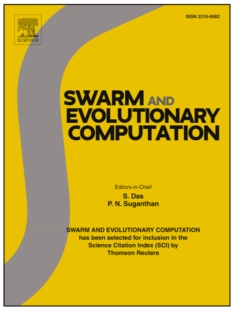
Accepted Manuscript



A Discrete and Improved Bat Algorithm for solving a medical goods distribution problem with pharmacological waste collection

Eneko Osaba, Xin-She Yang, Iztok Fister, Jr., Javier Del Ser, Pedro Lopez-Garcia, Alejo J. Vazquez-Pardavila

PII: S2210-6502(17)30763-0

DOI: 10.1016/j.swevo.2018.04.001

Reference: SWEVO 391

To appear in: *Swarm and Evolutionary Computation BASE DATA*

Received Date: 17 September 2017

Revised Date: 15 March 2018

Accepted Date: 6 April 2018

Please cite this article as: E. Osaba, X.-S. Yang, I. Fister Jr., J. Del Ser, P. Lopez-Garcia, A.J. Vazquez-Pardavila, A Discrete and Improved Bat Algorithm for solving a medical goods distribution problem with pharmacological waste collection, *Swarm and Evolutionary Computation BASE DATA* (2018), doi: 10.1016/j.swevo.2018.04.001.

A Discrete and Improved Bat Algorithm for solving a Medical Goods Distribution Problem with Pharmacological Waste Collection

Eneko Osaba[a,d,*], Xin-She Yang[b], Iztok Fister Jr.[c], Javier Del Ser[a,e,f], Pedro Lopez-Garcia[d], Alejo J. Vazquez-Pardavila[d]

[a] *TECNALIA, 48160 Derio, Spain*
[b] *School of Science and Technology, Middlesex University, Hendon Campus, London, NW4 4BT, United Kingdom*
[c] *Faculty of Electrical Engineering and Computer Science, University of Maribor, Smetanova 17, 2000 Maribor, Slovenia*
[d] *Deusto Institute of Technology (DeustoTech), University of Deusto, Av. Universidades 24, Bilbao 48007, Spain*
[e] *University of the Basque Country (UPV/EHU), 48013 Bilbao, Spain*
[f] *Basque Center for Applied Mathematics (BCAM), 48009 Bilbao, Spain*

Abstract

The work presented in this paper is focused on the resolution of a real-world drugs distribution problem with pharmacological waste collection. With the aim of properly meeting all the real-world restrictions that comprise this complex problem, we have modeled it as a multi-attribute or rich vehicle routing problem (RVRP). The problem has been modeled as a Clustered Vehicle Routing Problem with Pickups and Deliveries, Asymmetric Variable Costs, Forbidden Roads and Cost Constraints. To the best of authors knowledge, this is the first time that such a RVRP problem is tackled in the literature. For this reason, a benchmark composed of 24 datasets, from 60 to 1000 customers, has also been designed. For the developing of this benchmark, we have used real geographical positions located in Bizkaia, Spain. Furthermore, for the proper dealing of the proposed RVRP, we have developed a Discrete and Improved Bat Algorithm (DaIBA). The main feature of this adaptation is the use of the well-known Hamming Distance to calculate the differences between the bats. An effective improvement has been also contemplated for the proposed DaIBA, which consists on the existence of two different neighborhood structures, which are explored depending on the bat's distance regarding the best individual of the swarm. For the experimentation, we have compared the performance of our presented DaIBA with three additional approaches: an evolutionary algorithm, an evolutionary simulated annealing and a firefly algorithm. Additionally, with the intention of obtaining rigorous conclusions, two different statistical tests have been conducted: the Friedman's non-parametric test and the Holm's post-hoc test. Furthermore, an additional experimentation has been performed in terms of convergence. Finally, the obtained outcomes conclude that the proposed DaIBA is a promising technique for addressing the designed problem.

Keywords: Bat Algorithm, Medical Distribution, Rich Vehicle Routing Problem, Combinatorial Optimization, Traveling Salesman Problem

1. Introduction

Transportation and logistics are important issues for the society these days, both for citizens and the business sector. We are perfectly aware that public transportation is used by almost all the population, and that it directly affects the people quality of life. In addition, business logistics can also be considered as transportation problem, which requires optimization techniques to solve. Therefore, this paper will focus on the logistic problems concerning medical device distribution and pharmacological waste collection.

In the business world, the fast advance of technology has made the logistic increasingly important in this area. Additionally, anyone in the whole world can be well connected. This situation has led transport networks to be very demanding, something that was less important in the past. Nowadays, a competitive logistic network can make the difference between some companies, and can crucially contribute to their success.

This work is focused on the proper modeling and treatment of a real-world logistic problem. Specifically, the real-world situation tackled in this paper is related to the distribution of medical goods. In this case, we center our attention in a SME[1] medical distribution enterprise, with regional influence. This company has an established logistic philosophy, which needs to be followed when they perform the daily distribution planning. All the characteristics that integrate this philosophy are explained in the following section. Finally, despite the object of this study is a company physically placed on Bizkaia (Spain), the main objective of this study is to propose a model which can be applied to every similar company.

Hence, the main objective of this work is to tackle efficiently this Drugs Distribution System with Pharmacological Waste Collection (DDSPWC). For reaching this goal properly, we have modeled the DDSPWC as a Rich Vehicle Routing Problem (RVRP). Currently, this type of complex problems is catching the attention of the scientific community, as can be read in several works, such as (Caceres-Cruz et al. (2015)) or (Doerner & Schmid (2010)). As we can be found in these surveys, RVRPs are special cases of the conventional Vehicle Routing Problem (VRP) (Golden et al. (2008)). These special cases are characterized for having multiple variables and constraints, and a complex formulation.

The principal reasons for the importance and popularity of these problems are twofold: the social interest they generate, and their inherent scientific interest. Firstly, RVRPs are usually designed for dealing with a specific real-world situation related to transport or logistics. This is the reason why their efficient resolution entails a profit, either business or social one. Secondly, most of RVRPs have a great computational complexity, and their resolution represents a major challenge for the scientific community.

*Corresponding author

Email addresses: e.osaba@deusto.es,eneko.osaba@tecnalia.com (Eneko Osaba), x.yang@mdx.ac.uk (Xin-She Yang), iztok.fister1@um.si (Iztok Fister Jr.), javier.delser@tecnalia.com (Javier Del Ser), p.lopez@deusto.es (Pedro Lopez-Garcia), alejo.vazquez@deusto.es (Alejo J. Vazquez-Pardavila)

[1]SME: Small and medium-sized enterprise

Specifically, we present in this paper a Clustered Vehicle Routing Problem with Pickups and Deliveries, Asymmetric Variable Costs, Forbidden Roads and Cost Constraints (C-VRP-P*C) to tackle the proposed DDSPWC. As has been mentioned, RVRPs have caught the attention of the current community. In this sense, (Sicilia et al. (2016)) and (de Armas & Melián-Batista (2015)) are two examples of recently published RVRPs. The first of these works is related to the capillary transport of goods problem. The research project presented in that work was carried out for an important Spanish distribution company, and its main goal is to manage their resources in urban areas by reducing costs caused by inefficiency and ineffectiveness. The RVRP considered in that study comprises some constraints such as pick up and deliveries, backhauls, site-dependence, time-windows, capacities and openness. Authors proposed two different methods for its resolution: a variable neighborhood search (VSN) and a tabu search (TS). The second of the mentioned works presents also a VNS for the resolution of a dynamic RVRP. In that case, several real constraints have been considered, such as heterogeneous fleet of vehicles, multiple and soft time windows and customers priorities. Furthermore, it is worth mentioning that the software developed in that work has been incorporated into the fleet management system of a company in Spain. An additional example of recently developed RVRP is the one proposed in (Amorim & Almada-Lobo (2014)). In this paper, the authors present an RVRP to deal with the perishable food management. The RVRP designed in this case is a heterogeneous fleet site-dependent VRP with multiple time windows.

