

# 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** In this paper an evolutionary and discrete variant of the Bat Algorithm (EDBA) is proposed for the addressing of the Vehicle Routing Problem with Time Windows, or VRPTW. The EDBA developed not only presents an improved movement strategy, but it also counts with diverse heuristic operators to deal with this complex problem. One of the main concepts of these functions is to unify the search process and the minimization of the routes and total distance in the same operators. This hybridization is reached using node selective extractions and subsequent reinsertions. Additionally, the used functions analyze 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 and operators. The experimentation has been done with the 56 instances that compose the 100 customers Solomon's benchmark. Finally, two statistical tests have been carried out in order to obtain rigorous conclusions.

---

Eneko Osaba, Roberto Carballedo, Pedro Lopez-Garcia  
Deusto Institute of Technology (DeustoTech), University of Deusto, Av. Universidades 24, Bilbao 48007, Spain e-mail: [e.osaba, roberto.carballedo, p.lopez]@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

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

Javier Del Ser  
TECNALIA, 48160 Derio, Spain  
University of the Basque Country (UPV/EHU), 48013 Bilbao, Spain  
Basque Center for Applied Mathematics (BCAM), 48009 Bilbao, Spain  
e-mail: javier.dels@tecnalia.com

**Key words:** 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 increasingly important in nowadays society. Today, anyone in the world can be easily connected, and this fact has led transport networks to be very demanding, something less important in the past. In this way, a competitive logistic network can make the difference between some companies and others. Thanks to this importance, the development of efficient methods for a proper logistic and routing planning is a hot topic in the research community.

In this regard, some areas of knowledge can address this sort of issues in an appropriate way. In this case, we focus our attention in one of these areas: artificial intelligence. In this way, route planning problems and their resolution is one of the most recurrent topics inside artificial intelligence. More specifically, the problems arisen in this field are normally named as routing problems, and they fall into the combinatorial optimization category. In this sense, 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 paper, the attention is focused on one of these variants: the Vehicle Routing Problem with Time Windows, or VRPTW. Briefly explained, in the VRPTW each client imposes a time window for the start and the end of the service. This problem will be further detailed in future sections.

Furthermore, several effective methods can be found in the literature to face this kind of problems properly. Arguable, the most famous approaches for this purpose are the exact methods [1], heuristics and metaheuristics. We center our attention on the last ones. In line with this, some classical examples of local search-based methods are Simulated Annealing [2] and Tabu Search [3]. On the other hand, as population-based techniques the Ant Colony Optimization [4], Genetic Algorithms (GA) [5, 6], and Particle Swarm Optimization [7] are some of the most used alternatives.

Although the existence of classical techniques, the development of novel metaheuristics for tackling optimization problems, in general, and routing problems, in particular, is a hot topic for the current research community. In this way, many different methods have been proposed over the last years, showing a great performance. Some examples of these methods are the Imperialist Competitive Algorithm, proposed by Atashpaz-Gargari and Lucas in 2007 [8], or the Artificial Bee Colony, presented by Karaboga and Basturk in 2007 [9]. To the interested readers, some additional successful methods will be described in following sections.

In this case, 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]. As can be analyzed in some studies of the 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] confirms that BA still attracts a lot of interest from the scientific community. In this sense, despite the BA has been applied to many different optimization problems up to date, it has not been applied yet to the well known VRPTW. As a conclusion, this lack of works and the growing interest in the BA by the scientific community has motivated this work. Additionally, further details on the BA will be given in following sections.

It is also interesting to highlight that we have used some novel route optimization operators for enhancing the performance of the developed algorithm. These operators, which will be deeply 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 a BA.

Furthermore, and with the intention of proving that the implemented Evolutionary Discrete Bat Algorithm (EDBA) is a promising approach to solve the VRPTW, an experimentation composed by 56 different instances has been conducted. In this experimentation, the results obtained by some variants of the EDBA are compared. In addition, in order to extract valid conclusions, 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.

The rest of the paper is organized as follows. In the following section (Section 2) the related background is presented, which is focused in routing problems and nature-inspired metaheuristics for their resolution. After that, in Section 3 the philosophy of the basic BA is deeply detailed. Then, in Section 4 a brief description of the VRPTW can be found. We describe our proposed EDBA and our developed route optimization operators in Section 5. Furthermore, in Section 6, the experimentation performed for the validation of the study is detailed. Finally, the paper finishes with the conclusions and further work in Section 7.

