks of current methods is that not too much attention have been paid to the way in which the diversity is managed. Thus, as a first step towards this goal, this paper focuses on analyzing the implications of the diversity on the results for the CVRPTW. Additionally, new algorithms that induce different degrees of diversity than the most commonly used approaches are devised. The studies focus on the relations among diversity, crossover operators, and the parent selection phase. In order to perform our analyses, two different crossover operators are used. In addition, two different ways of selecting the parents that induce different selection pressures are applied. Taking into account the most promising approaches, our studies have been performed by developing a new memetic

978-1-5090-4601-0/17/$31.00 ©2017 IEEE

algorithm (MA) that is equipped with SA as trajectory-based method. This MA applies a novel crossover operator, called the Single breaking-point Sequence-Based Crossover (SBSBX). SBSBX is an extension of the Sequence-Based Crossover (SBX) [9]. Our analyses show that SBSBX is less disruptive than SBX, meaning that the properties related to the convergence of the population highly depends on the operator applied.

Our experimental validation with Solomon's benchmark shows that the behavior in terms of diversity and quality highly depends on the crossover and parent selection. The combination of SBSBX with random selection has resulted in the most promising scheme. In fact, this novel method has been able to generate a new best-known solution in a well-known instance, showing that in fact, the different levels of degree induced by the different approaches heavily impact on the produced results.

The paper is organized as follows: Section II includes the mathematical definition of the problem with all its constraints. Section III introduces the main benchmark used in the CVRPTW and summarizes the main algorithms that have been used to tackle the CVRPTW. Additionally, several crossover operators for the CVRPTW and some distance metrics available to evaluate the diversity of the population are described. Section IV presents the memetic algorithm and details the SBX and SBSBX operators. Additionally, the neighborhood used in SA and the transformation operations are discussed. Our experimental validation, including diversity and quality, is detailed and discussed in Section V. Finally, conclusions and some lines of future work are given in Section VI.

II. Vehicle Routing Problem: Mathematical Definition

There are many variants of the VRP where the difference among them are the constraints and/or objectives in the optimization model. In this article, we address the Capacitated Vehicle Routing Problem with Time Windows (CVRPTW) introduced previously.

The mathematical definition of the objectives and constraints of the CVRPTW can be modeled as follows: let $G = (V, E)$ be an undirected full graph, where vertices V = $\{1, \ldots, N\}$ correspond to a depot and customers, and the edges $e \in E\{(i, j) : i, j \in V\}$ are paths among them.

A candidate solution is represented by a set of K routes that start and end at the depot. Note that, some of these routes might be empty, so K represents the maximum number of routes. Each route k is a simple cycle, so they can be established by indicating whether a vehicle travels from node i to node j. Thus, these routes can be mathematically defined with a set of decision variables X_{ij}^k, to set routes as follows:

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

The CVRPTW can be stated as follows:

$$\min TD = \sum_{k=1}^{K} \sum_{i=1}^{N} \sum_{j=1}^{N} X_{ij}^k C_{ij} \tag{1}$$

Subject to

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

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

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

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

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

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

$$\begin{aligned} s_{ki} + C_{ij} - L(1 - X_{ij}^k) \le s_{kj}(\forall i, j \in \{1, \ldots, N\}, \\ \forall k \in \{1, \ldots, K\}) \end{aligned} \tag{8}$$

$$a_j \le s_{kj} \le b_j(\forall i, j \in \{1, \ldots, N\}, \forall k \in \{1, ..., K\}) \tag{9}$$

The objective function is defined in equation (1) where C_{ij} is the cost for traveling from node i to node j (here C_{ij} is considered as the distance or time required for traveling from node i to node j), K is the maximum number of vehicles that can be used, N is the number of customers plus the depot (the depot is tagged with number 1, and the customers are denoted as $2, \ldots, N$). Equation (2) enforces that a vehicle cannot travel from a customer to the same customer. Equation (3) restricts the value of x_{ij}^k to 0 or 1; when it is set to one it means that vehicle k travel from node i to node j, otherwise, it is set to 0. Equation (4) ensures that a customer is visited only one time. Equation (5) guarantees that sum of customer capacities visited by a vehicle does not exceed the capacity of the vehicle ($Q^k = Q$ for all vehicles), where Q^k is the loading capacity of vehicle k (in our case, the same capacity Q is considered for all vehicles) and d_j is the demand at customer j. Equation (6) requires that a maximum of K routes start at a depot. Equation (7) guarantees all routes start and finish in the depot. Equation (8) means that if vehicle k is traveling from customer i to customer j it cannot arrive at customer j previously to $s_{ki}+C_{ij}$, where s_{ki} is the visiting time of customer i by vehicle k and L is a large scalar ($L = \sum_{i=1}^{N} \sum_{j=1}^{N} C_{ij}$). Finally, Equation (9) ensures that time windows are fulfilled, where a_j is the earliest time for customer j to allow the service and b_j is the latest time for customer j to allow the service.