Furthermore, in 2016, Mancini presented in (Mancini (2016)) an interesting RVRP with multiple periods, multiple depots and heterogeneous fleet is presented. For tackling this challenging problem, the author developed an adaptive VNS based approach. The experimentation performed in that paper addresses 9 different datasets composed by 50 and 75 customers, and highlight the quality of the presented method. Besides that, in (Belmecheri et al. (2013)), Belchemeri el al. developed a particle swarm optimization (PSO) algorithm for solving a real-world based RVRP with heterogeneous fleet, time windows and mixed backhauls. In that paper, the results obtained by the presented approach are compared with a basic local search, and an ant colony optimization (ACO). For the experimentation, an ad-hoc modification of the well-known Solomon VRPTW Benchmark is used, with instances composed of 100 nodes. Finally, another interesting example was presented in (Penna et al. (2016)), in which an electric fleet size and mix VRP was designed, with recharging stations and time windows. For solving this novel problem, a hybrid iterative local search was implemented.

There are several appropriate approaches to deal with such complex optimization problems. Anyway, the most successful techniques to address the resolution of RVRP are heuristics and metaheuristics. In this paper, our attention focuses on the second of these categories: metaheuristics. In line with this, we propose a nature-inspired metaheuristic for the resolution of the designed C-VRP-P*C.

Lots of metaheuristics have been presented in the literature along the years (Fister Jr et al. (2013)). The implementation of new and classical methods, and their proper application still forms a hot topic in the scientific community (Precup et al. (2014); Mahdavi et al. (2015); Wari & Zhu (2016); Precup et al. (2015)). In fact, many novel approaches have been presented in the last decade, such as the Firefly Algorithm, proposed by Yang (Yang (2010a)), Charged System Search, presented by Kaveh and Talatahari in 2010 (Kaveh & Talatahari (2010)), or the Spider Monkey Optimization, proposed by Bansal et al. in 2014 (Bansal et al. (2014)). Another kind of methods that have demonstrated a good performance applied to RVRPs are the memetic algorithms (Moscato et al. (1989)). Some examples of this good performance are (Bortfeldt et al. (2015)), in which a RVRP with clustered backhauls and 3D loading constraints is tackled, or (Zhang et al. (2013)), where a multiperiod VRP with profit is addressed. Additional works can be found in (Gutierrez et al. (2016); Rogdakis et al. (2017)).

This way, we have highlighted some methods which have been already used in the literature for solving RVRP problems: VNS, TS, PSO, ACO, local search methods, and memetic algorithms. Additional approaches can be found in the current literature to properly addressing this kind of problems, such as the genetic algorithm (Shi et al. (2017)), or the simulated annealing (Wang et al. (2017)). The Large Neighborhood Search has also been recently used in the literature for solving a RVRP (Talarico et al. (2017)). As can be logical, each of these methods have their advantages and disadvantages. In this specific paper, and with the aim of properly addressing the designed RVRP, we propose a nature-inspired metaheuristic based on the Bat Algorithm (BA). The BA was firstly presented by Yang in 2010 (Yang (2010b)), 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 has been highlighted in several studies, such as (Yang & He (2013)) or (Parpinelli & Lopes (2011)), the BA has been applied to different optimization fields and problems up to now. Furthermore, the fact that many research works focused on BA are being currently published proves that this approach is still interesting for the researchers, in different areas such as the continuous optimization (Chakri et al. (2017)), or the thermal engineering (Tharakeshwar et al. (2017)). Furthermore, the algorithm itself is also the focus of recent research, such as the works presented in (Perez et al. (2017a)) and (Perez et al. (2017b)), in which the parameter adaptation of the algorithm is studied.

Focusing on routing problems, several recently published papers have shown that the BA is a promising technique also in this field. For example, in (Taha et al. (2015)), 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. Furthermore, in 2017, the same authors presented in (Taha et al. (2017)) an adaptation of the same technique for solving the well-known VRP with Time-Windows. Another interesting work is the one work proposed in (Zhou et al. (2016)) by Zhou et al., in which the 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.

Besides that, the BA has also been applied to the famous Traveling Salesman Problem several times in recent years. In (Osaba et al. (2016b)), Osaba et al. presented an improved adaptation of the BA for addressing both symmetric and asymmetric TSP. The results show that the improved version of BA could obtain promising results, in comparison with some reference techniques, such as an evolutionary simulated annealing, a genetic algorithm, a distributed genetic algorithm or an imperialist competitive algorithm. An additional example of this specific application is the one presented by Saji and Riffi in 2106 (Saji & Riffi (2016)). In that work, the performance of their discrete version of the BA is compared with three different meta-heuristics: a discrete particle swarm optimization (PSO) (Chen & Chien (2011)), a genetic simulated annealing ant colony system with PSO techniques and a discrete cuckoo search (Ouaarab et al. (2014)).

Nevertheless, despite this interest, the BA has never been applied before to any kind of RVRP. This lack of works is one of the motivations behind using the BA for our study. There are additional reasons for the choosing of this technique, such as the growing scientific interest shown by the community in recent years, or the proper balance between exploration and exploitation shown by the technique for solving complex problems. Anyway, and most importantly, the good performance demonstrated since its first proposal, along with its fast execution, its reduced number of parameters, and its easy implementation are the crucial reasons which have motivated the using of BA.

With all this, the main contributions and novelties of the work presented on this paper are twofold. On the one hand, we have used an RVRP for dealing with the proposed DDSPWC. As will be explained later, similar problems have been previously presented in the scientific community, but never using an RVRP as complete as the one proposed in this study. In this sense, the main originality is not only the application of the BA to the medical distribution problem. In fact, the designed problem itself presents also a novelty, being the first time that an RVRP with these features is proposed in the literature.

On the other hand, in order to address the proposed problem, we have developed a discrete and improved version of the classic BA, named DaIBA. As far as we know, this is the first time that a BA is applied to such a complex RVRP. Additionally, the proposed technique is an adaptation of a recently proposed discrete (BA) (Osaba et al. (2016b)), which has only been applied for both Symmetric and Asymmetric Traveling Salesman Problem. With the intention of proving that the DaIBA is a promising method for solving the raised C-VRP-P*C, we have compared its results with the ones obtained by an evolutionary algorithm (EA), an evolutionary simulated annealing (ESA) (Yip & Pao (1995)), and a Firefly Algorithm (FA) (Yang (2009)).

The structure of this paper is as follows. The following Section 2 is devoted to the problem formulation. In this section, first, we describe the real-world problem that motivated this study. After that, we present the proposed RVRP. In Section 3, the designed DaIBA is deeply described. Furthermore, the experimentation performed is detailed in Section 4, along with the proposed benchmark. Finally, we end this paper with the conclusions of the study, and our planned future work (Section 5).

2. Problem formulation

This section is divided into two different parts. The first one, Section 2.1, is dedicated to the conceptual definition and description of the problem. The main intention is to contextualize the study and highlight its real-world application. After that, the designed C-VRP-P*C is deeply detailed in Section 2.2, in which an overall description of the problem is depicted, as well as its mathematical formulation.

2.1. Drugs distribution and pharmacological waste collection

As has been discussed in the introduction, the real-world problem addressed in this paper is related to the distribution of drugs to hospitals, neighborhood health centers and drugstores. Specifically, the problem arises in a regional pharmaceutical distributor.

This distribution company serves the demand of hospitals, drugstores and health centers located in several cities and towns. The distribution company offers two services: delivery of prescription drugs and collection of pharmacological waste and expired or deteriorated medicines. The second service is aimed at collecting spoiled medicines and pharmacological wastes. These residues, like bio-sanitary waste, cannot be deposited in the usual trash containers since they must be processed in a special way.

The objective of the work presented in this paper is the design of an algorithm that plans the distribution and collection routes that minimize the operating costs of the distribution company. In addition to costs, the company's logistics planning is based on a series of principles. The first principle is to treat each city as a separate unit. In this sense, when a vehicle arrives in a city, it must take care of all the requests (distribution or collection) that the sanitary centers or drugstores of that location have. Therefore, a vehicle cannot enter a city and town if it does not have a sufficient capacity to attend all the requests of that location. The second principle is related to the schedule in which requests are handled. Requests are only served between 6:00 am and 3:00 pm. In addition, within this temporary window there is a range called "peak hours" (in this paper that range is set between 8:00 am and 10:00 am). The costs of traveling from one place to another are higher in the "peak hours". This range tries to simulate the temporary moments in which the traffic is denser in the cities and towns. Additionally, all vehicles must respect the rules of circulation. Therefore, the graph that configures the road map between the different locations will not be composed entirely of bi-directional links. Forbidden links will be defined, as if they were real roads. Finally, in order not to elaborate extremely long journeys for a single worker, all the routes have a maximum duration which cannot be exceeded.