## 2 Background

Nowadays, route planning is one of the most studied fields in artificial intelligence. Problems arisen in this field are usually known as vehicle routing problems, which are a particular case of problems within combinatorial optimization subject. In this sense, it is noteworthy that 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]. Additionally, the VRPTW, which is the problem in which we focus our attention in this work is also

one of the most cited and used ones, 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. For this reason, their resolution supposes 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. This is the main reason because their resolution entails a profit, either business or social one.

Furthermore, several approaches can be found in the scientific community to tackle with this kind of problems in an efficient way. In this sense, the most successful method are the exact methods [22, 1], heuristics and metaheuristics. Specifically, as can be seen in the work published by Braysy and Gendreau in 2005 [23], metaheuristics are an excellent approach for solving the VRPTW.

Being more specific, within the category of metaheuristics, nature-inspired methods are some of 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. As can be read in several surveys [11, 12], 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.

More recent is the work proposed in [26] by Zhou et al., in which the same Capacitated VRP is faced. In that 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 certain probability.

Regarding the second of the method above mentioned, the FA, it was proposed by Yang in 2008 [27]. This nature-inspired algorithm is based on the flashing behavior of fireflies, which acts as a signal system to attract other fireflies. This meta-heuristic has been also applied to a wide range of optimization fields and problems since its proposal [28, 29]. As 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, authors adapt the FA, which was firstly proposed for tackling continuous problems, providing 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 is a Gaussian Firefly Algorithm. The most interesting part of this paper is the real-world use case that authors present, focused on a distribution company, established in Esfahan, Iran. More recently, FA has been compared to 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 this 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, the CS, was firstly 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). Despite it was firstly developed for addressing other kind of problems, the CS has also been adapted to routing problems, as can be seen, for example, in the work published in 2014 by Ouaarab et al. [34]. In this paper, authors present the first adaptation of the CS to the well-known TSP, creating a discrete variant of the CS which shows really promising results. In order to prove that, authors tested their proposed discrete CS against a set of benchmarks of symmetric TSP from the well-known TSPLIB library.

Many 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 reputed Information Sciences journal published a paper in which four different soft computing methods are applied for solving also the Capacitated VRP [36]. One of these approaches is an advanced CS, which introduces new adjustments and features for improving its efficiency.

An additional example is the paper presented by Chen and Wang in 2016 [37], in which a hybrid CS is proposed for the solving the VRP in logistics distribution systems. This novel algorithm is based on the combination of Optical Optimization, Particle Swarm Optimization and CS. Specifically, in this method, optical optimization is introduced to initialize population for obtaining a group of initial values with high quality, which are optimized according to PSO. After each iterative operation for keeping the optimal individual, CS is used to optimize the rest of the individuals.

Another influential meta-heuristic which can be framed within the nature-inspired (or music-inspired in this case) category is the Harmony Search (HS). This technique firstly proposed by Geem et al. in 2001 is also a phenomenon-mimicking metaheuristic [38]. Specifically, HS is which is inspired by the improvisation process of jazz musicians. The success of this method can be confirmed with the huge number of papers focused on HS that can be found in the literature [39], [40], [41].

The HS has also been applied to routing problems several times, showing also a promising performance. In fact, the state of the art paper presented by Geem, Lee and Park in 2005 collects some of the most interesting works up to this date in this topic [42]. Another research related with the HS is the one that can be found in [43], which presents a discrete variant of the HS in order to face the challenging the Selective Pick-Up and Delivery VRP with Delayed Drop-Off. Additionally, in the recent work published by Bounzidi and Riffi in 2014 describes and adaptation of the HS for solving the TSP.

The last meta-heuristic mentioned in this background section is the Gravitational Search algorithm (GS). This method was proposed by Rashedi et al. in 2009 [44], and it is based on the metaphor of gravitational interaction between masses. As the rest of the approaches described before, the GS is a successful technique applied to a wide range of problems since its proposal [45, 46, 47]. Centering our attention in routing problems, in the work published by Nodehi et al. in 2016 [48], a randomized GS algorithm is proposed for the solving of the TSP. The GS implement in this work is 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 is compared with some additional well-known methods, such as the Genetic Algorithm, showing a promising performance.

Regarding VRP problems, an interesting work can be found in [49]. This research, published in 2016, explores 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 this paper is 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 published by Hosseinabad et al. in 2017 [50] presents another approach of the GS to solve the Capacitated VRP. Authors of this study propose what they called a Gravitational Emulation Local Search Algorithm, which is a hybrid variant of the GS and a local search. The main objective of this hybridization is to take advantage of the exploration capacity of the GS, and the exploitation capacity of the local search. As can be confirmed in this paper, the Gravitational Emulation Local Search Algorithm show a great performance for the Capacitated VRP, outperforming some successful state of the art methods.