III. Literature Review

In this section, we provide a brief review of some of the most relevant approaches that have been used to tackle the CVRPTW [5]. Considering that the crossover operator is an essential component of EAs to successfully address the CVRPTW, a brief analysis of the most well-known operators is performed. Furthermore, since different crossover operators induce different exploratory powers, and some of them are more disruptive than others, it is important to be able to analyze these properties. A way to perform this analysis is through the definition of distance metrics to measure the differences among candidate solutions. Thus, we briefly review distance metrics to analyze diversity in EAs. Additionally, some of the most well-known benchmarks of CVRPTW are introduced.

A. Benchmarks

The most commonly used benchmark for the CVRPTW is probably the Solomon's benchmark [10]. Solomon designed 3 groups of instances: Cxxx instances, Rxxx instances, and RCxxx instances. Cxxx instances have the customers clustered in geographic regions; in instances Rxxx, the positions of customers are randomized; finally, instances RCxxx are a combination between Cxxx and Rxxx. In the original definition of the benchmark cases with 25, 50, and 100 customers were designed. In order to perform large-scale analyses, new instances with 200, 400, 600, 800 and 1000 customers were proposed in [11].

B. Algorithms

Several trajectory-based schemes have been proposed in the specialized literature. Among them, simulated annealing (SA) has provided quite promising results. In [7], authors proposed a SA algorithm with multiple temperatures. Additionally, they introduced a parallel version of their algorithm using OpenMP and MPI for solving Solomon's benchmark. The results reported in [7] are competitive with Cxxx instances. However, they do not attain so high-quality results when dealing with instances such as Rxxx and RCxxx.

In the case of population-based metaheuristics, memetic algorithms are one of the most popular approaches. In [12], a multi-objective memetic algorithm to solve the CVRPTW was proposed. This proposal simultaneously minimizes the traveling distance and the number of routes. A simple hill-climbing approach is used as the local search engine. They also test their proposal with Solomon's benchmark and it is considered to be one of the most promising approaches. In fact, it reported some of the best-known solutions for Cxxx instances. However, results for other instances are not so promising. Since one of the effects of multi-objective approaches is to alter the way in which diversity is managed [13], this so promising results might be indicating that single objective schemes might be having issues with the proper control of diversity. In fact in many problems, the importance of the proper management of diversity has already been shown [14]. In this last proposal, the use of diversity as an additional objective provided important benefits for the frequency assignment problem.

A hybrid EA is proposed in [15]. The authors combine an adaptive large neighborhood search (ALNS) [16] and a population-based search to construct their hybrid EA. Since this algorithm is not applied with the classical Solomon's benchmark it is difficult to compare against another proposals.

Finally, in [17] a framework that uses a kind of dynamic programming heuristic to solve the Traveling Salesman Problem (TSP) is introduced. Authors use this approach to approximate solutions to any kind of VRP. The proposal is tested on Solomon's benchmark and Goel's instances [18] that are a modification of Solomon's benchmark. They only report the mean fitness obtained when taking into account the whole benchmark set, so properly comparing against this approach is also difficult.

C. Crossover Operators

In this section, we describe some crossover operators used to solve the CVRPTW. In [9], authors proposed two crossover operators: the Sequence-Based Crossover (SBX) and the Route-Based Crossover (RBX). The SBX generates an offspring by merging two routes from two parents. The RBX generates offspring by replacing a route from the second parent by another that is taken from the first one. Since both crossover operators might generate invalid offspring, a repairing phase is applied. Finally,the Optimized Crossover Operator, proposed in [19] to solve the CVRP, uses a complete undirected bipartite graph to find two new children. According to the specialized literature, SBX has obtained the most promising results. As a result, the SBX is selected to perform our studies.