Throughout the past few decades drug distribution problems have been modeled as classic VRP or as a variant to incorporate some constraints. For this reason, it is difficult to find works that focus specifically on vehicle routing for the drug distribution sector. More recently, with the rise of home health care systems, papers that address route planning for drug delivery can be found. In (Liu et al. (2013)), Liu et al. proposed a metaheuristic based on a Genetic Algorithm and a Tabu Search for home health care logistics. They model the problem as a VRPTW with delivery and pickup. The problem addressed has two types of delivery (from depot to patient and from hospital to patient) and two types of pickup (from patient to depot and from patient to medical lab). Authors test their new algorithm with instances derived from existing VRPTW benchmarks. Other work, also in the context of home health care, can be found in (Liu et al. (2014)). In this case, the problem used as reference is a Periodic Vehicle Routing Problem with Time Windows (PVRPTW). The problem involves 3 types of patient demands: transportation of drugs/medical devices between the depot and patients' homes, delivery of special drugs from the hospital to patients, and delivery of blood samples from patients to the medical lab. To solve the problem a metaheuristic based on the classical Tabu Search is defined.

The scope of application presented in the present work is novel. In addition, the problem of routing modeled also presents original aspects with respect to other problems of routing applied to problems of distribution in the sanitary field. In addition, as we will see in the rest of the paper, the RVRP proposed in this work has a great number of constraints, making easier its application to the real world.

2.2. Clustered Vehicle Routing Problem with Pickups and Deliveries, Asymmetric Variable Costs, Forbidden Roads and Cost Constraints

As has been pointed in the introduction, the real-world situation faced in this paper has been modeled as a RVRP. Now, in this section, we describe in depth the presented RVRP problem. First, in Section 2.2.1, we detail the basic characteristics of the problem. Then, in Section 2.2.2, the mathematical formulation is represented.

2.2.1. Overall description of the proposed problem

The proposed RVRP has been modeled taking into account each and every condition mentioned in Section 2.1. Furthermore, it should be borne in mind that we have considered some additional restrictions in order to develop a model closer to real-world conditions. Hence, the proposed RVRP has the following general features.

1. *Clustered*: This feature means that the clients placed in the environment are grouped in several clusters. In this sense, every cluster corresponds to a city. Additionally, if any vehicle enters a city, it must meet the demand of every customer placed here. In other words, a vehicle is not allowed to enter a cluster if it cannot meet the demand of all the clients belonging to this city. This same feature has been used studied in several papers of the literature (Defryn & Sörensen (2017); Expósito-Izquierdo et al. (2016)).
2. *Pickup and Delivery*: This feature has been use in several studies previously (Männel & Bortfeldt (2016); Avci & Topaloglu (2016)). This characteristic contemplates two different kind of nodes: the *delivery nodes* and the *pickup nodes*. On the one hand, *delivery nodes* are those points where medical supplies are delivered. On the other hand, in *pickup nodes* the used medical stuff is collected, with the aim of taking them back to the warehouse. It is important to mention that this feature has a *simultaneous* nature. In this way, a drugstore or sanitary center can ask for both delivery and collection of material. For this reason, *delivery-pickup nodes* can also be found. Finally, it is assumed that all clients request the delivery of material. Therefore, drugstores or health centers demanding only the pickup of material are not present in our scenario.
3. *Asymmetric Variable Travel Times*: In real logistic problems, the travel between two different points does not always take the same time, or the same cost. In almost all the cases, this cost is under the influence of some external variables. With the aim of creating a more realistic model, we have represented this situation in the problem that we have proposed in this study. To this end, we have fixed a working-day between 6:00 am and 3:00 pm. Within this schedule, we have set two time-periods: "peak hours" and "off-peak hours". The first period is from 8:00 am to 10:00 am. All travels carried out at this time window will imply higher costs. On the other hand, the same trips will take less time if they are conducted in the "off-peak" period. Additionally, all the traveling costs are asymmetric, meaning that the effort of traveling from one node i to another node j implies different costs comparing with the reverse trip. This specific feature is appreciated in real-world applications, and it has been previously used on this kind of problem (Leggieri & Haouari (2016); Pham et al. (2014)).
4. *Forbidden Roads*: In real-world situations, it is quite common to find roads in which the traffic is allowed only in one direction. Furthermore, we can also find

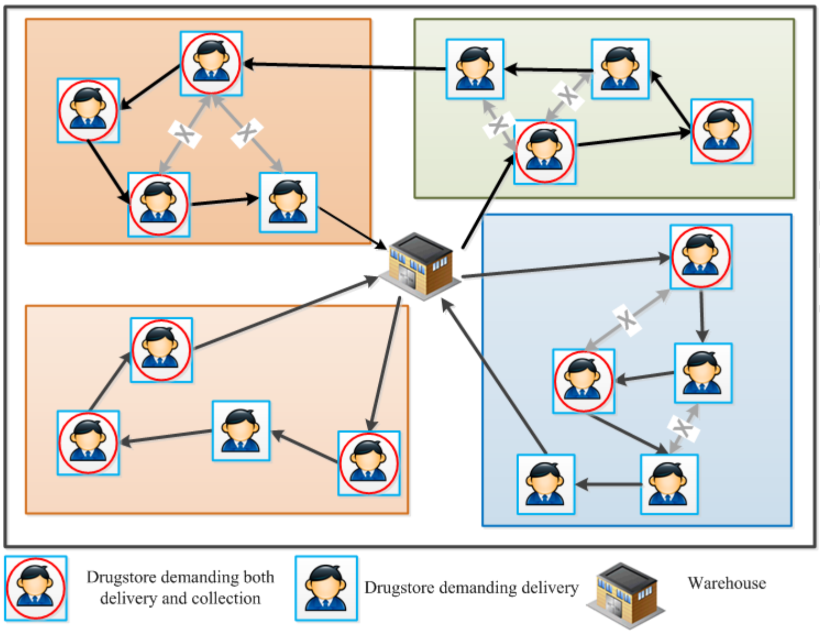


Figure 1: Possible C-VRP-P*C dataset composed of 15 nodes, and one feasible solution. Gray arcs represent forbidden paths.

pedestrian streets, where vehicles are prohibited to go through. With the intention of recreating this kind of paths, the proposed C-VRP-P*C has certain arcs (i, j) which are not allowed to be used in the final solution.

5. *Cost-Constrained*: The last characteristic is related with the maximum cost that a route can afford. This constraint, as can be easily deduced, guarantees that the total cost of the arcs in a single route does not exceed a maximum route cost. This feature ensures the avoidance of long routes, giving priority to a more distributed planning between the fleet of vehicles. This same characteristic has been referenced many times in the literature (Li et al. (1992)).

With all these features, the proposed C-VRP-P*C is an RVRP, whose main objective is to find a group of routes, taking into account the two different types of clients, trying to minimize the total traveling costs, not going through forbidden roads, and respecting the restrictions imposed by the capacity of the vehicles (C), the maximum allowed cost per route and the clusters. We show in Figure 1 a possible 15-noded dataset of the presented problem. We also show a feasible solution to this dataset in the same figure.

2.2.2. Mathematical formulation of the presented problem

The proposed C-VRP-P*C can be represented as a complete graph $G = (V, A)$, where $V = \{v_0, v_1, \ldots, v_n\}$ is the set of vertex which depicts the drugstores and sanitary centers that comprise the system. Furthermore, $A = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$ is the set of arcs which represents the interconnections between drugstores, hospitals and health centers. Each arc of the system has associated a c_{ij} cost. As we have mentioned above, the presented problem has asymmetric costs, for this reason, the cost of traveling from i to j is always different from the cost of traveling from j to i. In a formal way, $c_{ij} \neq c_{ji}$. Furthermore, in order to contemplate forbidden arcs, we have fixed as infinite the cost of these paths. This way, we can ensure that these arcs will not appear in the final solution. Additionally, each route cannot exceed a maximum cost of D.

