On Efficiently Solving the Vehicle Routing Problem with Time Windows Using the Bat Algorithm with Random Reinsertion Operators

Eneko Osaba, Roberto Carballedo, Xin-She Yang, Iztok Fister Jr., Pedro Lopez-Garcia and Javier Del Ser

Abstract An evolutionary and discrete variant of the Bat Algorithm (EDBA) is proposed for solving the Vehicle Routing Problem with Time Windows, or VRPTW. The EDBA developed not only presents an improved movement strategy, but it also combines with diverse heuristic operators to deal with this type of complex problems. One of the main new concepts is to unify the search process and the minimization of the routes and total distance in the same operators. This hybridization is achieved by using selective node extractions and subsequent reinsertions. In addition, the new approach analyzes all the routes that compose a solution with the intention of enhancing the diversification ability of the search process. In this study, several variants of the EDBA are shown and tested in order to measure the quality of both metaheuristic

E. Osaba (✉) · R. Carballedo · P. Lopez-Garcia
Deusto Institute of Technology (DeustoTech), University of Deusto,
Av. Universidades 24, 48007 Bilbao, Spain
e-mail: e.osaba@deusto.es

R. Carballedo
e-mail: roberto.carballedo@deusto.es

P. Lopez-Garcia
e-mail: p.lopez@deusto.es

X.-S. Yang
School of Science and Technology, Middlesex University,
Hendon Campus, London NW4 4BT, UK
e-mail: x.yang@mdx.ac.uk

I. Fister Jr.
Faculty of Electrical Engineering and Computer Science, University of Maribor,
Smetanova 17, 2000 Maribor, Slovenia
e-mail: iztok.fister1@um.si

J. Del Ser
TECNALIA, 48160 Derio, Spain

J. Del Ser
University of the Basque Country (UPV/EHU), 48013 Bilbao, Spain

J. Del Ser
Basque Center for Applied Mathematics (BCAM), 48009 Bilbao, Spain
e-mail: javier.delser@tecnalia.com

X.-S. Yang (ed.), *Nature-Inspired Algorithms and Applied Optimization*,
Studies in Computational Intelligence 744, https://doi.org/10.1007/978-3-319-67669-2_4

algorithms and their operators. The benchmark experiments have been carried out by using the 56 instances that compose the 100 customers Solomon's benchmark. Two statistical tests have also been carried out so as to analyze the results and draw proper conclusions.

Keywords Bat algorithm · Discrete bat algorithm · Vehicle routing problem with time windows · VRPTW · Combinatorial optimization · Traveling salesman problem

1 Introduction

The rapid advance of technology has made the logistic management increasingly important in ever-increasingly connected societies, which has led transport networks to be very demanding. To meet such demands, companies have to be innovative in designing their logistic networks, and a competitive logistic network can make the difference between some companies and others. Consequently, the development of efficient methods for proper logistics and routing planning is a hot topic in the research community.

To model and optimize a logistic network, all relevant issues have to be addressed in an appropriate way using appropriate techniques. In this case, we focus our attention here on one of these areas: artificial intelligence. In fact, route planning problems and their resolution is one of the most recurrent topics related to artificial intelligence. More specifically, the problems arisen in this field are normally named as routing problems, and they fall into the combinatorial optimization category. The most studied problems in this field are the Vehicle Routing Problem (VRP) and the Traveling Salesman Problem (TSP). Besides the basic TSP and VRP, many variations of these problems can be found in the literature. In this chapter, the attention is focused on one of these variants: the Vehicle Routing Problem with Time Windows, or VRPTW. Briefly speaking, in the VRPTW, each client imposes a time window for the start and the end of the service. This problem will be explained in greater detail later.

A few solution methods can be found in the literature to deal with this kind of problems properly. The most well-known approaches for this purpose are probably the exact methods [1], heuristics and metaheuristics. Here, we focus our attention on metaheuristic methods. For example, some classical examples of local search-based methods are Simulated Annealing [2] and Tabu Search [3]. On the other hand, population-based techniques such as the Ant Colony Optimization [4], Genetic Algorithms (GA) [5, 6], and Particle Swarm Optimization [7] are some of the most used alternatives.

Although classical techniques can somehow manage to solve certain class of such problems, they are not sufficiently effective, and thus the development of novel metaheuristics for tackling optimization problems, especially for routing problems, is a hot topic in this area of research. Consequently, many different methods have been

proposed in recent years. Some examples of these methods are the Imperialist Competitive Algorithm, proposed by Atashpaz-Gargari and Lucas in 2007 [8], the Artificial Bee Colony, presented by Karaboga and Basturk in 2007 [9], and the bat algorithm developed by Yang in 2010 [11]. To the interested readers, some additional successful methods will be described in following sections.

For the current study, the method that we have selected for addressing the above mentioned VRPTW is the Bat Algorithm (BA). This metaheuristic is a nature-inspired algorithm, based on the echolocation behavior of micro-bats, which was proposed by Yang in 2010 [10]. From the review of some of the recent literature [11, 12], the BA has been successfully applied to wide variety of optimization fields and problems since its proposal. Furthermore, recent works such as [13, 14] confirm that BA still attracts a lot of interest from the scientific community. In this sense, despite the fact that the BA has been applied to many different optimization problems up to date, it has not been applied yet to the well-known VRPTW. Thus, this motivates us to carry out this current work. The detailed explanation of BA will be given in following sections.

It is worth highlighting that we have used some novel route optimization operators for enhancing the performance of the developed algorithm. These operators, which will be described in following sections, perform selective extractions of nodes in an attempt of minimizing the number of routes of the current solution. At this moment, these operators have only been used once in the literature, inside a Firefly Algorithm [15]. For this reason, this is the first time in the literature that such heuristic functions are used in the BA for routing problems.

For the purpose of proving that the implemented Evolutionary Discrete Bat Algorithm (EDBA) is a promising approach to solve the VRPTW, an experiment composed by 56 different instances has been conducted in this work. The results obtained by some variants of the EDBA are compared. In addition, two different statistical tests have been conducted with the results obtained: the non-parametric Friedmans test for multiple comparisons, and the post-hoc Holm's test.

Therefore, the rest of the paper is organized as follows. Section 2 presents the related background with an emphasis on routing problems and nature-inspired metaheuristics for their resolution. After that, in Sect. 3, the philosophy of the basic BA is detailed. Then, in Sect. 4, a brief description of the VRPTW can be found. Then, the proposed EDBA and our route optimization operators are described in Sect. 5. Furthermore, in Sect. 6, the experimentation performed for the validation of the study is detailed. Finally, the paper concludes with with suggestions for further work in Sect. 7.