D. Distance Metrics

Distance metrics measure how far or close is an individual from another individual. A small distance between two individuals implies that they are searching in a similar region of the search space. With the aim of avoiding premature convergence, EAs should search in different regions. However, after detecting promising regions, it is important to be able to intensify the search in such regions.

Considering the VRP there are several suitable metrics which intend to capture the nature of this problem, for instance, in [20] authors discuss about exact match distances. This metric is based on counting the number of values that match in the same position in both individuals. A high number means that individuals are similar. In [21] authors develop the R-type distance. A similarity among two individuals can be defined as the number of edges that are common in both individuals. Then, the R-type distance between individual I_1 and individual I_2 is the amount of edges in I_1 minus the common edges.

Algorithm 1: General schema of memetic algorithm

```
   Data: population size, stopping criterion, initial
         temperature, crossover operator, parent
         selection operator
   Result: the best individual found
1  initialization of population;
2  while stopping criterion is not reached do
3      while |offspring|<population size do
4          select two parents with random selection or
           binary tournament;
5          apply SBX or SBSBX operator;
6          insert new individual in offspring;
7      end
8      apply simulated annealing to each offspring;
9      replace parents population with offspring
       population and insert elite individual (best
       individual of previous population);
10 end
```

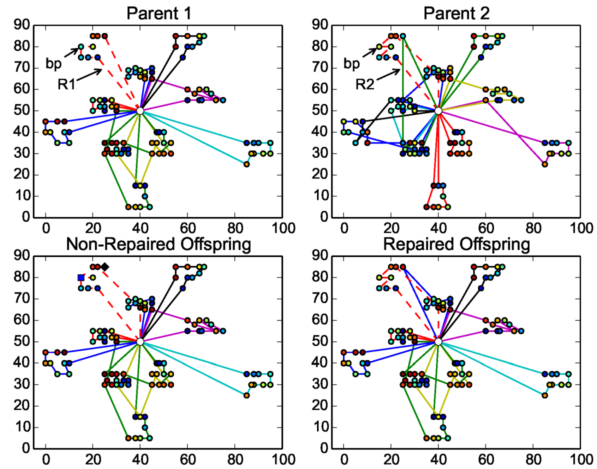

Fig. 1. Sequence-Based Crossover operator with repairing phase (bp=breakpoint)

IV. Memetic Algorithm

In this paper, a novel memetic algorithm to address the CVRPTW is proposed. It is a population-based algorithm that uses a crossover operators and a trajectory-based method to enhance offspring. Taking into account the performance of different state-of-art schemes, we decided to apply the Sequence-Based Crossover operator (SBX). In addition, we propose a novel variant of SBX that enhances its performance. SBX is deeply explained in section IV-A. The novel operator is called the Single Breaking-point Sequence-Based Operator (SBSBX).

The proposed memetic algorithm applies simulated annealing (SA) as trajectory-based method. Specifically, SA is applied on each offspring resulting from the crossover by setting its stopping criterion to 5 seconds. SA takes into account the definition of neighborhood proposed in [7]. Additionally, the population initialization proposed in [7] is used by our proposal. Specifically, the initial routes are built by using three fast strategies selected randomly. In the first one, the customers are assigned to routes randomly until all customer are assigned. In each step, the customer is chosen randomly. In the second one, the customers are inserted to random routes according to their identifier, i.e. customer 1 is assigned first, then customer 2 is assigned, and so on. Finally, in the third proposal, the customers are assigned to random vehicles in ascending order of time windows. Specifically, the end time is used in the sorting approach. In order to evaluate the individual, both the cost of the routes and the constraints are taken into account. In the cases where the constraints are not satisfied, the fitness function associated to the individual is the cost of the routes plus the constant L defined in Section II. Otherwise, the fitness function is just the cost of the routes. In according to [22], the cost associated to each edge is truncated to the first digit. Also, two different parent selections are used: random parent selection and binary tournament selection. The random parent selection has a low-pressure selection because any individual has the same probability to be chosen. However, for the binary tournament, the fitter individuals have larger probabilities to be chosen. A general overview of our proposal is showed in algorithm 1.

