An Evolutionary Discrete Firefly Algorithm with Novel Operators for Solving the Vehicle Routing Problem with Time Windows

Eneko Osaba, Roberto Carballedo, Xin-She Yang and Fernando Diaz

Abstract An evolutionary discrete version of the Firefly Algorithm (EDFA) is presented in this chapter for solving the well-known Vehicle Routing Problem with Time Windows (VRPTW). The contribution of this work is not only the adaptation of the EDFA to the VRPTW, but also with some novel route optimization operators. These operators incorporate the process of minimizing the number of routes for a solution in the search process where node selective extractions and subsequent reinsertion are performed. The new operators analyze all routes of the current solution and thus increase the diversification capacity of the search process (in contrast with the traditional node and arc exchange based operators). With the aim of proving that the proposed EDFA and operators are effective, some different versions of the EDFA are compared. The present work includes the experimentation with all the 56 instances of the well-known VRPTW set. In order to obtain rigorous and fair conclusions, two different statistical tests have been conducted.

Key words: Firefly Algorithm, Discrete Firefly Algorithm, Vehicle Routing Problem with Time Windows, VRPTW, Combinatorial Optimization

1 Introduction

Nowadays, transportation is a crucial activity for modern society, both for citizens and for business sectors. Regarding the transportation in the business world, the rapid advance of technology has made the logistic increasingly important in this

Eneko Osaba, Roberto Carballedo, Fernando Diaz
Deusto Institute of Technology (DeustoTech), University of Deusto, Av. Universidades 24, Bilbao 48007, Spain e-mail: [roberto.carballedo,e.osaba,fernando.diaz]@deusto.es

Xin-She Yang
School of Science and Technology, Middlesex University, Hendon Campus, London, NW4 4BT, United Kingdom e-mail: x.yang@mdx.ac.uk

area. The fact that anyone in the world can be well connected has led to transport networks that are very demanding, though such networks might be less important in the past. Today, a competitive logistic network can make a huge difference between some companies and others. On the other hand, public transport is used by almost all the population and can thus affect the quality of life. In addition, there are different kinds of public transportation systems, each one with its own characteristics. Nonetheless, all of them share the same disadvantages such as the finite capacity of the vehicles, the geographical area of coverage, and the service schedules and frequencies.

Because of their importance, the development of efficient methods for obtaining a proper logistic, or routing planning solution is a hot topic in the scientific community. In the literature, several areas of knowledge can be related to tackle with transport modelling and optimization issues. However, due to the complex nature of such transport networks, efficient methods are yet to be developed. Therefore, this work attempts to focus on artificial intelligence, one of the these active research areas.

In fact, route planning is one of most studied fields related to artificial intelligence. Problems arisen in this field are usually known as the so-called vehicle routing problems, which are a particular case of problems within combinatorial optimization. Different sorts of VRPs can lead to lots of research work annually in international conferences [1, 2], journals [3, 4, 5], technical reports [6] and edited books [7, 8].

There are many main reasons for such popularity and importance of the routing problems, however, we only highlight two reasons: the social interests they generate and their inherent scientific interests. On the one hand, routing problems are normally designed to deal with real-world situations related to the transport or logistics. Their efficient resolution can lead to profits, either social or business. On the other hand, most of the problems arising in this field have a great computational complexity, typically NP-Hard [9]. Thus, the resolution of these problems is a major challenge for researchers. Probably the most famous problems in this area are the Traveling Salesman Problem [10] and the Vehicle Routing Problem [11].

This chapter focuses on the VRP family of problems. The basic VRP consists of a set of clients, a fleet of vehicles with a limited capacity, and a known depot. The main objective of the VRP is to find the minimum number of routes with the minimum costs such that (i) each route starts and ends at the depot, (ii) each client is visited exactly by one route and (iii) the total demand of the customers visited by one route does not exceed the total capacity of the vehicle that performs the task.

Besides the basic TSP and VRP, many variations of these problems can be found in the literature. The emphasis of this work is one one of these variants: the well-known vehicle routing problem with time windows (VRPTW). In this variant, each client imposes a time window for the start of the service. The VRPTW will be described in detail in the following sections.

In line with this, several appropriate methods can be found in the literature to tackle this kind of problems in a relatively efficient way, especially for small-scale problems. Probably, the most effective techniques are the exact methods [12, 13], heuristics and metaheuristics. In our present work, our focus is nature-inspired metaheuristics. Some classic examples of metaheuristics are tabu search [14], simulated annealing [15], ant colony optimization [16], genetic algorithms (GA) ([17, 18]), and particle swarm optimization [19], though most recent metaheuristic algorithms are population-based. Despite having been proposed many years ago, these techniques remain active and useful with diverse applications ([20, 21, 22]).

Despite of the existence of these well-known techniques, it is still necessary to design new techniques because existing methods can have some disadvantages or still struggle to cope with such tough optimization problems. In fact, new and different metaheuristic algorithms have been proposed in recent years and they have been successfully applied to a wide range of fields and problems. Some examples of these methods are the harmony search proposed by Geem et al. in 2001 [23], the cuckoo search developed by Yang and Deb in 2009 [24, 25], the firefly algorithm developed by Yang in 2008 [27] and the gravitational search algorithm presented by Rashedi in 2009 [26].

The present work focus on the the metaheuristic, called Firefly Algorithm (FA). The FA was developed by Yang in 2008 [27] as a new nature-inspired algorithm based on the flashing behaviour of fireflies. The flashing acts as a signal system to attract other fireflies. As can be shown in several surveys [28, 29], the FA has been applied in several different optimization fields and problems since its proposal, and it still attracts a lot of interests in the current scientific community [30, 31, 32]. Nevertheless, the FA has been rarely applied to any VRP problem. This lack of works, along with the growing scientific interest in bio-inspired algorithms, and the good performance shown by the FA since its proposal in 2008, has motivated its use in this study.

In addition, it is worth mentioning that several novel route optimization operators will be presented in this paper. These operators perform selective extractions of nodes in an attempt to minimize the number of routes in the current solution. For this purpose, the size of the route, the distance of the nodes from the center of gravity of the route or just random criteria are used. Specifically, the experiments presented in this chapter try to delete a route at random and then re-insert the extracted nodes on the remaining routes.

Therefore, in order to prove that our Evolutionary Discrete Firefly Algorithm (EDFA), which uses our proposed novel operators, is a promising technique to solve the well-known VRPTW, experiments with 56 instances have been conducted. In this set of experiments, the performance of several versions of the EDFA will be compared. Besides that, with the aim of drawing fair and rigorous conclusions, in addition to the conventional comparison based on the typical descriptive statistics parameters (results average, standard deviation, best result, etc.), we have also conducted two different statistical tests: the Friedman test and the Holm's test.