2 Background

Nowadays, route planning is one of the most studied fields. Problems arisen in this field are usually known as vehicle routing problems, which are a particular case of problems of combinatorial optimization. Probably, the most used and well-known

routing problems are the Traveling Salesman Problem [16] and the Vehicle Routing Problem [17], which are the focus of a huge amount of studies in the literature [18, 19]. In addition, the VRPTW is the main problem of our attention here and is also one of the most cited and used, as can be seen in different works such as [20] and [21].

The reasons for the popularity and importance of these problems are two folds: the scientific aspect, and the social one. On the one hand, being NP-Hard, most of the problems arising in this field have an extraordinary complexity, and thus their solutions pose a major challenge for the scientific community. On the other hand, routing problems are usually built to address a real-world situation related to logistics or transportation, which is directly linked to the profit of a business service.

Even the problems are challenging to solve, several approaches can be found in the literature to tackle this kind of problems. The exact methods [1, 22], heuristics and metaheuristics have all been attempted. For example, as can be seen in the work by Braysy and Gendreau in 2005 [23], metaheuristics are a good approach for solving the VRPTW.

To be more specific within the category of metaheuristics, nature-inspired methods are among the most used approaches for tackling this sort of problems in the current literature [24]. In this sense, some of these recently proposed approaches that can be classified in this category are the Bat Algorithm (BA), Firefly Algorithm (FA), and Cuckoo Search (CS). The first one, and the one that is used in this work, is the BA. This metaheuristic was proposed by Yang in 2010 [10], and it is based on the echolocation behavior of microbats, which can find their prey and discriminate different kinds of insects even in complete darkness. Recent literature reviews and surveys [11, 12] show that BA has been successfully applied to different optimization fields and problems since its proposal. Focusing in routing problems, several recently published papers have shown that the BA is a promising technique also in this field. For example, in [25], which was published in 2015, an adapted variant of this algorithms for solving the well-known Capacitated VRP. The Adapted BA developed in that study allows a large diversity of the population and a balance between global and local search.

A more recent work is proposed in [26] by Zhou et al. in which the same Capacitated VRP is faced. In their paper, a hybrid BA with path relinking is described. This approach is constructed based on the framework of the continuous BA, in which the greedy randomized adaptive search procedure and path relinking are effectively integrated. Additionally, with the aim of improving the performance of the technique, the random subsequences and single-point local search are operated with a certain probability.

Regarding the second of above mentioned methods, that is FA, proposed by Yang in 2008 [27]. This a nature-inspired algorithm is based on the flashing behavior of fireflies, which acts as a signaling system to attract other fireflies. This metaheuristic algorithm has been also applied to a wide range of optimization fields and problems since its proposal [28, 29]. Like the BA, this method has also shown a promising

performance for routing problems. In [30], for example, the first application of the FA was presented for solving the TSP. In order to do that, the authors adapted the FA, which was firstly proposed for tackling continuous problems, enhancing it with an evolutionary and discrete behavior.

Another interesting example of application is the one presented in [31], in which a hybrid variant of the FA is proposed to solve a time-dependent VRP with multi-alternative graph, in order to reduce the fuel consumption. The developed variant of FA is a Gaussian Firefly Algorithm. The most interesting part of that paper is the real-world case study, focused on a distribution company, established in Esfahan, Iran. More recently, FA has been compared with other nature-inspired heuristics for a bi-objective variant of the classical VRP problem with pickup and delivery deadlines, multiple concurrent vehicles and selectivity of nodes. Interestingly, in their work, the quality of routes is determined by the Pareto trade-off between the profit gained by the delivery of goods along the routes and a measure of fairness in the share of the revenues of the transport company [32].

The third of the algorithms previously mentioned is the CS, developed by Yang and Deb in 2009 [33]. It was inspired by the obligate brood parasitism of some cuckoo species by laying their eggs in the nests of other host birds (of other species such as warblers). The CS has also been modified to solve routing problems, as can be seen, for example, in the work published in 2014 by Ouaarab et al. [34]. In that paper, the authors presented the first adaptation of the CS to the well-known TSP, creating a discrete variant of the CS with promising results. The authors also tested their proposed discrete CS against a set of benchmarks of symmetric TSP from the well-known TSPLIB library.

More examples of the CS applied to the VRP can be found in the literature. In [35], for example, a discrete CS algorithm for the capacitated VRP is presented. The main novelty of this method is not only its application itself, but also the Taguchi-based Parameter Setting developed for the parameter optimization. Besides that, in 2016, the reputable Information Sciences journal published a paper in which four different soft computing methods were applied for solving also the Capacitated VRP [36]. One of these approaches was an advanced CS, which introduced new adjustments and features for improving its efficiency. Another example is the paper presented by Chen and Wang in 2016 [37], in which a hybrid CS was proposed for the solving the VRP in logistics distribution systems. This novel algorithm was based on the combination of Optical Optimization, Particle Swarm Optimization and CS. Specifically, in their method, optical optimization was introduced to initialize population for obtaining a group of initial values with high quality, which were then optimized according to PSO. After each iterative operation for keeping the optimal individual, CS was used to optimize the rest of the individuals.

Another metaheuristic is a music-inspired Harmony Search (HS). This technique was firstly proposed by Geem et al. in 2001 as a phenomenon-mimicking metaheuristic [38], inspired by the improvisation process of jazz musicians. There are a wide range of applications of HS in the literature [39–41]. The HS has also been applied

to routing problems several times, showing also a promising performance. The paper presented by Geem et al. in 2005 collected some of the most interesting works up to that date on this topic [42]. Another research related to the HS was the one that can be found in [43], which presented a discrete variant of the HS in order to solve the challenging the Selective Pick-Up and Delivery VRP with Delayed Drop-Off. Additionally, the recent work by Bounzidi and Riffi in 2014 described the adaptation of the HS for solving the TSP.

Another meta-heuristic mentioned in this background section is the Gravitational Search algorithm (GS), proposed by Rashedi et al. in 2009 [44], and it was based on the metaphor of gravitational interaction between masses. GS has also been used in many applications [45–47]. Concerning routing problems, Nodehi et al. in 2016 [48] presented a randomized GS algorithm for the solving of the TSP. The GS implemented in this work was based on randomized search concepts using two of the four main parameters of velocity and gravitational force in physics. The performance of the developed method was compared with some additional well-known methods, such as the Genetic Algorithm, showing a promising performance.