In algorithm 1, step 1 initializes the population as previously described. Then, until a stopping criterion is reached (step 2), which in our case is set as the elapsed time, the steps 3 to 9 are executed. While the offspring population is not filled with N individuals (step 3), the steps 4, 5 and 6 are executed to get a new offspring. Step 4 selects two parents with random parent selection or binary tournament, whereas Step 5 applies a crossover operator to generate a new individual. Step 6 inserts the individual generated into the offspring population. Step 8 applies the SA to the offspring population. Finally, step 9 replaces the previous population with the offspring population. Additionally, the best individual of the previous population is also maintained.

A. Crossover Operator

Sequence-Based Crossover (SBX) [9] is used as crossover operator by our proposal. The procedure of SBX operator is illustrated in Fig. 1. Given two parents, P_1 and P_2, a new offspring is generated. From each parent, one route — R_1 and R_2 — is randomly selected. Then, a random customer is chosen from each route as breakpoint of the route. As result of this, two sub-routes $R_1 = (R_{11}, R_{12})$ and $R_2 = (R_{21}, R_{22})$ are created. A new route R_{new} is generated by merging the first part of route R_1 and the second part of route R_2 in $R_{new} = (R_{11}, R_{22})$. The new offspring is generated by replacing R_1 by R_{new} in P_1.

Algorithm 2: Single Breaking-point Sequence-Based Crossover Operator

Data: $Parent_1$ selected, $Parent_2$ selected
Result: offspring generated
1 copy $parent_1$ to offspring;
2 select $Route_1$ and $customer_1$ in $parent_1$;
3 search for $customer_1$ in $parent_2$ and select his route as $Route_2$;
4 merge $Route_1$ and $Route_2$ in $Route_{new}$ with $customer_1$ as breakpoint ;
5 apply repairing phase to offspring generated ;

Solutions with unrouted (see the black diamond in Fig. 1) or duplicated customers (blue square in Fig. 1) might appear. Thus, we apply a repairing procedure in order to always generate a valid offspring, i.e. an offspring where each customer is visited one time. Three different steps are used to repair solutions:

- **Duplicated customer in R_{new}:** A duplicated customer is chosen randomly and removed.
- **Duplicated customer in different routes:** Since we want to keep R_{new}, the duplicated customer that is not in R_{new} is removed.
- **Unrouted Customer:** This customer is inserted at the feasible insertion place that minimizes the cost. If there is not a feasible insertion place, it is assigned randomly.

If a non-feasible offspring is generated, i.e. one that does not fulfill all the constraints, in [9] the process is repeated until a feasible offspring is generated. This step is avoided in our case, because we could verify that our SA usually converts any non-feasible solution into a feasible one in very few time.

By applying the R-type metric, we could verify that the classical SBX is quite disruptive, specially when the breaking points selected in the candidate solutions are different. In fact, even if the two parents are equal, a new different offspring might be generated. Thus, intensification is in some way avoided by the application of the SBX operator because it usually highly modifies the region where the search is performed. This is quite an usual behavior for a crossover operator. As a result, we propose a new variant, the SBSBX which selects the same breaking points in both parents (see Algorithm 2). In this way, a less disruptive behavior is induced. In fact in this case, when two equal parents are considered, the offspring is also a copy of the parents. Thus, with the application of this operator the search might be more focused in specific regions. In addition, SBSBX is less elitist than the original SBX with respect to unrouted customers. Specifically, SBSBX places unrouted customers randomly instead of choosing the best position. The route selected for placing unrouted customers can be any except R_{new}.

The SBSBX operator is described in Algorithm 2. Step 1 copies the $parent_1$ to offspring candidate. In step 2 a customer is selected and, in this case, its route is also selected. The customer selected from $parent_1$ is searched in $parent_2$ and its route is selected in step 3. $Route_{new}$ is generated by merging both routes selected and the customer is used as the single breaking-point in step 4. Finally, in step 5 the repairing phase is applied to the offspring generated.