The rest of the chapter is organized as follows. In Section 2, a brief background is presented. In Section 3, the basic aspects of the FA are detailed. In addition, in Section 4, an in-depth description of the VRPTW is shown. Then, our proposed EDFA and Route optimization operators are described in Section 5. After that, the experimentation carried out is detailed in Section 6. Then, the chapter finishes with the conclusions of the study and further work in Section 7.

2 Related work

As we have mentioned in the previous section, the FA is a population-based algorithm proposed in 2008 by Yang. The basic FA is based on the flashing behaviour of fireflies, and its first version was proposed for solving continuous optimization problems. Since this first implementation, the FA has been applied in a wide range of areas. Some of these areas are the continuous optimization [33, 34], multi-modal optimization [35, 36] combinatorial optimization [37], and multi-objective optimization [38].

Regarding the application fields in which the FA has proven to be effective, we can list the image processing [39], the antena design [40], civil engineering [41], robotics [42], semantic web [43], chemistry [44], and metereology [45].

In addition, several modifications and hybrid algorithms have been presented in the literature. In [46], for example, a modification called modified Firefly Algorithm is proposed. In [47] and [48], on the other hand, a Chaos randomized firefly algorithm was developed. Besides that, in [49] and [50] two Parallel Firefly Algorithms were presented. Regarding hybrid techniques, in [51] and [52] two FAs hybridized with the GA were developed. Additionally, in [53] an ant colony hybridized with a FA was proposed. Finally, in [54] an approach was presented in which the FA was hybridized with neural networks.

In the present work, we develop a discrete version of the FA. Although the first version FA was designed for continuous problems, it has been modified many times in the literature with the intention of addressing discrete optimization problems. In [55], for instance, we can find a discrete FA adjusted to solve the class of discrete problems named Quadratic Assignment Problem. Another successful discrete FA was developed by Sayady et al. in 2010 [37] for solving minimizing the makespan for the permutation flow shop scheduling problem which is classified as a NP-Hard problem. Another discrete FA was presented in [56] by Marichelvam et al. in 2014 for the multi-objective flexible job shop scheduling problem. On the other hand, a novel evolutionary discrete FA applied to the symmetric TSP was presented in [57].

Despite the huge amount of related works, as we have pointed in the introduction, the FA has been rarely applied to any routing problem. This lack of application, along with the growing scientific interest in bio-inspired algorithms, and the good performance shown by the FA, has been the main motivation of its use in this study. Nevertheless, one of the main originalities of this work is the application field of the FA. Another novelty of our proposed approach is the use of the Hamming Distance function to measure the distance between two fireflies of the swarm. This approach has been used previously in other techniques applied to the TSP, proving its good performance [58], but it has been never used for any EDFA applied to VRPTW. In addition, the movement functions that have been used in the proposed EDFA have been never used before in the literature.

Regarding the VRPTW, the number of publications retated to this problem is really high. For this reason, we can only mention only a fraction of some recently published studies. In [59], an interesting paper published by Desaulniers et al. in 2014 can be found, in which a set of exact algorithms are presented to tackle the electric VRPTW. On other hand, Belhaiza et al. proposed in their work [60] a hybrid variable neighborhood tabu search approach for solving the VRPTW. Besides that, in 2014, a multiple ant colony system was developed for the VRPTW with uncertain travel times by Toklu et al. [61]. Finally, an interesting hybrid generational algorithm for the periodic VRPTW can be found in [62].

Finally, it is worth pointing that the set of papers and books listed in this section is only a small sample of all the related work that can be found in the literature. Because of this huge amount of related works, to summarize all the interesting papers is obviously a complex task. For this very reason, if any reader wants to extend the information presented in this work, we recommend the reading of the review paper presented in [29] about FAs. On the other hand, for additional information about the VRPTW and its solving methods, the work presented in [63, 64] is highly recommended.

3 Firefly Algorithm

The first verion of the FA was developed by Xin-She Yang in 2008 [27, 36], and it was based on the idealized behaviour of the flashing characteristics of fireflies. To understand this method in a proper way, it is important to clarify the following three idealized rules, which have been drawn from [27]:

- All the fireflies of the swarm are unisexual, and one firefly will be attracted to other ones regardless of their sex.
- Attractiveness is proportional to the brightness, which means that, for any two fireflies, the brighter one will attract the less bright one. The attractiveness decreases as the distance between the fireflies increases. Furthermore, if one firefly is the brightest one of the swarm, it moves randomly.
- The brightness of a firefly is directly determined by the objective function of the problem under consideration. In this manner, for a maximization problem, the brightness can be proportional to the objective function value. On the other hand, for a minimization problem, it can be the reciprocal of the objective function value.

The pseudocode of the basic version of the FA is depicted in Algorithm 1. This pseudocode was proposed by Yang in [27]. Consistent with this, there are three crucial factors to consider in the FA: the attractiveness, the distance and the movement. In the basic FA these three factors are addressed in the following way.

Algorithm 1: Pseudocode of the basic version of the FA.

```
Define the objective function f(x);
Initialize the firefly population X = x1, x2, ..., xn;
Define the light absorption coefficient γ;
for each firefly xi in the population do
    Initialize light intensity Ii;
end
repeat
    for each firefly xi in the swarm do
        for each other firefly xj in the swarm do
            if Ij > Ii then
                Move firelfy xi toward xj ;
            end
            Attractiveness varies with distance r via exp(-γr);
            Evaluate new solutions and update light intensity;
        end
    end
    Rank the fireflies and find the current best;
until termination criterion reached;
Rank the fireflies and return the best one;
```

First of all, the attractiveness of a firefly is determined by its light intensity, and it can be calculated as follows:

$$\beta(r) = \beta_0 e^{-\gamma r^2} \tag{1}$$

On the other hand, the distance r_{ij} between two fireflies i and j is determined using the Cartesian distance, and it is computed by this formula:

$$r_{ij} = ||X_i - X_j|| = \sqrt{\sum_{k=1}^{d} (X_{i,k} - X_{j,k})^2} \tag{2}$$

where $X_{i,k}$ is the kth component of the spatial coordinate X_i of the ith firefly in the d-dimensional space. Finally, the movement of firefly i toward any other brighter firefly j is calculated as follows:

$$X_i = X_i + \beta_0 e^{-\gamma r_{ij}^2}(X_j - X_i) + \alpha(rand - 0.5) \tag{3}$$

where α is the randomization parameter and *rand* is a random number uniformly distributed in [0,1]. On the other hand, the second term of the equation stems from the attraction assumption.