Regarding VRP problems, the work [49] explored the application of a discrete variant of the GS to the Open VRP. Being firstly proposed to solve continuous problems, the main challenge of the authors of that paper was to adapt all the characteristics of the basic variant of the GS to the discrete optimization. As has been mentioned, the problem to solve in this case is the Open VRP, which is a variant in which vehicles are not required to return to the depot. Finally, the paper by Hosseinabad et al. in 2017 [50] presented another approach of the GS to solve the Capacitated VRP with enhanced performance.

There are many challenging issues related to VRPTW, and the number of publications related to this problem is increasing. In [51], for example, Desaulniers et al. presented a set of exact algorithms to tackle the electric VRPTW. On the other hand, Belhaiza et al. proposed in their work [52] a hybrid variable neighborhood tabu search approach for solving the VRPTW. A multiple ant colony system was developed for the VRPTW with uncertain travel times by Toklu et al. [53]. Finally, an a hybrid generational algorithm for the periodic VRPTW can be found in [54]. In relation to the above mentioned nature-inspired methods and the VRPTW, in [15], an evolutionary discrete firefly algorithm was proposed for the resolution of this problem, using the same operators in the experimentation. An additional paper is the one presented by [55], in which a hybrid variant of the HS was presented to deal with the VRPTW.

Since the literature in this area is expanding, it is not possible to review all the relevant work. Interested readers can refer to literature reviews in [11] about the BA, [29] about FA, and [56] about the CS. On the other hand, for additional information about the VRPTW and its solution methods, the work presented in [57, 58] is highly recommended. As mentioned in the introduction, this present work is the first time in the literature that the BA is applied to the VRPTW. In the rest of this chapter, we will describe our proposed approach in greater detail.

Algorithm 1: Pseudo code of the basic BA

```
1  Define the objective function f(x);
2  Initialize the bat population X = x_1, x_2, ..., x_n;
3  for each bat x_i in the population do
4      Initialize the pulse rate r_i, velocity v_i and loudness A_i;
5      Define the pulse frequency f_i at x_i;
6  end
7  repeat
8      for each bat x_i in the population do
9          Generate new solutions through Equations (1), (2) and (3);
10         if rand>r_i then
11             Select one solution among the best ones;
12             Generate a local solution around the best one;
13         end
14         if rand<A_i and f(x_i)<f(x_*) then
15             Accept the new solution;
16             Increase r_i and reduce A_i;
17         end
18     end
19 until termination criterion not reached;
20 Rank the bats and return the current best bat of the population;
```

3 Bat Algorithm

In this section, the basic variant of the BA is fully described before we proceed to introduce further modifications and enhancements. As we have briefly mentioned in previous sections, the BA is a nature-inspired metaheuristic, whose main idea is to imitate the echolocation features of microbats with some idealized rules outlined as follows [10]:

- All bats use echolocation to detect the distance and can differentiate between an obstacle and a prey (bad or good solutions, respectively).
- All bats fly randomly with a velocity v_i at position x_i with a varying frequency from $f_{\min}$ to $f_{\max}$, loudness A_i and pulse emission rate r.
- In the real-world, the loudness and emission rates of bats can vary in many different ways. Here, we assume that the loudness varies monotonically from A_0 to a lower (quieter) value $A_{\min}$, while r varies from a lower value to a higher value.

The main steps of this BA are summarized as the pseudocode as shown in Algorithm 1. Taking a quick look at this pseudo-code, it can be seen that the first six lines correspond to the initialization process. First, the objective function is defined, and the initial population is initialized. Each bat of the population represents a possible solution to the addressed problem, in this case, the VRPTW. After that, velocity v_i, frequency f_i, pulse rate r_i and loudness A_i parameters are initialized and defined.

After this initialization phase, the main evolution of solutions in the algorithm are executed. At each generation, each bat of the swarm moves through the search space

by updating its velocity and position. More specifically, the following equations are used for this movement:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \tag{1}$$

$$v_i^t = v_i^{t-1} + [x_i^{t-1} - x_*]f_i \tag{2}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{3}$$

where β is a uniformly distributed random number in [0,1], and x_* represents the current best solution of the whole population. In addition, v_i^t and x_i^t denote the velocity and position, respectively, of a bat i at time step t. Furthermore, the results of Eq. (1) is used to control the pace and range of bats movement.

If a solution is selected among the best ones, a new solution for each bat is generated using a random walk

$$x_{new} = x_{old} + \varepsilon A^t \tag{4}$$

where ε is a randomly generated number within the interval $[-1, 1]$, and A^t is the average loudness of the swarm at time step t. Finally, the rate r_i and the loudness A_i of each bat are updated, only if the conditions shown in the line 14 of Algorithm 1 are met. This update is performed as follows:

$$r_i^{t+1} = r_i^0[1 - \exp(-\gamma t)] \tag{5}$$

$$A_i^{t+1} = \alpha A_i^t \tag{6}$$

where α and γ are constants. Thereby, for any $0 < \alpha < 1$ and $\gamma > 0$ we have

$$A_i^t \to 0, r_i^t \to r_i^0, \ as\ t \to \infty \tag{7}$$

In most cases in the literature, $\alpha = \gamma$ is used in order to simplify the implementation of the method. In the present study, $\alpha = \gamma = 0.98$ is used. We have selected this value after an empirical experiment using a range of values from 0.90 to 0.99.

4 Vehicle Routing Problem with Time Windows

As we have pointed out in Sect. 2, the VRPTW is an extension of the classic and widely studied VRP. In addition to the basic constraints inherent from the VRP, each client that composes a VRPTW instance has an associated time window $[e_i,l_i]$. More specifically, this time window has a lower limit e_i and an upper limit l_i which must be respected by the vehicle that will attend the demand of the client. This means that the service in every customer must be performed after e_i and before l_i.

Obviously, a route is not feasible if a vehicle tries to serve any customer after the upper limit of this range. On the other hand, a route would be feasible if the vehicle reaches a client before its lower limit. In this last special situation, the client cannot be served before this limit, so that the vehicle should be waiting until e_i to start the delivery.