TABLE I
TRANSFORMATION OPERATION PROBABILITY

Transformation Operation	Probability
Customer Random Reallocation	15%
Customer Best Reallocation	5%
Customer Random Migration	15%
Customer Best Migration	5%
Customers Random Exchange	10%
Customer Best Exchange	5%
Customer Exchange With coincident Time Window	10%
Route Partition	15%
New Route	15%
Route Elimination	5%

B. Trajectory-Based Method

The trajectory-based method applied in our proposal is a Simulated Annealing [23] based on 10 different transformations to generate neighbors. Specifically, the transformations are the following: Customer Random Reallocation, Customer Best Reallocation, Customer Random Migration, Customer Best Migration, Customers Random Exchange, Customer Best Exchange, Customer Exchange With coincident Time Window, Route Partition, New Route, and Route Elimination. Readers interested in the details of each transformation are referred to [7]. In each iteration of SA a transformation is chosen with the probabilities given in Table I.

V. EXPERIMENTAL VALIDATION

This section is devoted to present the experimental validation of our proposal. Since our proposal is a stochastic approach, every experiment takes into account 30 independent runs. The algorithm was coded in c++ and compiled with g++ 4.8.2 for Ubuntu 14.04.1. All tests were run on an Intel(R) Xeon(R) CPU E5-2620 v2 (2.10 GHz - 32 GB of RAM). Additionally, in order to properly compare the performance of the algorithms, a set of statistical tests are performed, following the guidelines presented in [24]. First, the results are checked to see if they followed a Gaussian distribution with the Shapiro-Wilk test. If they follow Gaussian distribution, the Levene test is applied to check the homogeneity of the variances. In case of homogeneity, an ANOVA test is done; otherwise, a Welch test was performed. When Shapiro-Wilk shows that they are not following a Gaussian distribution, the non-parametric Kruskal-Wallis test is used to test whether samples are

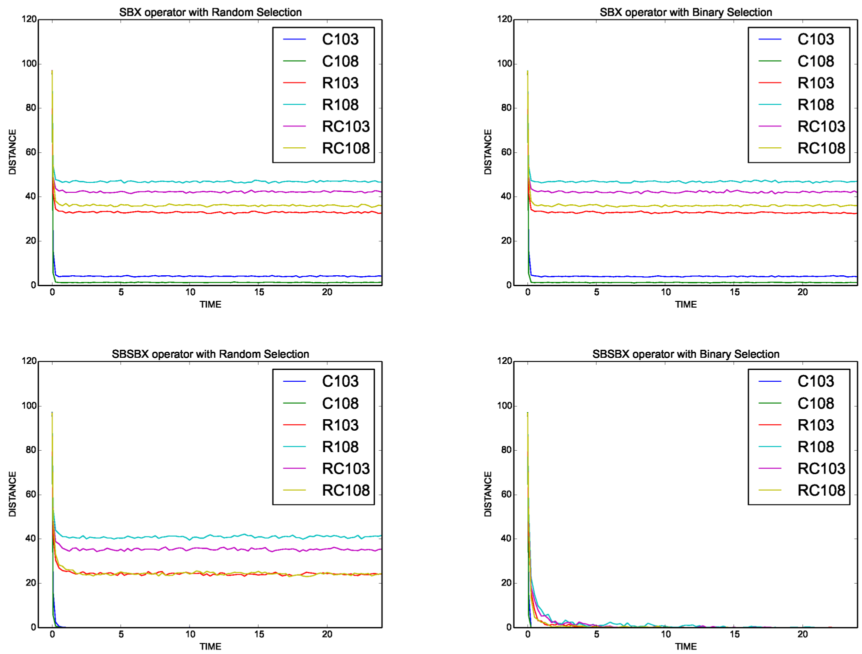

Fig. 2. Diversity maintained with different crossover and parent selection operators

drawn from the same distribution. Significance level of 5% were considered in every case.

The validation is presented in three sub-sections. The first subsection describes the set of instances selected in our experimental validation. The second subsection presents an analysis of the diversity induced by the different parent selection methods and crossover operators. Finally, the third subsection discusses on the performance obtained by the different approaches. In order to better validate our proposal, results are compared with the best-known solutions achieved by any optimizer.

A. Benchmark

The well-known Solomon's benchmark has been used to evaluate the performance of our proposal. The reason is that this benchmark is the most widely used benchmark with CVRPTW. Solomon's benchmark has three different set of instances. Each kind distributes the customers in different ways, as it was previously described. In order to evaluate our proposal with a diverse set of instances, we have selected four instances of each kind.