As has been mentioned before, the problem faced in this paper in the well known VRPTW. In this regard, the number of publications related to this problem is really high. For this reason, some recently published works are going to be described. In [51], for example, 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 [52] 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. [53]. Finally, an interesting 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 is proposed for the resolution of this problem, using the same operators that are used in the experimentation that is described in the present paper. An additional interesting paper is the one presented by [55], in which a hybrid variant of the HS is presented to deal with the VRPTW. This work, published by Information Sciences journal in 2015 by Yassen et al. explores the hybridization of the HS with a local search algorithm (LS). This work states that the success of this hybridization relies on the achievement of a proper balance between HS exploration and LS exploitation. To reach this balance, this research proposes a meta-harmony search algorithm which uses two different HS algorithms, an HS optimizer and HS solver. The first one adaptively adjusts the components and the configurations of the solver based on the search status. On the other hand, the solver, which is the pure hybridization of HSA and LS, takes the configuration generated by the HSA-optimizer as input and solves the given VRP with time windows problem instance.

Finally, it is worth to point 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, arguably, 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 literature review paper presented 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 solving methods, the work presented in [57, 58] is highly recommended. Additionally, after the revision of the state of the art of both fields, we want to highlight that this is the first time in the literature that the BA is applied to the VRPTW. In this sense, as has been mentioned in the introduction of this work, this is one of the main motivations that has pushed us to perform this research.

### 3 Bat Algorithm

In this section, the basic variant of the BA is fully described, in order to enhance the understandability of the research. As we have briefly mentioned in previous sections, the BA is a nature-inspired metaheuristic, which main idea is to imitate this echolocation system of the bats. In order to facilitate the execution of the algorithm, some idealized rules should be taken into account [10]:

- All bats use echolocation to detect the distance, and they have a called “magic ability”, which allows them to difference between an obstacle and a prey (obstacles and preys refer to bad and good solutions respectively).
- All bats fly randomly with a velocity  $v_i$  at position  $x_i$  with a fixed frequency  $f_{min}$ , varying wavelength  $\lambda$  and loudness  $A_i$  to search a prey. For this second rule, it is assumed that every bat can adjust the frequency (or wavelength) of their emitter pulses in an automatic way. Additionally, the rate of these pulses emission is

**Algorithm 1:** Pseudo code of the basic BA

```

1 Define the objective function  $f(x)$ ;
2 Initialize the bat population  $X = x_1, x_2, \dots, x_n$ ;
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  $\text{rand} > r_i$  then
11      Select one solution among the best ones;
12      Generate a local solution around the best one;
13    end
14    if  $\text{rand} < A_i$  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;

```

$r \in [0, 1]$ . This automatic adjustment depends on the proximity of the targeted prey.

- In the real-world, the loudness of bats emissions can vary in many different ways. Anyway, for this third rule, it is assumed that this loudness can vary from a minimum constant value  $A_{min}$  to a large positive  $A_0$ .

Additionally, we show in Algorithm 1 the pseudo-code of the basic BA. Taking a quick look to 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. In this sense, it is important to mention that every 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 steps of the algorithm are executed. In this sense, and for each generation, every bat of the swarm moves through the solution space by updating its velocity and position. More specifically, the following equations are used for this movement:

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

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

$$x_i^t = x_i^{t-1} + v_i^t \quad (3)$$

where  $\beta$  is a randomly generated number in the  $[0, 1]$  interval. Beside that,  $x_*$  represents the current best solution of the whole population, and  $v_i^t$  and  $x_i^t$  denote the



velocity and position of a bat  $i$  at time step  $t$ . Furthermore, the results of Equation (1) is used to control the pace and range of bats movement. In addition, if a solution is selected among the best ones, a new solution for each bat is generated using a random walk

$$x_{new} = x_{old} + \varepsilon A^t \quad (4)$$

where  $\varepsilon$  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:

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

$$A_i^{t+1} = \alpha A_i^t \quad (6)$$

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

$$A_i^t \rightarrow 0, r_i^t \rightarrow r_i^0, \text{ as } t \rightarrow \infty \quad (7)$$

In many researches of 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 experimentation using a  $[0.90, 0.99]$  range.