Besides that, the central depot, which is the starting and ending point of all the routes and vehicles, has also a time window, which restricts the period of the whole activity. Apart from this temporal window, the problem can also take into account the customer's service time. This parameter is the time that the vehicle needs to spend on the client in order to perform the delivery properly. This is a factor to be taken into account to calculate if the vehicle arrives on time to the next customer. Furthermore, the variant that we are using in this paper is the VRPTW with hard time windows. In this sense, there is also another variant that enables noncompliance with some time window (with a penalization in the objective function).

Being one of the most famous variant of the VRPs, this problem has been widely studied both in the past [20, 21], and nowadays [59, 60]. One reason why the VRPTW is so interesting is its dual nature, since it is considered as a two phase problem. The first of these phases concerns the vehicle routing, while the second one regards the planning phase or customer scheduling.

An additional reason for its popularity is its easy adaptation to the real-world applications. The great majority of distribution chains, customers have strong temporal constraints that have to be fulfilled, and the VRPTW perfectly fits with this kind of real-world situations.

Regarding the mathematical formulation of VRPTW, it can take several forms, using a different amount of variables [61, 62]. One of the most interesting formulations can be found in [63].

5 Our Proposed Approach for Solving the VRPTW

In this section, the description of our EDBA for the VRPTW is provided (Sect. 5.1). A more detailed description of the proposed novel route optimization operator will be given in Sect. 5.2.

5.1 An Evolutionary Discrete Bat Algorithm

Before starting with the description of our proposed method, it is worth mentioning that the original BA was firstly developed for solving continuous optimization problems, and thus the standard BA cannot be directly applied to solve any discrete problem such as the VRPTW. Hence, some modifications in the structure of the basic BA should be performed in order to prepare it to solve the VRPTW.

First, in the EDBA, each bat of the swarm represents a possible and feasible solution for the VRPTW. Since the VRPTW is a minimization problem, the most attractive bats are those with a lower objective function value. Regarding the philosophy of both r_i and A_i parameters, it has remained exactly in the same form as that in the standard BA. Furthermore, with the intention of simplifying the complexity of the algorithm, the parameter f_i has not been considered.

Furthermore, the "velocity", v_i, has been modified. In the continuous variant of the BA, this parameter is calculated as has been shown in Eq. (2). However, this formula cannot be used in the same way for solving a discrete problem such as the VRPTW. Thus, we have related v_i to a distance measure between the bat i and the best bat of the swarm. It is worth pointing out that all the quantities are treated as unitless, and thus there is no need to worry about the unit of velocity. Obviously, the true physical quantities have units and the solutions will be given the right units when the final solutions are interpreted. Thus, all the quantities in BA are considered as mathematical values without units. For this purpose, we have adapted v_i using the well-known Hamming Distance in the following way:

$$v_i^t = \text{Random}[1, \text{HammingDistance}(x_i^t, x_*)] \tag{8}$$

This means that the v_i of a bat i at time step t is a random number, which follows a discrete uniform distribution between 1 and the difference between this i and the best bat of the swarm. This difference is represented by the Hamming Distance, which is the number of non-corresponding elements in the sequence. A detailed example of this application can be found in [15].

Additionally, regarding the new bats generation, in the classic variant of the Bat Algorithm the movement of the bats is performed using the Eq. (3). Similar with the v_i parameters, this equation cannot be applied directly to a discrete problem such as the VRPTW. Thus, a modification has been proposed, and the movement of a bat i is determined by the following equation:

$$x_i^t \leftarrow \mathit{MovementFunction}(x_i^{t-1}, v_i^t) \tag{9}$$

In other words, every bat examines at every generation a v_i number of its neighbors, and it chooses the best one as its current movement. Explained in other way, the bat i conducts a v_i number of movements, and it chooses the best one. In the proposed EDBA, a single operator to simulate the movement of bats is used. This operator is described in the next section.

Furthermore, regarding the local search procedure represented in Lines 10–12 of Algorithm 1, whether $rand > r_i$, one solution is randomly chosen among the best ones (in our performed experiments, one bat among the 10 best ones; or less, if v_i is lower than 10), and a local solution is generated around this one, using the well-known 2-opt* operator. After that, if the new solution is accepted, it replaces the current bat.