Additionally, the vertex v_0 represents the depot, and the rest are the visiting drugstores or sanitary centers. Besides this, V is divided into *cl+1* mutually exclusive non-empty subsets, $CL = \{V_0, V_1, ..., V_{cl}\}$, each one for each cluster. These subsets are subject to these two conditions:

$$V_x \cap V_y = \emptyset, \quad x, y \in 0, 1, ..., cl, x \neq y \tag{1}$$

$$V = V_0 \cup V_1 \cup ... \cup V_{cl} \tag{2}$$

It should be borne in mind that V_0 contains only v_0. The remaining n hospitals, health centers and drugstores are distributed into cl different clusters. Furthermore, client i has two types of demands: one of them associated with the delivery of supplies, $d_i > 0$, and the other with the pick-ups $p_i \geq 0$.

Before showing the mathematical formulation of the proposed C-VRP-P*C, it should be highlighted that y_{ij} represents the demand picked-up in clients routed up to node i (including node i), and transported in the arc (i, j). Besides this, the total number of routes has been represented as k. Additionally, the parameter z_{ij} depicts the demand to be delivered to customers scheduled after node i and transported in arc (i, j) (Montané & Galvao (2006)). Furthermore, the binary variable x^r_{ij} is 1 if the vehicle r uses the arc (i, j), and 0 otherwise. Finally, w^r_s is a binary variable, which takes the value of 1 if the mobile unit r enters the cluster s, and 0 in other case. With all this information, the presented C-VRP-P*C can be mathematically formulated in the following way, where the main problem is now to minimize:

$$\sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{r=1}^{k} c_{ij} x^r_{ij} \tag{3}$$

subject to:

$$\sum_{j=0}^{n} \sum_{r=1}^{k} x^r_{ij} = 1, \quad i = k, \ldots, n; j \neq i, \tag{4}$$

$$\sum_{i=0}^{n} \sum_{r=1}^{k} x^r_{ij} = 1, \quad j = 0, \ldots, n; i \neq j, \tag{5}$$

$$\sum_{i=0}^{n} \sum_{r=1}^{k} x^r_{i0} = k, \tag{6}$$

$$\sum_{j=0}^{n} \sum_{r=1}^{k} x^r_{0j} = k, \tag{7}$$

$$\sum_{j=0}^{n} x^r_{ij} - \sum_{l=0}^{n} x^r_{li} = 0, \; i = 0, \ldots, n; r = 1 \ldots k, \tag{8}$$

$$\sum_{i=0}^{n} x_{ij}^{r} - \sum_{l=0}^{n} x_{jl}^{r} = 0,\ j = 0,\dots,n; r = 1\dots k, \tag{9}$$

$$\sum_{i=0}^{n} \sum_{r=1}^{k} d_{ij} x_{ij}^{r} < \infty, \quad j = 0,\dots,n; i \neq j, \tag{10}$$

$$\sum_{j=0}^{n} \sum_{r=1}^{k} d_{ij} x_{ij}^{r} < \infty, \quad i = 0,\dots,n; j \neq i, \tag{11}$$

$$\sum_{r=1}^{k} w_{s}^{k} = 1 \quad s = 1,...,c. \tag{12}$$

$$\sum_{i=0}^{n} z_{ji} - \sum_{i=0}^{n} z_{ij} = d_j, \quad j = 0,\dots,n, \tag{13}$$

$$\sum_{i=0}^{n} y_{ji} - \sum_{i=0}^{n} y_{ij} = p_j, \quad j = 0,\dots,n, \tag{14}$$

$$y_{ij} + z_{ij} \leq Q \sum_{r=1}^{k} x_{ij}^{r}, \quad i,j = 0,...,n, \tag{15}$$

$$\sum_{j=0}^{n} \sum_{r=1}^{k} c_{ij} < D, \quad i = 0,\dots,n; j \neq i, \tag{16}$$

where

$$y_{ij} \geq 0, \quad i,j = 0,\dots,n, \tag{17}$$

$$z_{ij} \geq 0, \quad i,j = 0,\dots,n. \tag{18}$$

$$w_{s}^{r} \in \{0,1\}, \quad r = 1,\dots,k; s = 1,...,c, \tag{19}$$

$$x_{ij}^{r} \in \{0,1\}, \quad i,j = 0,\dots,n; i \neq j; r = 1\dots k, \tag{20}$$

The first formula depicts the objective function, which must be minimized, and which is the sum of all the costs associated with the routes that compose the solution. Conditions (4) and (5) guarantee that all the drugstores and sanitary centers are visited exactly once. Additionally, equations (6) and (7) assure that the total amount of vehicles leaving the depot, and the number of vehicles that return to it is the same. Furthermore, the proper flow of each route is ensured by restrictions (8) and (9), avoiding the generation of subloops.

On the other hand, formulas (10) and (11) guarantee that every trip between two different nodes has not an infinite cost. In this way, we ensure that forbidden paths will not form part of the final solution. Moreover, function (12) assures that only one vehicle enters every cluster. This constraint, along with the above described (4) and (5) ensures that all the customers belonging the same cluster are visited by the same mobile unit.

In addition, constraints (13) and (14) guarantee that the flows for the delivery and the pick-ups are properly conducted. These clauses ensure that both demands are correctly satisfied for every drugstore and sanitary center. Besides that, formula (15) assures that the total capacity of any vehicle is always respected. This same restriction also represents that both delivery and collection demands will only be transported using arcs included in the solution (Montané & Galvao (2006)). Furthermore, constraint (16) guarantees that the total cost of each route does not exceed the fixed maximum. Finally, the formulas (17), (18), (19) and (20) represent the domains of the variables y_{ij}, z_{ij}, w_s^r and x_{ij}^r.

It is interesting to highlight that all the constraints inherent to the problem make the generating of feasible solutions a very complex task. This complexity makes impossible to directly apply most of the operators used for solving the common VRP. For this reason, the developing of appropriate functions has been one of the main difficulties for its solving. On the other hand, it is also noteworthy that all the constraints reduce the size of the search space which comprises all the feasible solutions, but increments the probability of falling into local optima. In order to avoid this fact, a simple but effective improvement has been implemented in the proposed DaIBA, which help to enhance its exploration ability. This mechanism, which endows each bat with a certain intelligence for performing its movements, is explained in the following Section 3.2.

3. Bat algorithm

As we have mentioned in the introduction of this work, a Discrete and Improved Bat Algorithm (DaIBA) is presented in this paper to face the designed C-VRP-P*C. In the present section, we introduce first the classic version of the BA (Section 3.1). After that, we describe in detail in Section 3.2 the proposed DaIBA.

3.1. Classic Bat Algorithm

As it has been mentioned in previous sections, the BA is a nature-inspired metaheuristic based on the echolocation system of bats. In the nature, bats emit ultrasonic pulses to the surrounding environment with navigation and hunting purposes. After the emission of these pulses, bats listen to the echoes, and based on them they can locate themselves and also identify and locate preys and obstacles. Besides that, each bat is able to find the most "nutritious" areas performing an individual search, or moving towards a "nutritious" location previously found by any other component of the swarm.

It is important to mention that some rules have to be previously established with the aim of making an appropriate adaptation (Yang (2010b)):

1. All bats use echolocation to detect the distance, and they have one "magic ability" that permit them to distinguish between an obstacle and a prey.
2. All bats fly randomly with a velocity v_i at position x_i with a fixed frequency f_{min}, varying wavelength λ and loudness A_i to search for a prey. In this idealized rule, it is assumed that every bat can adjust in an automatic way the frequency (or wavelength) of the emitted pulses, and the rate of these pulses emission $r \in [0, 1]$. This automatic adjustment depends on the proximity of the targeted prey.
3. In the real world, the bats emissions loudness can vary in many different ways. Nevertheless, we assume that this loudness can vary from a large positive A_0 to a minimum constant value A_{min}.