Solomon's benchmark comprises instances with 25 customers, 50 customers and 100 customers. Additionally, it has been extended to 200, 400, 600, 800 and 1000 customers. In our analyses, instances of 100 customers have been selected because they have large enough search spaces to be quite challenging and they have been used by many practitioners. In the case of larger instances, not many results are available which hinders the development of proper performance analyses.

B. Diversity Analysis

The memetic algorithm proposed has been integrated with two different parent selection operators. In order to analyze their effects on the diversity and quality, executions with six instances were considered. In every case, the population size was set to 50 and for the SA a temperature of 10 is chosen. As a measure of diversity, for each individual the closest one is found. Then, the mean of these distances is reported. Fig. 2 shows the effects of both parent selection and crossover operators on the population diversity. In the case of applying the SBX operator, it is clear that the diversity maintained in the population is not affected by the selection pressure of the parent selection. In fact, regardless of the parent selection, in some instances the diversity maintained in the population is quite large during the entire execution. Thus, when using this operator, trying to control the diversity by inducing different pressures in the parent selection phase is not possible. The reason seems to be that SBX is so disruptive that it induces a large diversity regardless of the action of the additional components of the EA. As a result, this operator never focuses on promising regions, so it might be claimed that MAs with SBX lack of a exploitation phase.

The SBSBX operates quite differently in terms of population diversity. We can appreciate that the diversity decreases to zero when using the binary tournament, meaning that initially the MA starts to explore different regions, and then it intensifies in a selected region. Since binary tournament induces a quite high selection pressure this is the expected behavior. The time to converge highly

TABLE II
Comparison between the SBX and the SBSBX with different parent selection(BKS = Best-Known Solutions)

Instance	BKS	SBX				SBSBX			
		Random		Binary		Random		Binary	
		Best	Mean	Best	Mean	Best	Mean	Best	Mean
C103	826.3 [25]	826.3	**826.3**	826.3	**826.3**	826.3	**826.3**	826.3	**826.3**
C108	827.3 [25]	827.3	**827.3**	827.3	**827.3**	827.3	**827.3**	827.3	**827.3**
R103	1208.7 [26]	1208.7	1210.36	1208.7	1210.46	1208.7	**1208.88**	1208.7	**1208.91**
R108	932.1 [27]	933.7	**936.63**	933.7	**936.74**	933.7	**935.91**	934.3	938.09
RC103	1258 [26]	1258	1258.65	1258	1258.72	1258	**1258.18**	1258	**1259.3**
RC108	1114.2 [25]	1114.2	1114.79	1114.2	1114.57	1114.2	**1114.22**	1114.2	1123.01

TABLE III
Statistical Comparison between the SBX and the SBSBX with the different parent selection(V=Victories, ND=Not-Different, L=Loss)

Instance	V	ND	L	Total
SBSBX_Random	**8**	**10**	**0**	**8**
SBX_Random	2	12	4	-2
SBX_Binary	2	12	4	-2
SBSBX_Binary	2	10	6	-4

depends on the instance, which is not a so desirable feature. For instance, in the case of the Cxxx, the convergence is obtained quite fast. Thus, this scheme might present premature convergence. Finally, in the case of random parent selection, there are some instances where the diversity decrease very fast, whereas in other cases, a large diversity is maintained. Thus, in some instances the operator has the same problem that the SBX, i.e. it never has a clear exploitation phase. However, the amount of diversity maintained is lower than in the case of SBX, meaning that at least it has a relatively more exploitative behavior.

C. Performance Analysis

Table II summarizes the results obtained by the four different tested approaches. Particularly, for each one, the best and mean of the obtained results are shown. Additionally, for each instance, the best-known solution is shown in the column BKS. In each instance, the lowest mean attained by any of the methods is shown in **bold face**. Additionally, the methods that attained results whose differences with respect to the best ones are not statistically significant are also highlighted. The superiority of SBSBX with random selection is clear. In fact, its result are highlighted in every instance.

The impact of the population diversity on the final results depends on the type of instances. In Cxxx instances, which are considered to be the easiest ones, any operator and any parents selection schema works properly. However, in harder instances, there is an impact on the results. The combination of binary tournament and SBSBX does not attain promising results. Since long-term executions are performed, the fast reduction of diversity implies an improper use of the computational resources. However, a too large diversity in not adequate. Thus, the scheme that attains the best results is the one that combines the SBSBX with the random selection, i.e. a non-disruptive operator with a parent selection that induces a low selection pressure.