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≠∅ 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
```

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

5.2 Description of the Bat Movement Operator

In this section the operator used to simulate the movement of the bats is described. This operator is responsible for creating the neighbor solutions generated when a bat is performed its movement (Line 9 of the Algorithm 1).

Using the inspiration by the concept of "ejection chains" [64], a family of operators (whose objective is the reduction of the number of routes) have been presented in a previous work related to the Firefly algorithm [65]. These operators combine the "ejection chains" technique with other simple measures (such as the size of a route and the proximity of the customers with respect to the "center of gravity of a route"). The proposed operators were designed to increase the diversification ability of the traditional node and arc interchange based operators.

Using the results obtained in our previous work focused on Firefly Algorithm [65], in the present work we center our attention only on one operator: the "Random Route Elimination Operator—RrE-opt". As the name suggests, the operator is based on the removal of a route at random and the subsequent reinsertion of the clients of that route in the remaining routes. The main objective is to reduce the number of routes. This is the first criterion of the classical evaluation function for VRPTW.

Figure 1 illustrates a simple worked example of the RrE-opt operator. Furthermore, Algorithm 2 shows the description of this operator:

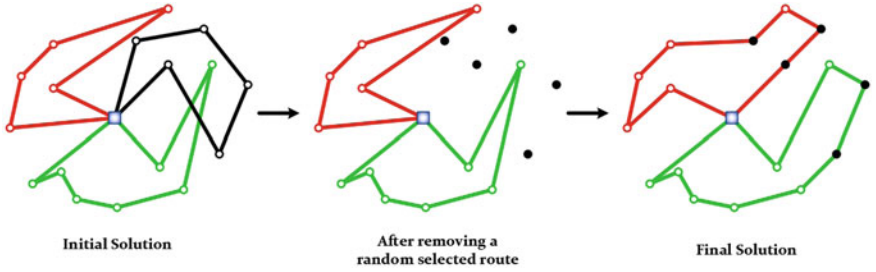

Fig. 1 Example of the RrE-opt operator

- In the first step, a route is selected at random and it is removed from the current solution. The clients that were part of the removed route configure the *ejectionPool*. In the next steps the aim is to reinsert the customers in the remaining routes.
- After the route removal, two optional processes can be performed:
  - A local route optimization using the well-known Or-opt operator. The objective of this process is the reordering of the remaining routes to facilitate the reinsertion of the customers of the *ejectionPool*. Other optimization operators could be used but the Or-opt operator has been chosen for its speed and efficiency.
  - The reinsertion of the customers by proximity in the closest route. This process checks all clients that are in the *ejectionPool* and tries to insert them into the geographically most surrounded route. In this way, the total distance traveled tries to be reduced. This is the second criterion of the VRPTW evaluation function. To perform this reinsertion in an efficient way, the use of neighbor lists is recommended [66].
- The last step is to use a parallel initialization heuristic to reinsert clients that are still in the *ejectionPool*. In this step the heuristic of Campbell and Savelsbergh [67] is used for its speed and simplicity of implementation.

This new operator performs a more complex process than traditional VRPTW operators, but in spite of being more expensive in runtime, this operator has a great ability to reduce the number of routes during the search process. Reducing the number of routes in the context of VRPTW is often done as an independent process. With the proposed new operator, this process is implicitly integrated into the search process.

In the experimentation section below, four variants of the proposed EDBA will be compared. These variants will allow the evaluation of the two optional processes of the operator for the reduction of the number of routes. Its nomenclature will be: OR (only Optimize Routes process), PR (only Proximity Reinsertion process), FULL (both optional processes) and NONE (no optional process).

6 Experimentation

In this section the details of the experimentation conducted are described. The experimentation has two clear objectives: first, to show the use of the proposed EDBA algorithm; and second, to analyze the behavior of the new operator to reduce the number of routes for the VRPTW.

For the experimentation, Solomon's VRPTW benchmark has been used [68]. This set of problems consists of 56 instances of 100 customers classified into 6 categories (C1, C2, R1, R2, RC1 y RC2). The categories differ in the geographical distribution of the customers, the capacity of the vehicles and the compatibility of the time windows.

There are other VRPTW benchmarks with larger problems instances (such as Gehring & Homberger's[1]), but the objective of the work presented focuses on the use of the EDBA and the analysis of the new optimization operator for the VRPTW. For this reason, Solomon's benchmark is adequate and representative.

All the tests conducted in this work have been performed on an Intel Core i5-6200U CPU @ 2.40 GHz with 8 GB of RAM. The algorithms have been programed in Java and double precision is used for all numeric variables and parameters. The used operating systems has been Windows 7.

The evaluation function used is the classic hierarchical one that prioritizes first the number of routes (the minimum the best) and then the total travel distance (again the lower the best).

The experimentation has been performed with 4 variants of the proposed EDBA. Such variants differ in the use (or not) of the optional processes included in the optimization operator presented in Sect. 5.2. They are identified as: EDBA-OR (only Optimize Routes process), EDBA-PR (only Proximity Reinsertion process), EDBA-FULL (both optional processes) and EDBA-NONE (no optional process).

The parameterization for the EDBA used in the experimentation is the following:

- The swarm of bats (population) is composed of 25 individuals.
- The initial population is initialized at random.
- The termination criterion is: a maximum of 100 iterations or 20 iterations without improvement.
- New solutions are generated with the new operator described in Sect. 5.2.
- The local solution around the best new solution is generated using the well-known 2-opt* operator.
- α and β have been initialized to with 0.98.
- r_i^0 for each bat of the population has been initialized with a random value between 0.0 and 0.40.
- A_0 has been set with a random value between 0.70 and 1.0 for each bat.
- v_i has been initialized with a random value between 0.0 and the Hamming Distance between a bat and the best solution found.

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

Table 1 Results obtained by EDBA-OR

Class	T	AVG_V	SD_V	AVG_D	SD_D
C1	7921	10.978	0.093	1512.744	35.109
C2	14625	3.200	0.209	779.528	43.174
R1	9350	14.367	0.162	1529.738	6.871
R2	18043	3.164	0.041	1211.082	12.836
RC1	4525	14.925	0.112	1915.900	15.860
RC2	12862	3.750	0.153	1467.878	15.434

Table 2 Results obtained by EDBA-PR

Class	T	AVG_V	SD_V	AVG_D	SD_D
C1	266	12.889	0.091	2270.032	102.794
C2	1054	4.563	0.217	1817.001	148.536
R1	192	18.167	0.152	2193.207	41.194