Algorithm 1: Pseudocode of the basic BA

```
1  Define the objective function f(x);
2  Initialize the 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 21, 22 and 23;
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;
```

We show the pseudocode of the classic BA in Algorithm 1. If we analyze this pseudocode, we can see that lines 1-6 correspond to the initialization process. Initially, both objective function and initial population are defined. In this sense, every bat of the population represents one possible solution to the faced problem. After this, every parameter related to each bat is initialized and defined. These parameters are the frequency f_i, velocity v_i, loudness A_i and pulse rate r_i.

After these initialization steps, the method begins its main phase. For each generation, every bat of the population moves by updating its position and velocity. For these movements, the following equations are used:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \tag{21}$$

$$v_i^t = v_i^{t-1} + [x_i^t - x_*]f_i \tag{22}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{23}$$

where the parameter β is a randomly generated number in the [0,1] interval. Furthermore, x_* represents the current best solution in the swarm, and x_i^t and v_i^t denote position and velocity of a bat i at time step t. Finally, the results of Equation (21) are used to control the pace and range of bats movement. Additionally, for the local search part, if one solution is selected among the best ones, a new solution for each bat is generated using a random walk.

$$x_{new} = x_{old} + \epsilon A^t \tag{24}$$

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 loudness A_i and the rate r_i of each bat are updated whether the conditions shown in the line 14 of Algorithm 1 are met. This update is carried out following these formulas:

$$A_i^{t+1} = \alpha A_i^t \tag{25}$$

$$r_i^{t+1} = r_i^0[1 - \exp(-\gamma t)] \tag{26}$$

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

$$A_i^t \to 0, r_i^t \to r_i^0, \ as \ t \to \infty \tag{27}$$

In many studies, $\alpha = \gamma$ is used in order to simplify the implementation of the technique. In the present work, we use $\alpha = \gamma = 0.98$. We have chosen this value after an empirical process using a [0.90, 0.99] range.

3.2. The proposed Discrete and Improved Bat Algorithm

Before starting with the deep description of our proposed DaIBA, it should be highlighted that the original version of the BA was developed primarily for addressing continuous optimization problems. Therefore, the classic BA cannot be directly applied to solve the presented C-VRP-P*C. For this reason, we have performed some modification to the original BA with the aim of preparing it for facing the designed C-VRP-P*C.

In the proposed DaIBA, every bat in the swarm represents a feasible solution for the C-VRP-P*C. Besides that, as has been mentioned in Section 2.2.2, the objective function is the sum of all the costs associated with the routes that composes the solution. In this sense, the C-VRP-P*C is a minimization problem, in which the bats with the lower objective function value are the best ones.

Additionally, for the solution encoding, we have used the permutation codification. This means that every solution is represented as a unique permutation of numbers, depicting the routes that comprise the solution. Additionally, routes in the same solution are separated by a zero in order to distinguish them. Next, an example of a 20-noded fictitious dataset is represented, composed by four clusters and two different routes. For this example, a horizontal bar has been used with the intention of visually distinguishing the clusters within the same route.

$$x_1 : \{10, 1, 12, 3, 14 \mid 5, 16, 7, 8, 9, \boldsymbol{0}, 11, 13, 2, 6, 4, 19 \mid 18, 17, 15, 20\},$$

In relation to the basic parameters of the classic BA, the philosophy of r_i and A_i has remained exactly in the same form. Additionally, with the intention of simplifying the complexity of the algorithm, the parameter f_i ("frequency") has not been taken into account in our DaIBA. Finally, the "velocity", v_i, has been modified. In the continuous version of the BA, this parameter is calculated as has been shown in Equation (22).

Analyzing this formula, it can be deduced that the velocity of a bat i at time step t relies on the v_i of the bat i in the previous time step, the f_i parameter, and the difference between i and the best bat in the swarm. As can be easily understood, the parameter v_i cannot be used in the same way for solving a discrete problem as our proposed C-VRP-P*C. With the objective of adapting our DaIBA as accurately as possible, we relate v_i with the distance between the bat i and the best bat of the swarm. We consider that this approach emulates faithfully the concepts of the classic BA. For this reason, we have adapted v_i using the well-known Hamming Distance in the following way:

$$v_i^t = \text{Random}[1, \text{HammingDistance}(x_i^t, x_*)] \tag{28}$$

In other words, 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. It should be pointed that despite the Hamming Distance have been previously used for measuring this kind of distances (Osaba et al. (2016a,b)), it is still few researches the ones that have adopted it. This fact enhances the innovation factor to our method.

In the proposed problem, the comparison between two different bats is performed cluster by cluster. For instance, taking two random bats, and one random cluster c composed of 10 nodes:

$$x_1(\text{cluster-}c) : \{10, 1, 2, 3, 4, 5, 6, 7, 8, 9\},$$

$$x_2(\text{cluster-}c) : \{10, 9, 7, 8, 4, 5, 6, 1, 3, 2\},$$

the Hamming Distance between x_1 and x_2 for the cluster c is 6. This comparison is also made for every cluster of the solution. Therefore, the distance between two bats is the sum of all the distances for every cluster.

Additionally, regarding the generation of new solutions, in the classic BA the movement of the bats is conducted using the Equation 23:

$$x_i^t = x_i^{t-1} + v_i^t$$

It can be deduced from this formula that the position of a bat i at time step t depends on the v_i of the bat i and its previous position at time step $t-1$. Anyway, this equation cannot be applied directly the proposed C-VRP-P*C. For this reason, a modification has been proposed. Thereby, the movement of a bat i is determined by

$$x_i^t \leftarrow InsertionFunction(x_i^{t-1}, v_i^t) \tag{29}$$

Namely, at every generation, every bat examines a v_i number of its neighbors, and it selects the best one as its current movement. In other words, the bat i conducts a v_i number of *InsertionFunction* executions, and it chooses the best one. In this sense, we have used the *InsertionFunction* as movement function. This function selects and extracts one random node from a random route. Then, this node is re-inserted in a random position inside its cluster. This function takes into account the capacity constraint, in order not to create infeasible solutions. Next, a simple example of a feasible *InstertionFunction* operation is shown for a 10-noded cluster. In this case, the node *9* has been randomly selected for being extracted and re-inserted in a random position.

$$\begin{aligned} x_i(\text{cluster-}c) := (1, \mathbf{9}, 2, 4, 3, 10, 5, 8, 6, 7) &\rightarrow x_i'(\text{cluster-}c) := (1, 2, 4, 3, 10, 5, 8, 6, 7) \\ &\rightarrow x_i'(\text{cluster-}c) := (1, 2, 4, 3, \mathbf{9}, 10, 5, 8, 6, 7) \end{aligned}$$

In addition to the *InsertionFunction*, we have also used the *ExchangeFunction* as will be described later. In this operator, two random nodes are selected from a randomly selected cluster, and they exchange their position. The following example depicts a possible *ExchangeFunction* operation for the same 10-noded cluster shown before. In this case, the nodes *1* and *4* of the cluster have been randomly selected to be exchanged.

$$x_i(\text{cluster-}c) := (\mathbf{1}, 9, 2, \mathbf{4}, 3, 0, 5, 8, 6, 7) \rightarrow x_i'(\text{cluster-}c) := (\mathbf{*}, 9, 2, \mathbf{*}, 3, 0, 5, 8, 6, 7)$$
$$\rightarrow x_i'(\text{cluster-}c) := (\mathbf{4}, 9, 2, \mathbf{1}, 3, 0, 5, 8, 6, 7)$$

It is important to highlight that these operators have been used due to their adequacy for dealing with the high demanding constraint of the problem, generating feasible solutions in every execution and helping to the improvement of the swarm. 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), and a local solution is generated around this one. To generate this new solution, the best neighbor of the selected bat is chosen using also the *InsertionFunction*.

Besides that, we also provide to our proposed method a simple but effective improvement in its structure. This innovative improvement is related with the movement behavior of the bats, and it has been applied in order to avoid the facility of falling into local optima. In most of BA versions, all the bats perform their movement using the same pattern throughout the entire execution. In the proposed DaIBA, some type of intelligence has been provided to all the bats of the swarm. In this way, each bat moves in a different way depending on its position regarding the best bat of the population.