TABLE IV
SBSBX with random selection(BKS = Best-Known Solutions)

Instance	BKS	Mean	std	Best	Worst
C203	**588.7** [8]	**588.7**	0.0	**588.7**	**588.7**
C208	**585.8** [8]	**585.8**	0.0	**585.8**	**585.8**
R203	**870.8** [27]	871.04	0.5144	**870.8**	872.5
R208	701.2 [27]	702.83	0.9007	**701.0**	704.3
RC203	**923.7** [27]	**923.7**	0.0	**923.7**	**923.7**
RC208	**776.1** [27]	776.45	0.2285	**776.1**	777.3

In order to confirm these findings, pairwise statistical test were performed, i.e. in each instance, all the pairs of algorithms were statistically compared. This means that, when considering the six instances, 18 statistical tests were performed for each scheme. Table III shows the results of this procedure. The column with 'V' indicates the number of victories of the corresponding algorithm listed in each row. The number of cases where differences are not statistically significant are shown in the column 'ND'. Finally, the amount of losses are shown in the column with letter 'L'. The final column shows a score that is calculated as the number of victories minus the number of losses. This table confirms that the adequate algorithm is the one that combines the SBSBX with the random selection.

VI. Conclusions

Finally, in order to better show the promising behavior of SBSBX with random parent selection, tests with additional instances were executed. Table IV summarizes the obtained results. In every case, quite competitive results were achieved. In fact, in one case a new best-known solution could be attained, and only in one case the best solution achieved by any other method could not be achieved. Note that, in this table, for each instance the minimum values are shown in **bold face**.

In this paper, a novel memetic algorithm is proposed to tackle the CVRPTW. The proposal is a MA that uses SA to enhance the quality of the offspring. In addition, a modification of the SBX operator is proposed. This novel operator is the SBSBX. The main feature of the SBSBX is that it is much less disruptive than the SBX operator. The analyses show that when using the SBX operator, controlling the diversity of the population is quite difficult. In fact, regardless of the selection pressure induced in the parent selection, a high diversity is maintained in some instances. Contrarily, in the case of the SBSBX operator, the degree of diversity highly depends on the selection pressure induced by the parent selection. In the case of using a binary tournament, premature convergence arises. However, when combined with a random selection operator, a lower degree of diversity than in the case of the SBX is maintained, but without the appearance of premature convergence.

Our experimental validation performed with a well-known set of benchmarks shows a really promising behavior. The combination of SBSBX and random selection has reported the most promising results. In almost all cases, the results are really close to the best-known solution. In addition, a new best-known solution for the instance R208 could be attained. The statistical tests confirm the superiority of the approach that combines SBSBX and random selection.

Several lines of future work might be explored. First, in order to improve further the performance, a model with an explicit control of diversity might be designed. Finally, in order to analyze the scalability of the proposals, the extensions of Solomon's benchmark might be used.

Acknowledgments

This work has been supported by the Spanish Ministry of Economy, Industry and Competitiveness inside the program "I+D+i Orientada a los Retos de la Sociedad" with contract number TIN2016-78410-R.

References

[1] W. Cook, W. Cunningham, Pulleyblank, and A. Schrijver, *Combinatorial Optimization.* John Wiley & Sons, 1998.

[2] G. B. Dantzig and J. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, pp. 90–91, 1959.

[3] M. Solomon and J. Desrosiers, "Time window constrained routing and scheduling problems," *Transportation Science*, vol. 22, pp. 1–13, 1988.

[4] R. Baldacci, A. Mingozzi, and R. Roberti, "Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints," *Oper. Research*, vol. 1, pp. 1–6, 2012.

[5] N. A. El-Sherbeny, "Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods," *Journal of King Saud University (Science)*, vol. 1, pp. 123–131, 2010.

[6] O. Braysy, W. Dullaert, and M. Gendreau, "Evolutionary algorithms for the vehicle routing problem with time windows," *Journal of Heuristics*, vol. 10, pp. 587–611, 2004.