## 4 Vehicle Routing Problem with Time Windows

As we have pointed in Section 2, the VRPTW is an extension of the classic and widely studied VRP. In this way, in addition to the basic constraints inherent to 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  $l_i$ .

In this way, a route would not be 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 VRP, this problem has been widely studied both in the past [59, 60], and nowadays [61, 62]. 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 success is its easy adaptation to the real world. The great majority of distribution chains, customers have strong temporal constraints that have to be fulfilled. This is the reason because 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 [63, 64]. One of the most interesting formulations can be found in [65].

## 5 Our proposed approach for solving the VRPTW

In this section, the description of our EDDBA for the VRPTW is provided (Section 5.1). Besides that, a depth description of the proposed novel route optimization operator can be read in Section 5.2.

### 5.1 An Evolutionary Discrete Bat Algorithm

Before starting with the deep description of our proposed method, it is worth mentioning that the original BA was firstly developed for solving continuous optimization problems. This is the reason because the classic variant of this metaheuristic cannot be directly applied to solve any discrete problem, such as the one in which we focus our study, the VRPTW. Hence, some modifications in the structure of the basic BA should be performed in order to prepare it to face the VRPTW.

First, in the developed EDDBA, each bat of the swarm represents a possible and feasible solution for the VRPTW. In addition, being the VRPTW 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. 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 Equation (2). Anyway, this formula cannot be used in the same way for solving a discrete problem such as the VRPTW. In this way, with the intention of finding an adaptation as much accurate as possible, we have related  $v_i$  with the distance between the bat  $i$  and the best bat

of the swarm. For this purpose, we have adapted  $v_i$  using the well-known Hamming Distance in the following way:

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

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 Equation 3. As same as happens with the  $v_i$  parameters, this equation cannot be applied directly to a discrete problem such as the VRPTW. For this purpose, a modification has been proposed. Thereby, the movement of a bat  $i$  is determined by the following equation:

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

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  $\text{rand} > r_i$ , 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.

Finally, and 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).

Taking as inspiration the concept of “ejection chains” [66], 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 [67]. 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”).