Thereby, when a bat i is prepared to perform a movement, it examines its v_i^t. If this parameter is high (greater than $n/2$, where n is the number of nodes of the problem dataset), it can be assumed that it is far from the best individual of the swarm. Therefore, it can be concluded that it needs a *wide move*. Otherwise, whether $v_i^t < n/2$, we can think that the bat is in a promising point of the space of solutions. For this reason, this bat will perform a *narrow move*. In our DaIBA, the *InsertionFunction* has been used for *narrow moves*, and the *ExchangeFunction* for *wide moves*.

This simple modification allows the bats to crawl the space of solutions using different neighborhood structures along the execution. This fact considerably enhances the exploration capacity of the algorithm, leading to an improvement in the results quality, and decreasing the probability of falling into local optima. The advantages of this mechanism have been tested in previous studies (Osaba et al. (2016b)). Finally, the pseudocode of the proposed DaIBA is depicted in Algorithm 2. Furthermore, in order to enhance the understandability of the method, its flowchart is shown in Figure 2.

4. Experimentation

The experimentation performed in this study is detailed in this section. First, we detail in Section 4.1 the designed benchmark for the developed C-VRP-P*C. Then, in Section 4.2, we present the outcomes get by the proposed DaIBA for the above mentioned benchmark. It should be highlighted that we have compared these obtained results with the ones obtained by the EA and the ESA. This comparison has been made in Section

Algorithm 2: Pseudocode of the proposed DaIBA. n = number of nodes of the dataset.

Define the objective function $f(x)$;
Randomly initialize the bat population $X = x_1, x_2, ..., x_n$;
for *each bat* x_i *in the population* **do**
  Initialize the pulse rate r_i, velocity v_i and loudness A_i;
end
repeat
  for *each bat* x_i *in the population* **do**
    v_i^t = Random[1,HammingDistance(x_i^t,x_*);
    if v_i^t<*n/2* **then**
      $x_i \leftarrow InsertionFunction(x_i^{t-1}, v_i^t)$;
    else
      $x_i \leftarrow ExchangeFunction(x_i^{t-1}, v_i^t)$;
    end
    if *rand*>r_i **then**
      Select one solution among the best ones;
      Generate a new bat selecting the best neighbor around the chosen bat using the InsertionFunction or the ExchangeFunction;
    end
    if *rand*<A_i *and* $f(x_i)$<$f(x_*)$ **then**
      Accept the new solution;
      Increase r_i and reduce A_i;
    end
  end
until *termination criterion reached*;
Rank the bats and return the current best bat of the population;

4.3 with the aim of proving that our DaIBA is a promising method for facing routing problems. Finally, we have conducted two different statistical tests in Section 4.4.

*4.1. The benchmark proposed for the C-VRP-P*C*

The problem proposed in this paper for solving the medical material distribution problem has never been faced before in the scientific community. For this reason, it is not possible to find a benchmark in the literature for the C-VRP-P*C. In this sense, and following the good practices described in (Osaba et al. (2018)), a benchmark composed of 24 instances has been proposed for this study. These instances are composed of 60 to 1000 nodes. As we have explained before, all the nodes are placed in the province of Bizkaia, in The Basque Country, Spain.

We have fixed the maximum number of clusters in 12, being also instances with 6 and 9 of them. In Figure 3, a map with the geographical locations of the central depot, the clusters and all the drugstores and sanitary centers is represented. This map has been made using Open Street Maps technology, via uMap tool[2]. Furthermore, this map

[2]http://umap.openstreetmap.fr

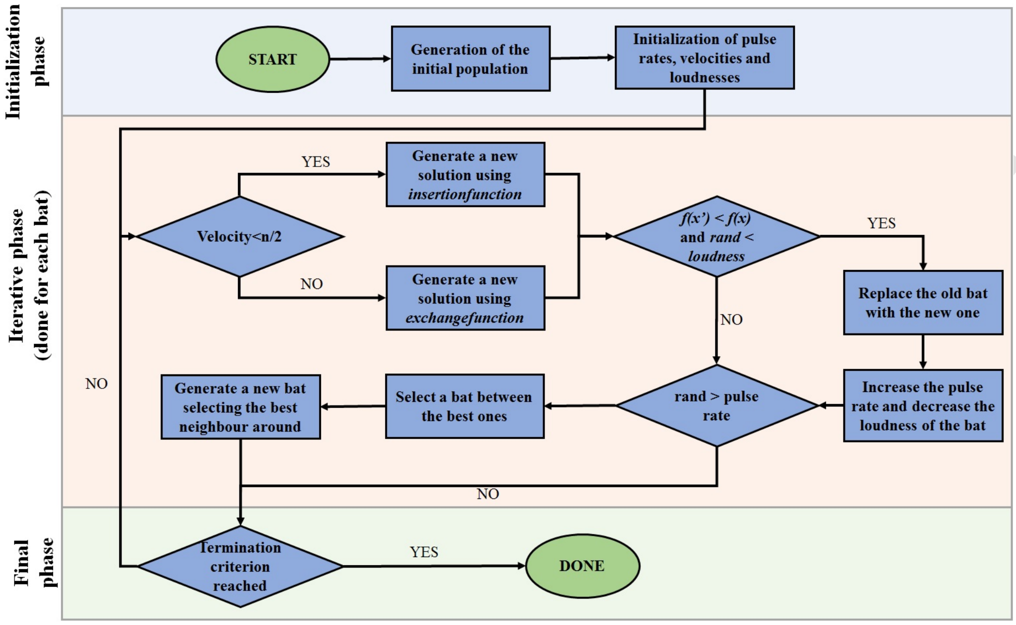


Figure 2: Flowchart of the proposed DaIBA.

represents the 120 real nodes existing on the designed environment. In addition to these 120 nodes, additional fictitious ones have been placed in further synthetic datasets, in order to prove the performance of the algorithm in larger instances. These four cases are the ones composed by 200, 500 and 1000 modes.

Clusters have been built sequentially. In other words, clients 1 to 10 compose the first cluster, clients 11 to 20 the second one, and so on. For the fictitious additional nodes that compose the large instances, they have been placed sequentially in the already existing nodes. This means that node 121, 133, 145 and so on have been place in cluster one, as well as 122, 134, 146 and so on in cluster two. This same trend has been applied for all the cluster and nodes. It should be born in mind that all clusters have the same size, in terms of number of nodes (except for larger instances, in which some of the clusters have one more node). Additionally, as has been previously mentioned, each node has two different demands assigned: deliveries (d_i), and pickups (p_i). The following method has been followed for the setting of these demands, taking into account that d_0=0 and p_0=0, since v_0 is considered the depot:

$$d_i = 5,\ p_i = 0, \quad \forall i \in \{1, 5, 9, \ldots, 997\}, \tag{30}$$

$$d_i = 5,\ p_i = 5, \quad \forall i \in \{2, 6, 10, \ldots, 998\}, \tag{31}$$

$$d_i = 10,\ p_i = 0, \quad \forall i \in \{3, 7, 11, \ldots, 999\}, \tag{32}$$

$$d_i = 10,\ p_i = 5, \quad \forall i \in \{4, 8, 12, \ldots, 1000\}, \tag{33}$$

Besides that, the traveling cost from any drugstore or sanitary center i to other client j have been established following the procedure represented in Algorithm 3. These

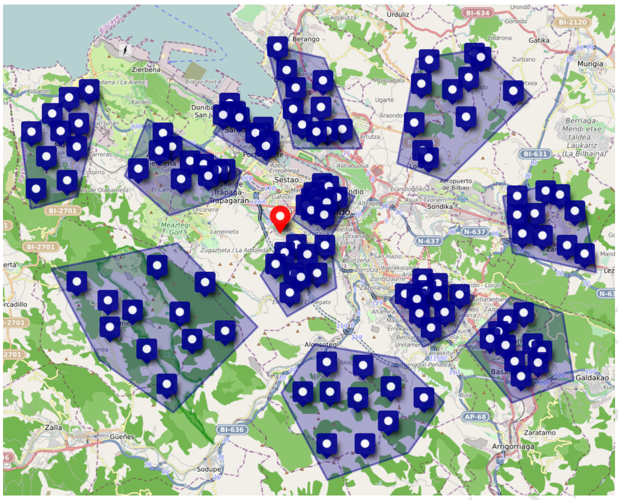


Figure 3: Geographical locations of the real nodes in Bizkaia. Source: Open Street Maps, via uMap, accessed September 2017.