[7] R. Baños, J. Ortega, C. Gil, F. de Toro, and M. G. Montoya, "Analysis of openmp and mpi implementations of metaheuristics for vehicle routing problems," *Applied Soft Computing*, vol. 43, pp. 262–275, 2016.

[8] B. Kallehauge, J. Larsen, and O. B. Madsen, "Lagrangean duality and non-differentiable optimization applied on routing with time windows - experimental results," in *Internal report IMM-REP-2000-8, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark*, 2000.

[9] J.-Y. Potvin and S. Bengio, "The vehicle routing problem with time windows - part ii: Genetic search," *Informs J Comput*, vol. 8(2), pp. 165–172, 1996.

[10] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Operations Research*, vol. 35, pp. 254–265, 1987.

[11] SINTEF. (2008) Problems and benchmarks, VRPTW. [Online]. Available: http://www.sintef.no/Projectweb/TOP/VRPTW/Homberger-benchmark/

[12] Y. Qi, Z. Hou, H. Li, J. Huang, and X. Li, "A decomposition based memetic algorithm for multi-objective vehicle routing problem with time windows," *Computers & Operations Research*, vol. 62, pp. 61–77, 2015.

[13] C. Segura, C. A. C. Coello, G. Miranda, and C. León, "Using multi-objective evolutionary algorithms for single-objective constrained and unconstrained optimization," *Annals of Oper. Research*, vol. 240, no. 1, pp. 217–250, 2016.

[14] C. Segura, A. H. Aguirre, F. Luna, and E. Alba, "Improving diversity in evolutionary algorithms: New best solutions for frequency assignment," *IEEE Trans. Evol. Comput.*, p. In Press, 2017.

[15] Çağrı Koç, T. Bektaş, O. Jabali, and G. Laporte, "A hybrid evolutionary algorithm for heterogeneous fleet vehicle routing problems with time windows," *Computers & Operations Research*, vol. 64, pp. 11–27, 2015.

[16] R. K. Ahuja, J. B. Orlin, and D. Sharma, "Very large-scale neighborhood search," *International Trans. in Operational Research*, vol. 7, pp. 307–307, 2000.

[17] J. Gromicho, J. van Hoorn, A. Kok, and J. Schutten, "Restricted dynamic programming: a flexible framework for solving realistic vrps," *Computers & Operations Research*, 2011.

[18] A. Goel, "Vehicle scheduling and routing with drivers' working hours," *Computers & Operations Research*, vol. 43(1), pp. 17–26, 2009.

[19] H. Nazif and L. S. Lee, "Optimised crossover genetic algorithm for capacitated vehicle routing problem," *Applied Mathematical Modelling*, vol. 36, pp. 2110–2117, 2012.

[20] S. Ronald, "More distance functions for order-based encodings," *Program and Data Optimization Group*, vol. 1, pp. 558–563, 1998.

[21] R. Martí, M. Laguna, and V. Campos, *Scatter Search vs. Genetic Algorithms: An Experimental Evaluation with Permutation Problems.* Kluwer Academic Publishers, 2005.

[22] M. Desrochers, J. Desrosiers, and M. Solomon, "A new optimization algorithm for the vehicle routing problem with time windows," *Oper. Research*, vol. 40, no. 2, pp. 342–354, 1992.

[23] A. Khachaturyan, S. Semenovskaya, and B. Vainshtein, "Statistical-thermodynamic approach to determination of structure amplitude phases," *Sov.Phys. Crystallography*, vol. 24, pp. 519–524, 1979.

[24] C. Segura, C. A. C. Coello, E. Segredo, and A. H. Aguirre, "A novel diversity-based replacement strategy for evolutionary algorithms," *IEEE Trans. on Cybernetics*, vol. 46, no. 12, pp. 3233–3246, 2016.

[25] J. Desrosiers, O. B. G. Madsen, and F. Soumis, "2-path cuts for the vehicle routing problem with time windows," *Transportation Science*, vol. 1, pp. 147–167, 1999.

[26] W. Cook and J. L. Ritch, "A parallel cutting plane algorithm for the vehicle routing problem with time windows," *Computational and Applied Mathematics*, 1999.

[27] R. Roberti, "Exact Algorithms for Different Classes of Vehicle Routing Problems," Ph.D. dissertation, University of Bologna, 2012.