4 Vehicle Routing Problem with Time Windows

The VRPTW is an extension of the basic VRP, in which, apart from capacity constraints of each of the vehicles, each client has an associated time window $[e_i,l_i]$. This time window has a lower limit e_i and an upper limit l_i which have to be respected by all the vehicles. In other words, the service in every customer must necessarily start after e_i and l_i before. This would be the variant with hard time windows; there is also another variant that enables noncompliance with some time window (with a penzalizacin in the objective function).

Therefore, a route is not feasible if a vehicle reaches the position of any client after the upper limit of the range. By contrast, the route is feasible whether a vehicle reaches a customer before its lower limit. In this case, the client cannot be served before this limit, so that the vehicle has to wait until e_i. In addition, the central depot has also a time window, which restricts the period of activity of each vehicle in order to adapt to this range. Apart from this temporal window, it can also take into account the customer's service time. This parameter is the time that the vehicle is parked on the client while it is performing the supply. It is a factor to be taken into account to calculate if the vehicle arrives on time to the next customer.

This problem has been widely studied both in the past [63, 65, 66], and nowadays [67, 68]. One reason why the VRPTW is so interesting is its dual nature. It might be considered as a problem of two phases, one phase concerning the vehicle routing and other concerning the planning phase or customer scheduling. Another reason is its easy adaptation to the real world, because in the great majority of distribution chains, customers have strong temporal constraints that have to be fulfilled. For example, in the distribution of the press or of perishable foods these windows are really necessary.

Regarding the mathematical formulation of VRPTW, it can take several forms, using more or less variables [69, 70]. One of the most interesting formulations can be found in [71].

5 Our proposed approach for solving the VRPTW

In this section, the description of our EDFA for the VRPTW is provided (Section 5.1). Besides that, a detailed description of the proposed novel route optimization operators can be found in Section 5.2.

5.1 An Evolutionary Discrete Firefly Algorithm

It is worth mentioning that the original FA was primarily developed for solving continuous optimization problems. This is the reason because the classic FA cannot be applied directly to solve any discrete problem, such as the VRPTW. Hence, some modifications in the flowchart of the basic FA must be conducted with the aim of preparing it for tackling the VRPTW.

First of all, in the proposed EDFA, each firefly in the swarm represents a possible and feasible solution for the VRPTW. In addition, as it is well-known, the VRPTW is a minimization problem. For this reason, the most attractive fireflies are those with a lower objective function value. The concept of light absorption is also represented in this version of the FA. In this case, $\gamma = 0.95$, and this parameter is used in the same way as has been depicted in Equ. (3). This parameter has been set following the guidelines proposed in several studies of the literature [36, 27].

Furthermore, as has been mentioned in the introduction of this paper, the distance between two fireflies is calculated using the well-known Hamming distance. The Hamming distance between two fireflies is the number of non-corresponding elements in the sequence. In the experimentation, VRPTW solutions are represented by a giant-tour, which consists of the client identifiers, being 0 the depot. Thus, the Hamming distance is calculated from the comparison of the order who have the clients in the giant-tour. (excluding the depot). For example, given two solutions (or firefly) problem consisting of 7 nodes:

$$x_1 : \{0,1,2,5,0,3,4,6,7,0\} \rightarrow 1,2,\mathbf{5},\mathbf{3},4,\mathbf{6},\mathbf{7},$$

$$x_2 : \{0,1,2,6,0,7,4,0,5,3,0\} \rightarrow 1,2,\mathbf{6},\mathbf{7},4,\mathbf{5},\mathbf{3},$$

the Hamming Distance between x_1 and x_2 would be 4.

This same example serves to analyze the brightness (light intensity I_i) of a firefly. In this case, the fitness function used is traditional for the VRP. It has two hierarchical objectives: first the number of routes and as a secondary objective the total traveled distance. As shown in the above example, firefly x_1 is better than the firefly x_2 because the former has fewer routes than the second; 2 versus 3. Thus x_2 will be attracted to x_1 using the proposed route optimization operator.

Finally, the movement of firefly i attracted to another brighter firefly j is determined by

$$n = \text{Random}(2, r_{ij} \cdot \gamma^g) \tag{4}$$

where r_{ij} is the Hamming Distance between firefly i and firefly j, and g is the iteration number. In this case, the length of the movement of a firefly will be a random number between 2 and $r_{ij} \cdot \gamma^g$. This value is used to generate n successors from the solution corresponding to the firefly to be moved. Once all successors are generated, the best of them is selected to replace the original firefly. For comparison of different alternatives, two criteria for selecting the best successor will be used: the successor with the best objective function value, or the successor with the lower Hamming distance towards the firefly j (the one that is used as reference to perform the movement of firefly i).

In the proposed EDFA, a single operator to simulate the movement of fireflies is used. This operator is based on the description given in section 5.2 with the following features:

- The *ejectionPool* is initialized with all the nodes assigned to a randomly selected route.
- To speed up the process, the optimization of the remaining routes and the re-insertion into the nearest route phases are not performed.

Furthermore, Regarding the termination criterion, each technique finishes its execution when it reach the generation (iteration) 101, or when there are 20 generations without any improvement in the best solution found.

Finally, after conducting an empirical analysis, the "first-movement" criterion is used to stop the process of attracting a firefly in each global iteration. In this sense, when a firefly at x_i is attracted by other firefly x_j, the movement of x_i during the current iteration is finished. After that, the algorithm continues with the process of the firefly x_{i+1}. This scheme accelerates the whole process without significantly affecting the quality of the final solution obtained by the algorithm.

5.2 Description of the proposed operators

In the context of VRP and its variants there are a number of operators (Or-opt, 2-opt, String-reallocation, String-exchange, GENI-exchange, GENI-CROSS, etc. see [72]) whose objective focuses on the improvement of routes through the exchange of nodes (clients) or paths (sequences of clients) both for a single route and between small groups of routes. These operators perform small modifications to the current solution which allow to control the algorithm computational complexity and runtime. While processing time is an important element, these operators focus their analysis on solutions close to the current solution (intensification capacity) by limiting the space of solutions that are able to explore. This may limit the exploration of the search space avoiding the movement to areas that might contain more promising solutions (diversification ability). In addition, these operators have a limited or negligible capacity to reduce the number of routes since only on rare occasions, the movement of a set of nodes between two routes can leave one of them empty, allowing reduction of the number of routes in the current solution.