R2	1421	4.704	0.087	1965.860	79.367
RC1	112	19.219	0.157	2645.416	32.932
RC2	728	5.594	0.120	2206.801	10.821

Table 3 Results obtained by EDBA-FULL

Class	T	AVG_V	SD_V	AVG_D	SD_D
C1	1375	10.967	0.105	1531.749	39.450
C2	4737	3.725	0.079	889.098	22.066
R1	1212	14.533	0.137	1634.549	30.242
R2	7040	3.237	0.064	1297.255	20.965
RC1	696	15.100	0.184	1960.489	35.536
RC2	4123	3.825	0.121	1566.401	39.975

Finally, in order to calculate proper statistics, each variant of the EDBA has been executed 10 times.

The results of the experimentation are shown in Tables 1, 2, 3 and 4. All the tables have the same structure: one row for each class of the Solomon's benchmark (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 vehicles (V) and the total cumulative travel distance (D).

Table 1 presents the results obtained by EDBA-OR. This variant of the algorithm is characterized by using only the route optimization process. This means that once the *ejectionPool* is generated, the routes that remain in the solution are optimized (using the Or-opt operator) to facilitate the reinsertion of the customers of the

Table 4 Results obtained by EDBA-NONE

Class	T	AVG_V	SD_V	AVG_D	SD_D
C1	1338	12.867	0.093	2183.340	64.583
C2	3385	4.575	0.112	1766.683	131.954
R1	1162	18.217	0.173	2206.293	26.755
R2	4461	4.854	0.138	1878.603	32.055
RC1	624	19.175	0.190	2702.481	109.543
RC2	2680	5.500	0.088	2293.172	66.782

removed route. According to the experimentation conducted, this variant obtained the best results (both in vehicles and traveled distance) for all the classes except C1. For the Class C1, this variant obtained the best results in terms of distance and the number of vehicles is only about 0.1% worse than the best one. The results obtained are consistent since the standard deviation for both vehicles and for distance does not exceed 6.5%. The results obtained confirm that the local optimization of the routes before reinserting the clients of the ejectionPool allows to obtain better solutions. However, the runtime time is significantly higher than the other variants.

In Table 2 the results of EDBA-PR are presented. In this case only nearest reinsertion process is performed. After the removal of the random selected route and before the final parallel initialization, the customers in the *ejectionPool* try to be reinserted in the geographically closest path. This variant is the fastest. However, together with the EDBA-NONE variant, it reports the worst results being 35.5% and 62% worse (than the best results) in terms of number of vehicles and total distance traveled.

EDBA-FULL results are shown in Table 3. In this case both processes are performed (route optimization and proximity reinsertion processes are carried out). This has obtained the second best results. The average percentage differences in number of vehicles and total distance traveled (for all classes) are 3.85% and 5.7%, respectively. In addition, it is the one that obtains the best result in number of vehicles for the class C1. Furthermore, analyzing standard deviations, it can be seen that the values obtained are the lowest. This implies that this method is more robust. One last important fact is the runtime. This variant obtains values significantly better than those obtained by the EDBA-OR variant.

Finally, Table 4 shows the results of EDBA-NONE. In this variant the customers of the removed route are reinserted directly using the parallel construction heuristic without any extra process. This variant, like EDBA-PR, gets poor results that are (on average for all classes) 36% worse in number of vehicles and 56% worse in distance traveled. On the other hand, the execution times are slightly higher than the EDBA-PR variant, but smaller than any of the two variants that get the best results. Finally, analyzing the standard deviations of the obtained results can be said that the algorithm is consistent (like the rest of variants).

To summarize, Table 5 shows the comparison of all variants and the difference with respect to the EDBA-OR (which reported the best results).

Table 5 Summary of the results and comparison between all the methods

	EDBA-OR				EDBA-FULL				EDBA-PR				EDBA-NONE			
	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	10.978	0.1	**1512.744**	0.0	**10.967**	0.0	1531.749	1.3	12.889	17.5	2270.032	50.1	12.867	17.3	2270.032	50.1
C2	**3.200**	0.0	**779.528**	0.0	3.725	16.4	889.098	14.1	4.563	42.6	1817.001	133.1	4.575	43.0	1817.001	133.1
R1	**14.367**	0.0	**1592.738**	0.0	14.533	1.2	1634.549	2.6	18.167	26.4	2193.207	37.7	18.217	26.8	2193.207	37.7
R2	**3.164**	0.0	**1211.082**	0.0	3.237	2.3	1297.255	7.1	4.704	48.7	1965.860	62.3	4.854	53.4	1965.860	62.3
RC1	**14.925**	0.0	**1915.000**	0.0	15.100	1.2	1960.489	2.3	19.219	28.8	2645.416	38.1	28.5	0.088	2645.416	38.1
RC2	**3.750**	0.0	**1467.878**	0.0	3.825	2.0	1566.401	6.7	5.594	49.2	2206.801	50.3	5.500	46.7	2206.801	50.3

Table 6 Average ranking obtained by the Friedman's test

Algorithm	AVG_V	AVG_D
EDBA-OR	1.1667	1
EDBA-FULL	1.8333	2
EDBA-PR	3.5	3.5
EDBA-NONE	3.5	3.5

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. [69]. The objective of this task is to ensure that comparisons between the different variants of the EDBA are fair and objective. First, the non-parametric Friedman's test for multiple comparison was conducted. This test aims to check for significant differences between the four variants of the EDBA.

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: number of vehicles and total traveled distance. Regarding the number of vehicles, the resulting Friedman statistic has been 15.2. Taking into account that the confidence interval has been stated at the 99.5% confidence level, the critical point in a χ^2 distribution with 3 degrees of freedom is 12.838. Because $15.2 > 12.838$, it can be concluded that there are significant differences among the results reported by the four compared algorithms, being EDBA-OR the one with the lowest rank. Finally, for this Friedman's test, the computed p-value has been 0.001653. On the other hand, in relation to the distance, the resulting Friedman statistic has been 16.2. In this case, taking the same confidence interval, the differences are again significant; and the EDBA-OR variant is the one that reports the best results. In this case, the computed p-value is 0.001032. These results confirm the superiority of the EDBA-OR variant.

Once discovered significant differences in the number of vehicles, it is appropriate to compare technique by technique. For this reason, a post-hoc Holm's test, using EDBA-OR as reference (which ranks first in number of vehicles), has been made. The results of this test are shown in Table 7. As can be seen, for EDBA-PR and EDBA-NONE adjusted and unadjusted p-values are simultaneously less than or equal to 0.05. Therefore, it can be confirmed statistically that the difference in the number of routes for EDBA-PR and EDBA-NONE with respect to EDBA-OR is significant. The same does not happen between the EDBA-FULL and EDBA-OR variants.