Algorithm 3: Method for the assignment of travel costs for "off-peak" period.

for $\forall i \in \{1, 2, \ldots, 999\}$ **do**
    for $\forall j \in \{i+1, \ldots, 1000\}$ **do**
        c_{ij} = EuclideanDistance(i,j);
        if *j is an odd number* **then**
            c_{ji} = EuclideanDistance(j,i) $\cdot$ 1.3 ;
        else
            c_{ji} = EuclideanDistance(j,i) $\cdot$ 0.7 ;
        end
    end
end

costs are assigned for the "off-peak" period, and they are incremented when they are conducted on the "peak" period. These incremented costs are assigned following the method represented in Algorithm 4. It is interesting to mention that these two methods assure the asymmetry characteristic of the problem. Additionally, it is also interesting to point that the trip time between two customers is the same as its traveling cost (in seconds).

At last, depending on the instance, some trips of every cluster have been chosen to be forbidden.

In order to facilitate the understanding of this benchmark, we have summarized in Table 1 the characteristics of all the 24 developed instances. Additionally, some clarifications should be made to correctly understand this table. DaIBA_RVRP_60_1_1 and DaIBA_RVRP_60_1_2 are made up by six clusters, which are {1, 3, 5, 7, 9, 11}. Furthermore, DaIBA_RVRP_60_2_1 and DaIBA_RVRP_60_2_2 are composed of clusters {2, 4, 6, 8, 10, 12}. On the other hand, each cluster in DaIBA_RVRP_60_1_3 and

Algorithm 4: Method for the assignment of travel costs for "peak" period.

```
for ∀i ∈ {1, 2, ..., 999} do
    for ∀j ∈ {i + 1, ..., 1000} do
        c_ij = EuclideanDistance · 1.4;
        if j is an odd number then
            c_ji = (EuclideanDistance(j,i) · 1.2) · 1.3 ;
        else
            c_ji = (EuclideanDistance(j,i) · 0.8) · 1.5 ;
        end
    end
end
```

DaIBA_RVRP_60_1_4 is composed of the first five drugstores, hospitals or health centers, while in DaIBA_RVRP_60_2_3 and DaIBA_RVRP_60_2_4 clusters are comprised by the last five nodes. Finally, to create all DaIBA_RVRP_9X_X instances, the first 9 clusters, or the first 8 customers (depending on the case) have been chosen.

Following the good practices shown in (Osaba et al. (2018)), and with the intention of enhancing the replication of this whole experimentation, the benchmark developed for C-VRP-P*C is available under request to the corresponding author of this paper.

4.2. Results

First of all, it is interesting to mention that all the tests conducted in this work have been run on an Intel Xeon E5 – 2650 v3 computer, with 2.30 GHz and a RAM of 32 GB. Furthermore, we have utilized all the instances described in Section 3 for the experimentation, running each of them 30 times. As has been mentioned in the introduction of this paper, we have compared the results obtained by the proposed DaIBA with the following three techniques:

- *Evolutionary Simulated Annealing* (ESA): SA is one of the most used local search method in the literature, and is based on the physical principles explaining the metal cooling process. For the sake of fairness, an evolutionary version of the SA is used (ESA) (Yip & Pao (1995)) so as to consider a population-based method roughly similar to the rest of considered solvers. For the developed ESA, two different successor functions have been used, which means that every individual has its own randomly assigned successor function.

- *Evolutionary Algorithm* (EA): The EA developed for this study is a mutation based evolutionary algorithm which bases the movement of its individuals in different mutation operators. Following the same procedure of the above described ESA, two different mutation operators have been used for this technique. In this sense, and before each mutation is applied, the operator that will be used is randomly selected. Similar algorithms have been employed in previously published papers, such as (Osaba et al. (2016b)), (Conesa-Muñoz et al. (2016)) or (Wang et al. (2015)), which have inspired us to using this method for the comparison.

Instance	Nodes	Clusters	Vehic. capacity	Max. Cost	Forbidden trips
DaIBA_RVRP_60_1_1	60	6	350	35k	6
DaIBA_RVRP_60_1_2	60	6	200	35k	12
DaIBA_RVRP_60_1_3	60	12	350	35k	6
DaIBA_RVRP_60_1_4	60	12	200	35k	12
DaIBA_RVRP_60_2_1	60	6	350	35k	6
DaIBA_RVRP_60_2_2	60	6	200	35k	12
DaIBA_RVRP_60_2_3	60	12	350	35k	6
DaIBA_RVRP_60_2_4	60	12	200	35k	12
DaIBA_RVRP_90_1	90	9	350	40k	6
DaIBA_RVRP_90_2	90	9	200	40k	12
DaIBA_RVRP_96_1	96	12	350	40k	6
DaIBA_RVRP_96_2	96	12	200	40k	12
DaIBA_RVRP_120_1	120	12	200	40k	6
DaIBA_RVRP_120_2	120	12	250	40k	12
DaIBA_RVRP_120_3	120	12	350	40k	18
DaIBA_RVRP_120_4	120	12	400	40k	24
DaIBA_RVRP_200_1	200	12	300	50k	10
DaIBA_RVRP_200_2	200	12	400	50k	20
DaIBA_RVRP_200_3	200	12	500	50k	30
DaIBA_RVRP_200_4	200	12	600	50k	40
DaIBA_RVRP_500_1	500	12	800	100k	30
DaIBA_RVRP_500_2	500	12	900	100k	40
DaIBA_RVRP_1000_1	1000	12	1000	150k	30
DaIBA_RVRP_1000_2	1000	12	1200	150k	40

Table 1: Characteristics of the developed benchmark for the C-VRP-P*C. *Forbidden trips* represents the quantity of forbidden trips for each cluster.

- *Firefly Algorithm*: FA was proposed in (Yang (2009)), and it is based on the flashing behavior of fireflies, which acts as a signal system to attract other fireflies. This meta-heuristic optimization algorithm has been also applied to a wide range of optimization fields and problems since its proposal, with extensive surveys (Fister et al. (2014)) evincing the momentum gained by this solver within the community. For this research, a discrete version of the FA has been developed (Osaba et al. (2016a)).

We have chosen these three techniques for the comparison since all of them are famous methods, which have been widely used for successfully solving different kind of routing problems. In this way, if we prove that the DaIBA can perform better than these methods, we can conclude that it is a promising meta-heuristic for solving the proposed problem. Finally, all these approaches have three similarities: all techniques use short-step functions for the movement of their population individuals, they are easy and intuitive to implement, and they are easily adaptable to solve new problems.

It should be highlighted that we have used similar parameters and the same operators for all the implemented algorithms. Our purpose is to analyze which technique gets better outcomes using similar operators the same number of times. Besides that, and with the aim of making easier the replicability of this study, we show in Table 2 the parametrization used for all the three meta-heuristics. For the development and parameterization of the FA, the guidelines given in (Li et al. (2015)) have been followed. Furthermore, the same population size and same movement operators have been used for both methods. Besides that, the Hamming Distance function has also been used for the distance calculation.

All the individuals are generated randomly. Furthermore, regarding the termination criterion, each meta-heuristic ends its execution when there are $n + \sum_{k=1}^{n} k$ iterations

ESA		EA		FA		DaIBA	
Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
Population size	100	Population size	100	Population size	100	Population size	100
Successor Functions	Insertion Function & Exchange function	Mutation functions	Insertion Function & Exchange function	Movement Function	Insertion Function & Exchange function	Movement Function	Insertion Function & Exchange function
Temperature	$-sup\Delta f/ln(p)$	Mutation prob.	1.0	γ	0.95	α & γ	0.98
Cooling constant	0.95	Survivor func.	70% Elitist - 30% Random	Threshold for movement selection	$n/2$	Threshold for movement selection	$n/2$
						Initial A_i^0	Random number in [0.7,1.0]
						Initial r_i^0	Random number in [0.1,0.4]

Table 2: Parametrization of all the methods. $-sup\Delta f$ represents the difference in the objective function of the best and the worse individuals of the initial population, and p=0.95. n = number of nodes of the instance

with no improvements in the best solution, where n is the problem size. It is interesting to remember that the parameters used for the DaIBA have been described in Section 3.2.