On the other hand, there are heuristics that focus their efforts on minimizing the number of routes. These techniques, which have their origin in the "ejection chains" method [73], carried out processes of extraction and reinsertion of nodes on the routes of the current solution. This methods could also remove a complete route in order to minimize the number of routes. In the latter case, probably one of the most representative and successful route minimization heuristics was developed by Nagata and Brysy [74].

Taking as inspiration the concept of "ejection chains", a family of operators whose objective is the reduction of the number of routes has been presented in this work. These operators combine the "ejection chains" technique with other simple measures (such as the size of a route and the proximity with respect to the "centre of gravity of a route"). The proposed operators are initially designed to be integrated into local search processes. In this way, the developed operators increase

Algorithm 2: Pseudocode of the route minimization operator.

input : $Solution_{current}$, $optimizeRoutes$, $proximityReinsertion$
1 $ejectionPool = initEjectionPool(Solution_{current})$;
2 $Solution_{new} = removeEmptyRoutes(Solution_{current})$;
3 **if** $optimizeRoutes$ **then**
4 | $optimizeRoutes(Solution_{new})$;
5 **end**
6 **if** $proximityReinsertion$ **then**
7 | $reinsert(ejectionPool, Solution_{new})$;
8 **end**
9 **if** $ejectionPool \neq \emptyset$ **then**
10 | $Solution_{new} = parallelReconstruction(ejectionPool, Solution_{new})$;
11 **end**
12 **if** $Solution_{new}$ *better than* $Solution_{current}$ **then**
13 | $Solution_{current} = Solution_{new}$;
14 **end**
output: $Solution_{current}$

the diversification ability of the traditional node and arc interchange based operators. After describing the basic notion of the proposed operators, the main characteristics of them are depicted. The descriptions of the operators focus on VRPTW, but they could be easily adapted for any other variant of the VRP.

VRPTW construction heuristics focus their efforts on the generation of an initial solution in a fast and efficient way. This fact hinders the ability to explore the space of solutions, taking irreversible decisions when assigning clients to a vehicle, and sort them in a route. For that reason, after applying a construction heuristic, improvement processes are needed. These processes review allocation and sort decisions to obtain better solutions. This argument is consistent due to the nature of the VRP, but could even explicitly confirmed if the construction process could analyze in detail the structure of the generated routes and the location of customers. For example, after obtaining a solution to a VRP, a person might suggest changes (in the allocations made) visually analyzing the solution. This process could be based on simple calculations to analyze the number of clients of a route, or the proximity between customers that form a route. This notion, combined with random behavior, has been used as a basis for designing the new operators for VRPTW.

The description of the proposed operators is shown in Algorithm 2

- First the $ejectionPool$ is initialized (line 1). This structure is composed by the nodes who are extracted from their original route and will be reinserted again to create the new solution. The construction of the $ejectionPool$ allows several variants to extract nodes from its original location, for instance:
  - Extract the nodes further away from the center of gravity of their original route.
  - Extract all the nodes belonging to the smallest route(s).
  - Extract all the nodes from a randomly selected route.

In the work presented here only the third variant is applied, but other variants can be defined.

- Once the *ejectionPool* is initialized "empty routes elimination" step is performed (line 3). In this phase the routes that are empty after node extraction are eliminated.
- After the removal of empty routes, an optional optimization process is done (line 4). This process is based on an intra-route operator (it modifies a single route). Its aim is to increase the chances for reinserting the extracted nodes. For example the use of Or-opt or 2-opt is suggested. In the experimentation conducted in this work this step is skipped to speed up the overall process.
- After optimizing the remaining routes, the algorithm continues with the "reinsertion phase" (line 7). This phase is also optional. The basic idea would be to reinsert each of the nodes that are part of the *ejectionPool* in its "nearest" route. To perform this reinsertion in an efficient way, the use of neighbor lists is recommended [75].
- As a last step, the final reconstruction of the new solution is performed (line 10). In this phase, a parallel construction heuristic is used. This heuristics combines the routes of the current solution and nodes remaining in the "ejection pool". After invoking the parallel construction heuristic all the nodes are again assigned to a route and the process ends returning the new solution. The reconstruction algorithm could be any construction technique but in this case the one proposed by Campbell and Savelsbergh [76] is used.

With the scheme described above, it can be seen that this new type of operator performs a more complex process than traditional node and arc exchange operators. This can affect the runtime but the proposed operator possesses a good ability to reduce the number of routes that are in the current solution.

Normally, the process of minimizing the number of routes is the last step of a heuristic or a metaheuristic. Actually, in some cases it is run as a completely separate process. But with the new proposed operators this process can be integrated implicitly in the optimization algorithm. In fact, the new operator can be a perfect complement to increase the diversification ability in the population. The proposed operator will be used in the proposed EDFA algorithm to implement the movement of the fireflies in the swarm.

6 Experiments and results

In order to test our proposed approaches properly, the 100 customers Solomon's problems and instances will be used [77]. This set of problems consists of 56 instances classified into 6 categories (C1, C2, R1, R2, RC1 y RC2) which differ in the geographical distributions of the customers, the capacities of the vehicles and the compatibility of the time windows.

Table 1: Results of $EDFA_{FR-HD}$.

Class	T	AVG_V	SD_V	AVG_D	SD_D
C1	3792	11.022	0.050	1716.296	51.919
C2	5119	3.975	0.056	1099.617	25.176
R1	4339	14.033	0.045	1567.214	16.077
R2	7608	3.182	0.000	1325.060	18.094
RC1	2672	14.225	0.105	1847.529	13.604
RC2	4910	3.800	0.112	1600.324	21.278

Although there are VRPTW benchmarks with larger problems instances (such as Gehring & Homberger's[1]), the objective of the work presented focuses on analyzing the adaptation of the EDFA algorithm to the VRPTW. For this reason, Solomon's benchmark is adequate and representative to analyze the behavior of the EDFA applied to VRPTW.

All the tests conducted in this work have been performed on an Intel Core i5 2410 laptop, with 2.30 GHz and a RAM of 4 GB. Java has been used as the programming language.

It is important to point out that the objective function used for the VRPTW is the classic one, which prioritizes the minimization of the routes number, leaving the traveled distance as the second optimizing criterion.