**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:  $Solution_{current}$ 

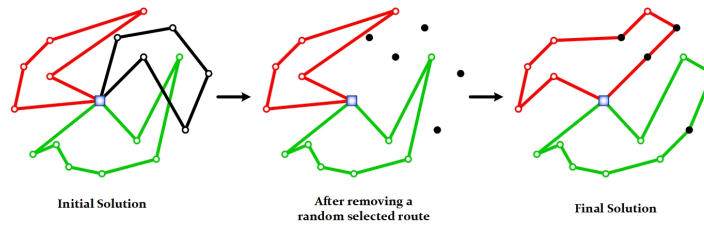
```

The 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 [67], in the present work we center our attention only in 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 an example of the RrE-opt operator. Furthermore, Algorithm 2 shows the description of that 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 process 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 [68].



**Fig. 1** Example of the RrE-opt operator.

- 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 [69] 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 pose 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 four variants of the proposed EDDBA will be compared. These variants will allow the evaluation of the two optional processes of the operator of 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 EDDBA 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 [70]. This set of problems consists of a 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<sup>1</sup>), but the objective of the work presented focuses on the use of the EDDBA and the analysis of the new optimization operator for the VRPTW. For this reason, Solomon's benchmark is adequate and representative.

<sup>1</sup> <https://www.sintef.no/projectweb/top/vrptw/homberger-benchmark/>

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 programmed 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 section 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 by 25 individuals.
- The initial population is initialized at random.
- The termination criterion used is the following: a maximum of 100 iterations or 20 iterations without improvement.
- New solutions are generated with the optimization operator described in section 5.2.
- The local solution around the best new solution is generated using the well-known 2-opt\* operator.
- $\alpha$  and  $\beta$  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.

Finally, in order to obtain better results each variant of the EDBA has been executed 10 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 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 removed route. According to the experimentation conducted this variant obtain the best results (both in vehicles and traveled distance) for all the classes except C1. For the Class C1 this variant obtains the best results in terms of distance and the number of vehicles is only a 0,1% worse than the best one. The results obtained are consistent since the standard deviation for both vehicles and for distance does not

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 1** Results obtained by EDBA-OR.

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 2** Results obtained by EDBA-PR.

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

**Table 3** Results obtained by EDBA-FULL.

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. In this case, 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 difference in number of vehicles and total distance traveled (for all classes) is 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.

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

**Table 4** Results obtained by EDDBA-NONE.

	EDDBA-OR				EDDBA-FULL				EDDBA-PR				EDDBA-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	<b>1512.744</b>	0.0	<b>10.967</b>	0.0	1531.749	1.3	12.889	17.5	2270.032	50.1	12.867	17.3	2270.032	50.1
C2	<b>3.200</b>	0.0	<b>779.528</b>	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	<b>14.367</b>	0.0	<b>1592.738</b>	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	<b>3.164</b>	0.0	<b>1211.082</b>	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	<b>14.925</b>	0.0	<b>1915.000</b>	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	<b>3.750</b>	0.0	<b>1467.878</b>	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 5** Summary of the results and comparison between all the methods.

Finally, Table 4 shows the results of EDDBA-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 EDDBA-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 EDDBA-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, the Table 5 shows the comparison of all variants and the difference with respect to the EDDBA-OR (which reported the best results).

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. [71]. The objective of this task is to ensure that comparisons between the different variants of the EDDBA 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 EDDBA.

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

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

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  $\chi^2$  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 EDDBA-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 EDDBA-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 EDDBA-OR variant.

Algorithm	adjusted p	unadjusted p
EDDBA-PR	0.005235	0.001745
EDDBA-NONE	0.005235	0.001745
EDDBA-FULL	0.371093	0.371093

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

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 EDDBA-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 EDDBA-PR and EDDBA-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 EDDBA-PR and EDDBA-NONE with respect to EDDBA-OR is significant. The same does not happen between the EDDBA-FULL and EDDBA-OR variants.

To conclude the 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 EDDBA-PR and EDDBA-NONE with respect to EDDBA-OR.

Algorithm	adjusted p	unadjusted p
EDDBA-PR	0.002389	0.000796
EDDBA-NONE	0.002389	0.000796
EDDBA-FULL	0.179712	0.179712

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

Finally, as a conclusion of the experimentation and the subsequent statistical analysis of the results, it can be ensured that the EDDBA-OR variant is the one that obtains the best results. These results are statistically better than those obtained by the EDDBA-PR and EDDBA-NONE variants. On the contrary, the results obtained by the EDDBA-FULL variant are worse than those obtained by EDDBA-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 of the swarm, or 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 proving that the developed 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 extracted conclusions: the non-parametric Friedmans test for multiple comparisons, and the post-hoc Holm's test.

The future work related with the research presented in this paper is quite wide, and it includes the comparison of the performance of the proposed EBFA with the one presented by some recently proposed metaheuristics. Additionally, we have planned to use the route optimization heuristic operators described in this work in other recent and classic techniques, such as the Genetic Algorithm or the Tabu Search, in order to test their efficiency.

## Acknowledgements

## References

1. Laporte, G.: The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research* **59**(3) (1992) 345–358