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

Algorithm	Adjusted p	Unadjusted p
EDBA-PR	0.005235	0.001745
EDBA-NONE	0.005235	0.001745
EDBA-FULL	0.371093	0.371093

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

Algorithm	Adjusted p	Unadjusted p
EDBA-PR	0.002389	0.000796
EDBA-NONE	0.002389	0.000796
EDBA-FULL	0.179712	0.179712

To conclude our statistical analysis, new Holm's tests has been performed. In this case the test is related to the traveled distance. The results of this test are depicted in Table 8. In this case, related to the traveled distance, there are significant differences between EDBA-PR and EDBA-NONE with respect to EDBA-OR.

Finally, as a conclusion of the experimentation and the subsequent statistical analysis of the results, it can be ensured that the EDBA-OR variant is the one that obtains the best results. These results are statistically better than those obtained by the EDBA-PR and EDBA-NONE variants. On the contrary, the results obtained by the EDBA-FULL variant are worse than those obtained by EDBA-OR. But the difference in results is not statistically significant.

7 Conclusions

We have presented in this work an Evolutionary Discrete Bat Algorithm for solving the famous Vehicle Routing Problem with Time Windows. The developed method presents some originality, such as the use of the Hamming distance to measure the distance between two bats (solutions) of the swarm, and the application of some recently proposed optimization operators, which have been firstly used in a BA. Specifically, these operators perform selective extractions of nodes in an attempt to minimize the number of routes in the current solution.

With the intention of validating that the proposed EDBA and the used route optimization operators are effective for solving the VRPTW, the results obtained by the EDBA has been compared with the ones obtained by different variants of the technique. For this experimentation, the 56 instances of the well-known Solomon's VRPTW benchmark have been used. Furthermore, two different statistical tests have been performed in order to enrich the conclusions: the non-parametric Friedmans test for multiple comparisons, and the post-hoc Holm's test.

The opportunities for future work related to the research presented in this paper are broad. For example, more complex benchmarks and further comparison of the performance of the proposed EBFA with other metaheuristics can be carried. In addition, it may be useful to apply the route optimization heuristic operators described in this work to other techniques (including classic techniques) such as the Genetic Algorithm or the Tabu Search, in order to test their efficiency. Furthermore, it can be expected that the proposed approach and operators can also used to solve travelling salesman problems and other combinatorial optimization problems.

References

1. Laporte, G.: The vehicle routing problem: an overview of exact and approximate algorithms. Eur. J. Operat. Res. **59**(3), 345–358 (1992)
2. Kirkpatrick, S., Gellat, C., Vecchi, M.: Optimization by simmulated annealing. Science **220**(4598), 671–680 (1983)
3. Glover, F.: Tabu search, part I. ORSA J. Comput. **1**(3), 190–206 (1989)
4. Dorigo, M., Blum, C.: Ant colony optimization theory: a survey. Theor. Comput. Sci. **344**(2), 243–278 (2005)
5. Goldberg, D.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Professional (1989)
6. De Jong, K.: Analysis of the behavior of a class of genetic adaptive systems. PhD thesis, University of Michigan, Michigan, USA (1975)
7. Kennedy, J., Eberhart, R., et al.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks. Vol. 4., Perth, Australia, pp. 1942–1948 (1995)
8. Atashpaz-Gargari, E., Lucas, C.: Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: Evolutionary Computation, 2007. CEC 2007. IEEE Congress on, IEEE pp. 4661–4667 (2007)
9. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. J. Glob. Optim. **39**(3), 459–471 (2007)
10. Yang, X.S.: A new metaheuristic bat-inspired algorithm. In: Nature Inspired Cooperative Strategies for Optimization. Springer pp. 65–74 (2010)
11. Yang, X.S., He, X.: Bat algorithm: literature review and applications. Int. J. Bio-Ins. Comput. **5**(3), 141–149 (2013)
12. Parpinelli, R.S., Lopes, H.S.: New inspirations in swarm intelligence: a survey. Int. J. Bio-Ins. Comput. **3**(1), 1–16 (2011)
13. Dhar, S., Alam, S., Santra, M., Saha, P., Thakur, S.: A novel method for edge detection in a gray image based on human psychovisual phenomenon and bat algorithm. In: Computer, Communication and Electrical Technology. CRC Press, pp. 3–7 (2007)
14. Tharakeshwar, T., Seetharamu, K., Prasad, B.D.: Multi-objective optimization using bat algorithm for shell and tube heat exchangers. Appl. Therm. Eng. **110**, 1029–1038 (2017)
15. Osaba, E., Carballedo, R., Yang, X.S., Diaz, F.: An evolutionary discrete firefly algorithm with novel operators for solving the vehicle routing problem with time windows. In: Nature-Inspired Computation in Engineering. Springer, pp. 21–41 (2016)
16. Lawler, E.L.: The traveling salesman problem: a guided tour of combinatorial optimization. Wiley-Interscience Series in Discrete Mathematics (1985)
17. Christofides, N.: The vehicle routing problem. RAIRO Operat. Res. Recherche Opérationnel. **10**(V1), 55–70 (1976)
18. Wassan, N., Wassan, N., Nagy, G., Salhi, S.: The multiple trip vehicle routing problem with backhauls: formulation and a two-level variable neighbourhood search. Comput. Operat. Res. **78**, 454–467 (2017)
19. Veenstra, M., Roodbergen, K.J., Vis, I.F., Coelho, L.C.: The pickup and delivery traveling salesman problem with handling costs. Eur. J. Operat. Res. **257**(1), 118–132 (2017)
20. Bräysy, O., Gendreau, M.: Vehicle routing problem with time windows, part I: route construction and local search algorithms. Transport. Sci. **39**(1), 104–118 (2005)
21. Potvin, J.Y., Bengio, S.: The vehicle routing problem with time windows part II: genetic search. INFORMS J. Comput. **8**(2), 165–172 (1996)
22. Laporte, G.: The traveling salesman problem: an overview of exact and approximate algorithms. Eur. J. of Oper. Res. **59**(2), 231–247 (1992)
23. Bräysy, O., Gendreau, M.: Vehicle routing problem with time windows, part II: metaheuristics. Transport. Sci. **39**(1), 119–139 (2005)
24. Yang, X.S.: Nature-inspired metaheuristic algorithms. Luniver press (2010)
25. Taha, A., Hachimi, M., Moudden, A.: Adapted bat algorithm for capacitated vehicle routing problem. Int. Rev. Comput. Soft. (IRECOS) **10**(6), 610–619 (2015)

26. Zhou, Y., Luo, Q., Xie, J., Zheng, H.: A hybrid bat algorithm with path relinking for the capacitated vehicle routing problem. In: Metaheuristics and Optimization in Civil Engineering. Springer, pp. 255–276 (2016)
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, pp. 347–360 (2014)