The experimentation has been performed with 4 variants of the proposed EDFA described in Section 5.2: $EDFA_{FR-OF}$, $EDFA_{HR-OF}$, $EDFA_{FR-HD}$ and $EDFA_{HR-HD}$. Such variants differ in the use of two criteria for initializing the swarn of fireflies (Full Random = 100% random and Half Random = 50% random + 50% good solutions) and two criteria to select the best successor to move a firefly x_i towards a firefly x_j (Objective Function = successor with the best value of the objective function, Hamming Distance = successor with the lower Hamming distance from x_j). To create the good initial solutions, Solomon's I1 construction heuristic [77] has been used. Additionally, the initial population size has been set to 50 fireflies. Finally, all the variants of the EDFA have been executed 20 times.

The results of the experimentation are shown in Table 1, Table 2, Table 3 and Table 4. All the tables have the same structure: one row for each class of the Solomon's bechmark (summarizing the results of all the instances of a class) and five columns. Each column corresponds to the average runtime for all the instances of each class (T, in seconds), and average (AVG) and standard deviation (SD) for the number of routes (V) and the total cumulative distance (D).

Table 1 presents the results obtained by $EDFA_{FR-HD}$. This version of the algorithm is characterized by using a completely random initial population. Best successor of firefly x_i is chosen based on the Hamming distance between the successor and firefly x_j (which attracts x_i). According to the experimentation conducted, this variant is the one that gets worse results both in number of vehicles and traveled distance. This confirms the importance of the quality of the initial solution in the VRPTW. On the other hand, the results also serve to justify the

[1] https://www.sintef.no/projectweb/top/vrptw/homberger-benchmark/

Table 2: Results of $EDFA_{FR-OF}$.

Class	T	AVG_V	SD_V	AVG_D	SD_D
C1	3089	10.689	0.093	1513.885	26.913
C2	4300	3.900	0.105	1031.001	32.767
R1	3605	13.667	0.084	1506.030	5.671
R2	6629	3.218	0.050	1288.694	8.178
RC1	2166	13.650	0.105	1734.507	21.553
RC2	4364	3.750	0.088	1590.146	14.793

Table 3: Results of $EDFA_{HR-HD}$.

Class	T	AVG_V	SD_V	AVG_D	SD_D
C1	2887	10.000	0.000	914.323	2.568
C2	4634	3.000	0.000	671.709	2.771
R1	4146	13.250	0.059	1464.889	10.628
R2	7387	3.182	0.000	1261.209	2.617
RC1	2384	12.975	0.105	1633.282	16.704
RC2	4914	3.500	0.000	1499.629	1.925

Table 4: Results of $EDFA_{HR-OF}$.

Class	T	AVG_V	SD_V	AVG_D	SD_D
C1	2480	10.000	0.000	907.105	0.615
C2	3978	3.000	0.000	666.225	2.360
R1	3341	13.188	0.042	1442.712	3.956
R2	6693	3.182	0.000	1243.179	2.507
RC1	2013	12.969	0.063	1568.936	6.000
RC2	4407	3.500	0.000	1490.360	4.891

Hamming distance offers a worse performance than the objective function, to choose the best successor to move a firefly. Like the other variants, standard deviation in relation to the number of vehicles is not very high.

In table 2 the results of $EDFA_{FR-OF}$ are presented. In this case the initial population has been generated 100% at random. Regarding the selection of successors this variant uses the same criteria as $EDFA_{HR-OF}$. This variant improves the foregoing initialization issue and its results are slightly better. However, the two variants are tied for the number of vehicles in the R2 class.

$EDFA_{FR-HD}$ results are shown in Table 3. In this case the initial population is generated 50% randomly and 50% using Solomon's I1 construction heuristic. The best successor to every movement of a firefly is chosen based on Hamming distance. As we can see, this variant performs better than the previous two in terms of the number of vehicles and distance traveled. The improvement is mainly due to the quality of the initial solutions. This confirms the relevance of the initial solution in the VRPTW. Furthermore, by analyzing standard deviations, it can be seen that the values are lower. This implies that this method is also more robust.

Finally, Table 4 shows the results of $EDFA_{HR-OF}$. This variant of the EDFA combines the initialization process of $EDFA_{HR-XX}$ with the selection of the

Table 5: Summary of the results and comparison with $EDFA_{HR-OF}$.

	$EDFA_{HR-OF}$				$EDFA_{HR-HD}$				$EDFA_{FR-OF}$				$EDFA_{FR-HD}$			
	AVG_V	$\%_V$	AVG_D	$\%_D$	AVG_V	$\%_V$	AVG_D	$\%_D$	AVG_V	$\%_V$	AVG_D	$\%_D$	AVG_V	$\%_V$	AVG_D	$\%_D$
C1	**10.000**	0.000	**907.105**	0.000	10.000	0.000	914.323	0.008	10.689	0.064	1513.885	0.401	11.022	0.093	1716.296	0.471
C2	**3.000**	0.000	**666.225**	0.000	3.000	0.000	671.709	0.008	3.900	0.231	1031.001	0.354	3.975	0.245	1099.617	0.394
R1	**13.188**	0.000	**1442.712**	0.000	13.250	0.005	1464.889	0.015	13.667	0.035	1506.030	0.042	14.033	0.060	1567.214	0.079
R2	**3.182**	0.000	**1243.179**	0.000	3.182	0.000	1261.209	0.014	3.218	0.011	1288.694	0.035	3.182	0.000	1325.060	0.062
RC1	**12.969**	0.000	**1568.936**	0.000	12.975	0.000	1633.282	0.039	13.650	0.050	1734.507	0.095	14.225	0.088	1847.529	0.151
RC2	**3.500**	0.000	**1490.360**	0.000	3.500	0.000	1499.629	0.006	3.750	0.067	1590.146	0.063	3.800	0.079	1600.324	0.069

best successor used by $EDFA_{XX-OF}$. This variant is the one that gets the best results. Always gets the best results in terms of traveled distance, and ties with $EDFAHR-HD$ in terms of the number of vehicles (except for the R1 class). Given the characteristics of this variant (and always according to the experimentation carried), it confirms that the objective function is better than the Hamming distance in selecting successors. Furthermore, the quality of the initial solution also affects the final solution: the better the quality of the initial solution, the better the final solution.

To summarize, Table 5 shows the comparison of all variants and the difference from the $EDFA_{HR-OF}$ (which reported the best results). From the table, it can be observed that $EDFA_{HR-OF}$ obtains the best results in terms of distance, having a draw with $EDFA_{HR-HD}$ regarding the number of vehicles. To finish this preliminary analysis, in relation to execution times, all values are quite similar. They are between 300 and 600 seconds to solve an instance of a problem.

Once the results of the experimentation have been presented, two statistical tests (using the number of vehicles and traveled distance) have been made. These tests are based on the guidelines suggested by Derrac et al. [78]. The objective of this task is to ensure that comparisons between the different variants of the EDFA are fair and objective. First, the non-parametric Friedmans test for multiple comparison was conducted. This test aims to check for significant differences between the four variants of the EDFA.