2. Kirkpatrick, S., Gellat, C., Vecchi, M.: Optimization by simulated annealing. *science* **220**(4598) (1983) 671–680
3. Glover, F.: Tabu search, part i. *ORSA Journal on computing* **1**(3) (1989) 190–206
4. Dorigo, M., Blum, C.: Ant colony optimization theory: A survey. *Theoretical computer science* **344**(2) (2005) 243–278
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*. Volume 4., Perth, Australia (1995) 1942–1948
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 (2007) 4661–4667*
9. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of global optimization* **39**(3) (2007) 459–471

10. Yang, X.S.: A new metaheuristic bat-inspired algorithm. In: Nature inspired cooperative strategies for optimization. Springer (2010) 65–74
11. Yang, X.S., He, X.: Bat algorithm: literature review and applications. *International Journal of Bio-Inspired Computation* **5**(3) (2013) 141–149
12. Parpinelli, R.S., Lopes, H.S.: New inspirations in swarm intelligence: a survey. *International Journal of Bio-Inspired Computation* **3**(1) (2011) 1–16
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 (2017) 3–7
14. Tharakeshwar, T., Seetharamu, K., Prasad, B.D.: Multi-objective optimization using bat algorithm for shell and tube heat exchangers. *Applied Thermal Engineering* **110** (2017) 1029–1038
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 (2016) 21–41
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-Operations Research-Recherche Opérationnelle* **10**(V1) (1976) 55–70
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. *Computers & Operations Research* **78** (2017) 454–467
19. Veenstra, M., Roodbergen, K.J., Vis, I.F., Coelho, L.C.: The pickup and delivery traveling salesman problem with handling costs. *European Journal of Operational Research* **257**(1) (2017) 118–132
20. 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
21. 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
22. Laporte, G.: The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research* **59**(2) (1992) 231–247
23. Bräysy, O., Gendreau, M.: Vehicle routing problem with time windows, part ii: Metaheuristics. *Transportation science* **39**(1) (2005) 119–139
24. Yang, X.S.: Nature-inspired metaheuristic algorithms. Luniver press (2010)
25. Taha, A., Hachimi, M., Moudén, A.: Adapted bat algorithm for capacitated vehicle routing problem. *International Review on Computers and Software (IRECOS)* **10**(6) (2015) 610–619
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 (2016) 255–276
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* **13** (2013) 34–46
30. Jati, G.K., Suyanto. In: Evolutionary Discrete Firefly Algorithm for Travelling Salesman Problem. Springer Berlin Heidelberg (2011) 393–403
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. *Computers & Industrial Engineering* **99** (2016) 210–222
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. 2017 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 (2009) 210–214

34. Ouabarab, A., Ahiod, B., Yang, X.S.: Discrete cuckoo search algorithm for the travelling salesman problem. *Neural Computing and Applications* **24**(7-8) (2014) 1659–1669
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 (2016) 71–79
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. *Information Sciences* **334** (2016) 354–378
37. Chen, X., Wang, J.: A novel hybrid cuckoo search algorithm for optimizing vehicle routing problem in logistics distribution system. *Journal of Computational and Theoretical Nanoscience* **13**(1) (2016) 114–119
38. Geem, Z.W., Kim, J.H., Loganathan, G.: A new heuristic optimization algorithm: harmony search. *Simulation* **76**(2) (2001) 60–68
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. *Engineering Applications of Artificial Intelligence* **26**(8) (2013) 1818–1831
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 (2016) 863–874
41. Mohd Alia, O., Mandava, R.: The variants of the harmony search algorithm: an overview. *Artificial Intelligence Review* **36**(1) (2011) 49–68
42. Geem, Z.W., Lee, K.S., Park, Y.: Application of harmony search to vehicle routing. *American Journal of Applied Sciences* **2**(12) (2005) 1552–1557
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 (2016) 121–131
44. Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S.: Gsa: a gravitational search algorithm. *Information sciences* **179**(13) (2009) 2232–2248
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 transactions on cybernetics* **44**(11) (2014) 1997–2009
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 Theory & Applications* **7**(1) (2013) 99–107
47. Duman, S., Güvenç, U., Sönmez, Y., Yörükeren, N.: Optimal power flow using gravitational search algorithm. *Energy Conversion and Management* **59** (2012) 86–95
48. Nodehi, A.N., Fadaei, M., Ebrahimi, P.: Solving the traveling salesman problem using randomized gravitational emulation search algorithm. *Journal of Current Research in Science* (2) (2016) 818
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 (2016) 93–103
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. *Applied Mathematical Modelling* (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. *Computers & Operations Research* **52** (2014) 269–281
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. *Journal of Traffic and Logistics Engineering Vol* **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. *Journal of Heuristics* **20**(4) (2014) 383–416
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. *Information Sciences* **325** (2015) 140–158
56. Yang, X.S., Deb, S.: Cuckoo search: recent advances and applications. *Neural Computing and Applications* **24**(1) (2014) 169–174
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. 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
60. 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
61. 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
62. 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
63. 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
64. Bräysy, O., Gendreau, M.: Tabu search heuristics for the vehicle routing problem with time windows. *Top* **10**(2) (2002) 211–237
65. Cordeau, J.F., Desaulniers, G., Desrosiers, J., Solomon, M.M., Soumis, F.: Vrp with time windows. *The vehicle routing problem* **9** (2001) 157–193
66. Glover, F.: Ejection chains, reference structures and alternating path methods for traveling salesman problems. *Discrete Applied Mathematics* **65**(1-3) (1996) 223–253
67. Osaba, E., Yang, X.S., Diaz, F., Onieva, E., Masegosa, A.D., Perillos, A.: A discrete firefly algorithm to solve a rich vehicle routing problem modelling a newspaper distribution system with recycling policy. *Soft Computing* (2016) 1–14
68. 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
69. Campbell, A.M., Savelsbergh, M.: Efficient Insertion Heuristics for Vehicle Routing and Scheduling Problems. *Transportation Science* **38**(3) (August 2004) 369–378
70. Solomon, M.M.: Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research* **35**(2) (1987) 254–265
71. 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