29. Fister, I., Fister Jr., I., Yang, X.S., Brest, J.: A comprehensive review of firefly algorithms. Swarm Evolut. Comput. **13**, 34–46 (2013)
30. Jati, G.K., Suyanto. In: Evolutionary Discrete Firefly Algorithm for Travelling Salesman Problem. Springer, Berlin Heidelberg, pp. 393–403 (2011)
31. Alinaghian, M., Naderipour, M.: A novel comprehensive macroscopic model for time-dependent vehicle routing problem with multi-alternative graph to reduce fuel consumption: a case study. Comput. Indust. Eng. **99**, 210–222 (2016)
32. Del Ser, J., Torre-Bastida, A.I., Lana, I., Bilbao, M.N., Perfecto, C.: Nature-inspired heuristics for the multiple-vehicle selective pickup and delivery problem under maximum profit and incentive fairness criteria. In: IEEE Congress on Evolutionary Computation (2017)
33. Yang, X.S., Deb, S.: Cuckoo search via lévy flights. In: World Congress on Nature & Biologically Inspired Computing. IEEE, pp. 210–214 (2009)
34. Ouaarab, A., Ahiod, B., Yang, X.S.: Discrete cuckoo search algorithm for the travelling salesman problem. Neural Comput. Appl. **24**(7–8), 1659–1669 (2014)
35. Alssager, M., Othman, Z.A.: Taguchi-based parameter setting of cuckoo search algorithm for capacitated vehicle routing problem. In: Advances in Machine Learning and Signal Processing. Springer, pp. 71–79 (2016)
36. Teymourian, E., Kayvanfar, V., Komaki, G.M., Zandieh, M.: Enhanced intelligent water drops and cuckoo search algorithms for solving the capacitated vehicle routing problem. Informat. Sci. **334**, 354–378 (2016)
37. Chen, X., Wang, J.: A novel hybrid cuckoo search algorithm for optimizing vehicle routing problem in logistics distribution system. J. Comput. Theor. Nanosci. **13**(1), 114–119 (2016)
38. Geem, Z.W., Kim, J.H., Loganathan, G.: A new heuristic optimization algorithm: harmony search. Simulation **76**(2), 60–68 (2001)
39. Manjarres, D., Landa-Torres, I., Gil-Lopez, S., Del Ser, J., Bilbao, M.N., Salcedo-Sanz, S., Geem, Z.W.: A survey on applications of the harmony search algorithm. Eng. Appl. Artific. Intell. **26**(8), 1818–1831 (2013)
40. Assad, A., Deep, K.: Applications of harmony search algorithm in data mining: a survey. In: Proceedings of Fifth International Conference on Soft Computing for Problem Solving. Springer , pp. 863–874 (2016)
41. Mohd Alia, O., Mandava, R.: The variants of the harmony search algorithm: an overview. Artific. Intell. Rev. **36**(1), 49–68 (2011)
42. Geem, Z.W., Lee, K.S., Park, Y.: Application of harmony search to vehicle routing. Am. J. Appl. Sci. **2**(12), 1552–1557 (2005)
43. Del Ser, J., Bilbao, M.N., Perfecto, C., Salcedo-Sanz, S.: A harmony search approach for the selective pick-up and delivery problem with delayed drop-off. In: Harmony Search Algorithm. Springer, pp. 121–131 (2016)
44. Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S.: Gsa: a gravitational search algorithm. Informat. Sci. **179**(13), 2232–2248 (2009)
45. Precup, R.E., David, R.C., Petriu, E.M., Radac, M.B., Preitl, S.: Adaptive gsa-based optimal tuning of pi controlled servo systems with reduced process parametric sensitivity, robust stability and controller robustness. IEEE Trans. Cybernet. **44**(11), 1997–2009 (2014)
46. Precup, R.E., David, R.C., Petriu, E.M., Preitl, S., Rădac, M.B.: Fuzzy logic-based adaptive gravitational search algorithm for optimal tuning of fuzzy-controlled servo systems. IET Control Theor. Appl. **7**(1), 99–107 (2013)
47. Duman, S., Güvenç, U., Sönmez, Y., Yörükeren, N.: Optimal power flow using gravitational search algorithm. Energy Convers. Manag. **59**, 86–95 (2012)

48. Nodehi, A.N., Fadaei, M., Ebrahimi, P.: Solving the traveling salesman problem using randomized gravitational emulation search algorithm. J. Curr. Res. Sci. **2**, 818 (2016)
49. Hosseinabadi, A.A.R., Kardgar, M., Shojafar, M., Shamshirband, S., Abraham, A.: Gravitational search algorithm to solve open vehicle routing problem. In: Innovations in Bio-Inspired Computing and Applications. Springer, pp. 93–103 (2016)
50. Hosseinabadi, A.A.R., Rostami, N.S.H., Kardgar, M., Mirkamali, S., Abraham, A.: A new efficient approach for solving the capacitated vehicle routing problem using the gravitational emulation local search algorithm. Appl. Mathemat, Model (2017)
51. 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)
52. Belhaiza, S., Hansen, P., Laporte, G.: A hybrid variable neighborhood tabu search heuristic for the vehicle routing problem with multiple time windows. Comput. Operat. Res. **52**, 269–281 (2014)
53. 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. J. Traffic Logist. Eng. **2**(1) (2014)
54. Nguyen, P.K., Crainic, T.G., Toulouse, M.: A hybrid generational genetic algorithm for the periodic vehicle routing problem with time windows. J. Heurist. **20**(4), 383–416 (2014)
55. Yassen, E.T., Ayob, M., Nazri, M.Z.A., Sabar, N.R.: Meta-harmony search algorithm for the vehicle routing problem with time windows. Informat. Sci. **325**, 140–158 (2015)
56. Yang, X.S., Deb, S.: Cuckoo search: recent advances and applications. Neural Comput. Appl. **24**(1), 169–174 (2014)
57. Kallehauge, B., Larsen, J., Madsen, O.B., Solomon, M.M.: Vehicle routing problem with time windows. Springer (2005)
58. Gendreau, M., Tarantilis, C.D.: Solving large-scale vehicle routing problems with time windows: the state-of-the-art. CIRRELT (2010)
59. 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, pp. 422–437 (2014)
60. Agra, A., Christiansen, M., Figueiredo, R., Hvattum, L.M., Poss, M., Requejo, C.: The robust vehicle routing problem with time windows. Comput. Operat. Res. **40**(3), 856–866 (2013)
61. Azi, N., Gendreau, M., Potvin, J.Y.: An exact algorithm for a single-vehicle routing problem with time windows and multiple routes. Eur. J. Operat. Res. **178**(3), 755–766 (2007)
62. Bräysy, O., Gendreau, M.: Tabu search heuristics for the vehicle routing problem with time windows. Top **10**(2), 211–237 (2002)
63. Cordeau, J.F., Desaulniers, G., Desrosiers, J., Solomon, M.M., Soumis, F.: Vrp with time windows. Vehicle Rout. Prob. **9**, 157–193 (2001)
64. Glover, F.: Ejection chains, reference structures and alternating path methods for traveling salesman problems. Discr. Appl. Mathemat. **65**(1–3), 223–253 (1996)
65. Osaba, E., Yang, X.S., Diaz, F., Onieva, E., Masegosa, A.D., Perallos, A.: A discrete firefly algorithm to solve a rich vehicle routing problem modelling a newspaper distribution system with recycling policy. Soft Comput. 1–14 (2016)
66. Irnich, S.: A unified modeling and solution framework for vehicle routing and local search-based metaheuristics. INFORMS J. Comput. **20**(2), 270–287 (2008)
67. Campbell, A.M., Savelsbergh, M.: Efficient Insertion heuristics for vehicle routing and scheduling problems. Transport. Sci. **38**(3), 369–378 (2004)
68. Solomon, M.M.: Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper. Res. **35**(2), 254–265 (1987)
69. 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 Evol. Computat. **1**(1), 3–18 (2011)