Table 6: Average ranking obtained by the Friedman's test.

Algorithm	AVG_V	AVG_D
$EDFA_{HR-OF}$	1.500	1.166
$EDFA_{HR-HD}$	1.916	1.833
$EDFA_{FR-OF}$	3.000	3.000
$EDFA_{FR-HD}$	3.583	4.000

Table 6 shows the average ranking obtained for each variant (the lower the value, the better the performance of the variant). The test has been conducted for both criteria of the objective function: the number of vehicles and total traveled distance. Regarding the number of vehicles, the resulting Friedman statistic has been 9.95. Taking into account that the confidence interval has been stated at the 97.5% confidence level, the critical point in a χ^2 distribution with 3 degrees of freedom is 9.348. Since 9.95>9.348, it can be concluded that there are significant differences among the results reported by the four compared algorithms, being $EDFA_{HR-OF}$ the one with the lowest rank. Finally, for this Friedman test, the computed p-value has been 0.018996.

Table 7: Adjusted and unadjusted p-values of Holm's test for the number of vehicles.

Algorithm	adjusted p	unadjusted p
$EDFA_{FR-OF}$	0.005189	0.015566
$EDFA_{FR-HD}$	0.044171	0.088343
$EDFA_{HR-HD}$	0.576150	0.576150

Once discovered significant differences in the number of vehicles, it is appropriate to compare technique by technique. For that reason, a post-hoc Holm's test, using $EDFA_{HR-OF}$ as reference (which ranks first in the number of vehicles), has been made. The results of this test are shown in Table 7. As can be seen, only for $EDFA_{FR-OF}$ adjusted and unadjusted p-values are simultaneously less than or equal to 0.05. Therefore, it can be confirmed statistically that the differences in the number of routes for all variants regarding $EDFA_{HR-OF}$ are only significant for $EDFA_{HR-OF}$.

The statistical test of the number of vehicles show no significant differences. For that reason, a new statistical analysis has been performed. This second analysis has been focused on the total traveled distance. This has involved the implementation of new Friedman's and Holm's tests.

Table 8: Adjusted and unadjusted p-values of Holm's test for the total traveled distance.

Algorithm	adjusted p	unadjusted p
$EDFA_{FR-OF}$	0.000144	0.000432
$EDFA_{FR-HD}$	0.013906	0.027813
$EDFA_{HR-HD}$	0.371093	0.371093

For the traveled distance, the resulting Friedman statistic has been 17. In this case 17>9.348, so it can be concluded that there are also significant differences among the results reported by the compared algorithms, being $EDFA_{HR-OF}$ the one with the lowest rank. Finally, regarding this Friedman test, the computed p-value has been 0.000707. In addition, a new Holm test using traveled distance has been performed. $EDFA_{HR-OF}$ has been the reference again. The results confirm the existence of significant differences with respect to $EDFA_{FR-HD}$ and $EDFA_{FR-OF}$; but no significant differences regarding $EDFA_{HR-HD}$ exists. These results confirm the superiority of the initialization good solutions with respect to 100% random initialization. Finally, after combining the results of the two rankings (number of routes and total distance), it may conclude that the $EDFA_{HR-OF}$ variant is the one that gets the best results for the experimentation conducted.

7 Conclusions

In this work, an Evolutionary Discrete Firefly Algorithm applied to the well-known Vehicle Routing Problem with Time Windows has been presented. The proposed technique presents some novelties, such as the use of the Hamming Distance to measure the distance between two different fireflies. Another interesting originality is the novel route optimization operators that have been developed for the EDFA. These operators perform selective extractions of nodes in an attempt to minimize the number of routes in the actual solution. For this, the size of the route, the distance of the nodes from the center of gravity of the route or just random criteria are used. Specifically, the experimentation conducted in the work presented uses an operator that removes a random selected route and then try to reinsert the extracted nodes in the remaining routes.

In order to demonstrate that the proposed EDFA and the developed route optimization operators are promising approaches, the perfomance of the presented EDFA has been compared with those obtained by several versions of the EDFA. For this comparison the 56 instances of 100 customers Solomon's VRPTW bechmark have been used. Besides that, in order to obtaind fair conclusions, two different statistical tests have been performed: the Friedman's Test and the Holm's Test.

As for future work, we plan to extend the experimentation of this study, comparing the performance of the proposed EDFA with those presented by some recently proposed metaheuristic, such as the Bat Algorithm, or the Golden Ball Algorithm [79]. In addition, we intend to use the novel route optimization operators proposed in this work in other recent and classic techniques, such as the Simulated Annealing, or Genetic Algorithm.

Acknowledgements This project was supported by the European Unions Horizon 2020 research and innovation programme through the TIMON: Enhanced real time services for optimized multimodal mobility relying on cooperative networks and open data project (636220); as well as by the projects TEC2013-45585-C2-2-R from the Spanish Ministry of Economy and Competitiveness, and PC2013-71A from the Basque Government.

References

1. Soonpracha, K., Mungwattana, A., Manisri, T.: A re-constructed meta-heuristic algorithm for robust fleet size and mix vehicle routing problem with time windows under uncertain demands. In: Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems, Springer (2015) 347–361
2. Wen, Z., Dong, X., Han, S.: An iterated local search for the split delivery vehicle routing problem. In: International Conference on Computer Information Systems and Industrial Applications, Atlantis Press (2015)
3. Escobar, J.W., Linfati, R., Toth, P., Baldoquin, M.G.: A hybrid granular tabu search algorithm for the multi-depot vehicle routing problem. Journal of Heuristics **20**(5) (2014) 483–509
4. Lin, C., Choy, K.L., Ho, G.T., Chung, S., Lam, H.: Survey of green vehicle routing problem: Past and future trends. Expert Systems with Applications **41**(4) (2014) 1118–1138

5. Reed, M., Yiannakou, A., Evering, R.: An ant colony algorithm for the multi-compartment vehicle routing problem. Applied Soft Computing **15** (2014) 169–176
6. Coelho, L.C., Renaud, J., Laporte, G.: Road-based goods transportation: A survey of real-world applications from 2000 to 2015. Technical report, Technical Report FSA-2015-007, Québec, Canada (2015)
7. Toth, P., Vigo, D.: The vehicle routing problem. Society for Industrial and Applied Mathematics (2015)
8. Laporte, G., Ropke, S., Vidal, T.: Heuristics for the vehicle routing problem. Vehicle Routing: Problems, Methods, and Applications **18** (2014) 87
9. Lenstra, J.K., Kan, A.: Complexity of vehicle routing and scheduling problems. Networks **11**(2) (1981) 221–227
10. Lawler, E.L.: The traveling salesman problem: a guided tour of combinatorial optimization. WILEY-INTERSCIENCE SERIES IN DISCRETE MATHEMATICS (1985)
11. Dantzig, G.B., Ramser, J.H.: The truck dispatching problem. Management science **6**(1) (1959) 80–91
12. Laporte, G.: The traveling salesman problem: An overview of exact and approximate algorithms. European Journal of Operational Research **59**(2) (1992) 231–247
13. Laporte, G.: The vehicle routing problem: An overview of exact and approximate algorithms. European Journal of Operational Research **59**(3) (1992) 345–358
14. Glover, F.: Tabu search, part i. ORSA Journal on computing **1**(3) (1989) 190–206
15. Kirkpatrick, S., Gellat, C., Vecchi, M.: Optimization by simmulated annealing. science **220**(4598) (1983) 671–680
16. Dorigo, M., Blum, C.: Ant colony optimization theory: A survey. Theoretical computer science **344**(2) (2005) 243–278
17. Goldberg, D.: Genetic algorithms in search, optimization, and machine learning. Addison-Wesley Professional (1989)
18. De Jong, K.: Analysis of the behavior of a class of genetic adaptive systems. PhD thesis, University of Michigan, Michigan, USA (1975)
19. Kennedy, J., Eberhart, R., et al.: Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks. Volume 4., Perth, Australia (1995) 1942–1948
20. Rodriguez, A., Gutierrez, A., Rivera, L., Ramirez, L.: Rwa: Comparison of genetic algorithms and simulated annealing in dynamic traffic. In: Advanced Computer and Communication Engineering Technology. Springer (2015) 3–14
21. Cao, B., Glover, F., Rego, C.: A tabu search algorithm for cohesive clustering problems. Journal of Heuristics (2015) 1–21
22. İnkaya, T., Kayalıgil, S., Özdemirel, N.E.: Ant colony optimization based clustering methodology. Applied Soft Computing **28** (2015) 301–311
23. Geem, Z.W., Kim, J.H., Loganathan, G.: A new heuristic optimization algorithm: harmony search. Simulation **76**(2) (2001) 60–68
24. Yang, X.S., Deb, S.: Cuckoo search via lévy flights. In: World Congress on Nature & Biologically Inspired Computing, IEEE (2009) 210–214
25. Yang, X.S., Deb, S.: Engineering optimisation by cuckoo search. International Journal of Mathematical Modelling and Numerical Optimisation **1**(4) (2010) 330–343
26. Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S.: Gsa: a gravitational search algorithm. Information sciences **179**(13) (2009) 2232–2248
27. Yang, X.S.: Nature-inspired metaheuristic algorithms. Luniver press, UK (2008)
28. Fister, I., Yang, X.S., Fister, D., Fister Jr, I.: Firefly algorithm: A brief review of the expanding literature. In: Cuckoo Search and Firefly Algorithm. Springer (2014) 347–360
29. Fister, I., Fister Jr, I., Yang, X.S., Brest, J.: A comprehensive review of firefly algorithms. Swarm and Evolutionary Computation (2013)
30. Ma, Y., Zhao, Y., Wu, L., He, Y., Yang, X.S.: Navigability analysis of magnetic map with projecting pursuit-based selection method by using firefly algorithm. Neurocomputing (2015)
31. Liang, R.H., Wang, J.C., Chen, Y.T., Tseng, W.T.: An enhanced firefly algorithm to multi-objective optimal active/reactive power dispatch with uncertainties consideration. International Journal of Electrical Power & Energy Systems **64** (2015) 1088–1097

32. Zouache, D., Nouioua, F., Moussaoui, A.: Quantum-inspired firefly algorithm with particle swarm optimization for discrete optimization problems. Soft Computing (2015) 1–19
33. Yang, X.S.: Metaheuristic optimization: algorithm analysis and open problems. In: Experimental Algorithms. Springer (2011) 21–32
34. Yang, X.S.: Efficiency analysis of swarm intelligence and randomization techniques. Journal of Computational and Theoretical Nanoscience **9**(2) (2012) 189–198
35. Das, S., Maity, S., Qu, B.Y., Suganthan, P.N.: Real-parameter evolutionary multimodal optimizationa survey of the state-of-the-art. Swarm and Evolutionary Computation **1**(2) (2011) 71–88
36. Yang, X.S.: Firefly algorithms for multimodal optimization. In: Stochastic algorithms: foundations and applications. Springer (2009) 169–178
37. Sayadi, M., Ramezanian, R., Ghaffari-Nasab, N.: A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems. International Journal of Industrial Engineering Computations **1**(1) (2010) 1–10
38. Abedinia, O., Amjady, N., Naderi, M.S.: Multi-objective environmental/economic dispatch using firefly technique. In: International Conference on Environment and Electrical Engineering, IEEE (2012) 461–466
39. Zhang, Y., Wu, L.: A novel method for rigid image registration based on firefly algorithm. International Journal of Research and Reviews in Soft and Intelligent Computing (IJRRSIC) **2**(2) (2012)
40. Basu, B., Mahanti, G.K.: Fire fly and artificial bees colony algorithm for synthesis of scanned and broadside linear array antenna. Progress In Electromagnetics Research B **32** (2011) 169–190
41. Talatahari, S., Gandomi, A.H., Yun, G.J.: Optimum design of tower structures using firefly algorithm. The Structural Design of Tall and Special Buildings **23**(5) (2014) 350–361
42. Jakimovski, B., Meyer, B., Maehle, E.: Firefly flashing synchronization as inspiration for self-synchronization of walking robot gait patterns using a decentralized robot control architecture. In: Architecture of Computing Systems-ARCS 2010. Springer (2010) 61–72
43. Pop, C.B., Rozina Chifu, V., Salomie, I., Baico, R.B., Dinsoreanu, M., Copil, G.: A hybrid firefly-inspired approach for optimal semantic web service composition. Scalable Computing: Practice and Experience **12**(3) (2011)
44. Fateen, S.E.K., Bonilla-Petriciolet, A., Rangaiah, G.P.: Evaluation of covariance matrix adaptation evolution strategy, shuffled complex evolution and firefly algorithms for phase stability, phase equilibrium and chemical equilibrium problems. Chemical Engineering Research and Design **90**(12) (2012) 2051–2071
45. Santos, A.F.d., Campos Velho, H.F.d., Luz, E.F., Freitas, S.R., Grell, G., Gan, M.A.: Firefly optimization to determine the precipitation field on south america. Inverse Problems in Science and Engineering **21**(3) (2013) 451–466
46. Tilahun, S.L., Ong, H.C.: Modified firefly algorithm. Journal of Applied Mathematics **2012** (2012)
47. Gandomi, A., Yang, X.S., Talatahari, S., Alavi, A.: Firefly algorithm with chaos. Communications in Nonlinear Science and Numerical Simulation **18**(1) (2013) 89–98
48. Coelho, L.D.S., de Andrade Bernert, D.L., Mariani, V.C.: A chaotic firefly algorithm applied to reliability-redundancy optimization. In: IEEE Congress on Evolutionary Computation, Ieee (2011) 517–521
49. Subutic, M., Tuba, M., Stanarevic, N.: Parallelization of the firefly algorithm for unconstrained optimization problems. Latest Advances in Information Science and Applications (2012) 264–269
50. Husselmann, A.V., Hawick, K.: Parallel parametric optimisation with firefly algorithms on graphical processing units. In: Proc. Int. Conf. on Genetic and Evolutionary Methods. (2012) 77–83
51. Farahani, S.M., Abshouri, A.A., Nasiri, B., Meybodi, M.: Some hybrid models to improve firefly algorithm performance. International Journal of Artificial Intelligence **8**(S12) (2012) 97–117

52. Luthra, J., Pal, S.K.: A hybrid firefly algorithm using genetic operators for the cryptanalysis of a monoalphabetic substitution cipher. In: World Congress on Information and Communication Technologies, IEEE (2011) 202–206
53. Aruchamy, R., Vasantha, K.: A comparative performance study on hybrid swarm model for micro array data. International Journal of Computer Applications **30**(6) (2011) 10–14
54. Hassanzadeh, T., Faez, K., Seyfi, G.: A speech recognition system based on structure equivalent fuzzy neural network trained by firefly algorithm. In: International Conference on Biomedical Engineering, IEEE (2012) 63–67
55. Durkota, K.: Implementation of a discrete firefly algorithm for the qap problem within the sage framework. BSc thesis, Czech Technical University (2011)
56. Marichelvam, M.K., Prabaharan, T., Yang, X.S.: A discrete firefly algorithm for the multi-objective hybrid flowshop scheduling problems. Evolutionary Computation, IEEE Transactions on **18**(2) (2014) 301–305
57. Jati, G.K., et al.: Evolutionary discrete firefly algorithm for travelling salesman problem. Adaptive and Intelligent Systems (2011)
58. Zhou, L., Ding, L., Qiang, X.: A multi-population discrete firefly algorithm to solve tsp. In: Bio-Inspired Computing-Theories and Applications. Springer (2014) 648–653
59. Desaulniers, G., Errico, F., Irnich, S., Schneider, M.: Exact algorithms for electric vehicle-routing problems with time windows. Les Cahiers du GERAD G-2014-110, GERAD, Montréal, Canada (2014)
60. Belhaiza, S., Hansen, P., Laporte, G.: A hybrid variable neighborhood tabu search heuristic for the vehicle routing problem with multiple time windows. Computers & Operations Research **52** (2014) 269–281
61. Toklu, N.E., Gambardella, L.M., Montemanni, R.: A multiple ant colony system for a vehicle routing problem with time windows and uncertain travel times. Journal of Traffic and Logistics Engineering Vol **2**(1) (2014)
62. Nguyen, P.K., Crainic, T.G., Toulouse, M.: A hybrid generational genetic algorithm for the periodic vehicle routing problem with time windows. Journal of Heuristics **20**(4) (2014) 383–416
63. Kallehauge, B., Larsen, J., Madsen, O.B., Solomon, M.M.: Vehicle routing problem with time windows. Springer (2005)
64. Gendreau, M., Tarantilis, C.D.: Solving large-scale vehicle routing problems with time windows: The state-of-the-art. CIRRELT (2010)
65. Potvin, J.Y., Bengio, S.: The vehicle routing problem with time windows part ii: genetic search. INFORMS journal on Computing **8**(2) (1996) 165–172
66. Bräysy, O., Gendreau, M.: Vehicle routing problem with time windows, part i: Route construction and local search algorithms. Transportation science **39**(1) (2005) 104–118
67. Afifi, S., Guibadj, R.N., Moukrim, A.: New lower bounds on the number of vehicles for the vehicle routing problem with time windows. In: Integration of AI and OR Techniques in Constraint Programming. Springer (2014) 422–437
68. Agra, A., Christiansen, M., Figueiredo, R., Hvattum, L.M., Poss, M., Requejo, C.: The robust vehicle routing problem with time windows. Computers & Operations Research **40**(3) (2013) 856–866
69. Azi, N., Gendreau, M., Potvin, J.Y.: An exact algorithm for a single-vehicle routing problem with time windows and multiple routes. European journal of operational research **178**(3) (2007) 755–766
70. Bräysy, O., Gendreau, M.: Tabu search heuristics for the vehicle routing problem with time windows. Top **10**(2) (2002) 211–237
71. Cordeau, J.F., Desaulniers, G., Desrosiers, J., Solomon, M.M., Soumis, F.: Vrp with time windows. The vehicle routing problem **9** (2001) 157–193
72. Brysy, O., Gendreau, M.: Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms. Transportation Science **39**(1) (February 2005) 104–118
73. Rego, C.: Node-ejection chains for the vehicle routing problem: Sequential and parallel algorithms. Parallel Computing **27**(3) (February 2001) 201–222

74. Nagata, Y., Brysy, O.: A powerful route minimization heuristic for the vehicle routing problem with time windows. Operations Research Letters **37**(5) (September 2009) 333–338
75. Irnich, S.: A Unified Modeling and Solution Framework for Vehicle Routing and Local Search-Based Metaheuristics. INFORMS Journal on Computing **20**(2) (January 2008) 270–287
76. Campbell, A.M., Savelsbergh, M.: Efficient Insertion Heuristics for Vehicle Routing and Scheduling Problems. Transportation Science **38**(3) (August 2004) 369–378
77. Solomon, M.M.: Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations research **35**(2) (1987) 254–265
78. Derrac, J., García, S., Molina, D., Herrera, F.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation **1**(1) (2011) 3–18
79. Osaba, E., Diaz, F., Onieva, E.: Golden ball: a novel meta-heuristic to solve combinatorial optimization problems based on soccer concepts. Applied Intelligence **41**(1) (2014) 145–166