The results get by the DaIBA, ESA, EA and FA for the proposed C-VRP-P*C are represented in Table 3. In addition, we have depicted in Table 4 the best results found for each problem instance. We have shown also in this table the size of the fleet needed to carry out every solution, and the method which has obtained each of these results.

Instance	DaIBA			ESA			EA			FA		
Name	Avg.	S. dev.	Time	Avg.	S. dev.	Time	Avg.	S. dev.	Time	Avg.	S. dev.	Time
DaIBA_RVRP_60_1_1	**55761.4**	301.4	51.8	56175.5	488.6	49.8	55943.7	512.4	52.7	55894.7	482.0	50.7
DaIBA_RVRP_60_1_2	**60138.0**	267.7	53.0	60861.4	428.7	52.7	60764.4	485.6	53.8	60417.1	376.1	52.8
DaIBA_RVRP_60_1_3	**69435.1**	864.0	52.7	70133.1	998.9	51.8	69683.1	1001.2	52.1	69617.8	913.7	53.0
DaIBA_RVRP_60_1_4	**76616.8**	909.1	52.9	77002.1	1271.6	52.6	76903.3	1250.4	54.0	77132.4	1042.3	53.1
DaIBA_RVRP_60_2_1	53615.5	513.4	54.4	53843.7	776.4	53.4	53973.1	845.1	54.7	**53571.1**	673.0	54.8
DaIBA_RVRP_60_2_2	**59005.9**	700.9	52.6	59342.8	986.4	52.0	59444.8	1071.0	53.4	59384.5	974.5	52.9
DaIBA_RVRP_60_2_3	**67438.0**	1121.8	51.9	68110.3	1227.5	50.8	67866.0	1202.0	51.8	67618.8	1274.6	53.4
DaIBA_RVRP_60_2_4	81762.3	1077.4	52.7	81939.9	1364.4	50.1	**81698.4**	1419.1	52.0	81758.4	1202.8	54.6
DaIBA_RVRP_90_1	83137.8	1342.9	93.4	**82989.4**	1645.7	90.7	83325.0	1732.6	91.7	83302.9	1541.4	95.7
DaIBA_RVRP_90_2	86110.0	1417.0	97.8	**86031.2**	1794.0	96.4	86170.7	1833.8	97.0	86201.7	1671.8	98.0
DaIBA_RVRP_96_1	**86734.4**	1437.4	96.1	86843.7	1869.9	95.1	86900.1	2013.4	96.2	86868.1	1645.5	96.4
DaIBA_RVRP_96_2	**90114.1**	1701.3	96.9	90878.0	2113.1	94.0	90437.7	2275.0	96.7	90346.4	2120.1	98.0
DaIBA_RVRP_120_1	109135.9	2001.7	184.4	109961.7	2597.7	180.3	109073.1	2734.5	182.0	**109008.2**	2399.8	188.0
DaIBA_RVRP_120_2	**107332.0**	1982.0	189.0	108039.6	2466.1	186.2	107834.8	2406.0	188.7	107564.5	2374.2	190.4
DaIBA_RVRP_120_3	**102272.8**	2043.1	180.6	102844.3	2509.4	177.9	102866.1	2847.2	177.5	102611.3	2444.5	182.9
DaIBA_RVRP_120_4	**98131.7**	2131.8	182.1	98747.6	2685.6	176.4	98441.0	2797.8	178.8	98311.7	2294.8	179.0
DaIBA_RVRP_200_1	**153176.4**	2400.3	263.7	153516.7	3281.9	259.7	153436.4	3127.4	261.8	153364.7	2700.9	272.9
DaIBA_RVRP_200_2	151371.4	2315.1	256.4	152283.1	3184.9	258.4	152301.7	3274.6	262.0	**151355.6**	2755.4	260.7
DaIBA_RVRP_200_3	150735.6	2571.1	260.7	151300.3	3201.7	255.7	151539.7	2847.2	258.5	**150655.1**	3004.5	261.0
DaIBA_RVRP_200_4	**147390.1**	2643.8	252.0	147905.7	3105.1	250.8	148001.7	2797.8	178.8	147644.0	2858.8	255.4
DaIBA_RVRP_500_1	**200253.8**	3800.3	334.5	200635.7	4002.8	340.0	200986.1	4167.6	333.0	200734.9	3887.3	338.4
DaIBA_RVRP_500_2	196760.4	3311.4	328.5	197012.5	3454.8	332.1	197090.2	3557.0	325.7	**196706.4**	3302.0	335.8
DaIBA_RVRP_1000_1	**277340.5**	4741.4	410.4	277943.3	5130.6	408.3	277830.5	5069.5	403.1	277661.3	4858.0	415.6
DaIBA_RVRP_1000_2	**276161.0**	4641.5	421.9	276601.6	4966.8	419.4	276711.3	4904.0	417.3	276783.8	4883.6	428.5

Table 3: Results of DaIBA, ESA, EA and FA for the proposed C-VRP-P*C.

4.3. Analysis and Discussion

If we analyze the results shown in Table 3, the main conclusion we could mention is that the proposed DaIBA performs better in terms of results quality. More concretely, DaIBA obtains better results than the ESA in the 91.6% of the cases (22 out of 24). On the other hand, it performs better than the EA in 95.8% of the instances (23 out of 24). Finally, the DaIBA outperforms the FA in 18 out of 24 datasets. These outcomes are

Name	Best Result	Vehicles	Method
DaIBA_RVRP_60_1_1	55381.71	2	DaIBA
DaIBA_RVRP_60_1_2	59837.12	3	DaIBA
DaIBA_RVRP_60_1_3	68716.43	2	DaIBA
DaIBA_RVRP_60_1_4	74816.11	3	DaIBA
DaIBA_RVRP_60_2_1	53048.08	2	DaIBA
DaIBA_RVRP_60_2_2	58200.78	3	DaIBA
DaIBA_RVRP_60_2_3	65876.55	2	DaIBA
DaIBA_RVRP_60_2_4	78434.97	3	EA
DaIBA_RVRP_90_1	80038.61	3	FA
DaIBA_RVRP_90_2	84442.16	4	ESA
DaIBA_RVRP_96_1	84973.01	3	DaIBA
DaIBA_RVRP_96_2	87738.90	4	DaIBA
DaIBA_RVRP_120_1	105882.18	5	DaIBA
DaIBA_RVRP_120_2	104363.12	4	DaIBA
DaIBA_RVRP_120_3	98139.38	3	DaIBA
DaIBA_RVRP_120_4	94391.90	4	EA
DaIBA_RVRP_200_1	151234.24	8	DaIBA
DaIBA_RVRP_200_2	148412.75	7	FA
DaIBA_RVRP_200_3	147527.18	7	FA
DaIBA_RVRP_200_4	145001.60	6	DaIBA
DaIBA_RVRP_500_1	197683.19	10	DaIBA
DaIBA_RVRP_500_2	194039.06	9	FA
DaIBA_RVRP_1000_3	273167.61	12	DaIBA
DaIBA_RVRP_1000_4	272500.43	11	DaIBA

Table 4: Best solutions found for each instance of the proposed benchmark.

coherent with the ones that we show in Table 4. In this table, we can see how DaIBA obtained the best solution in the 70% of the instances (17 out of 24).

Regarding this Table 4, we have depicted in Figure 4 the partial representation of the best solution found by the DaIBA in for the case DaIBA_RVRP_120_2. It should be highlighted that we have represented only partial solutions with the aim of facilitating the visibility of the reader. This way, this solution is depicted at cluster-level, showing also several clusters in detail.

In terms of computational effort, we can see in Table 3 how our proposal, EA and FA present similar runtimes. Meanwhile, ESA shows a better performance in this aspect, needing less time that its competitors. Analyzing this fact along with the quality of the obtained results, we can conclude that the DaIBA shows a better exploitation capacity than the EA and FA, because it reaches better outcomes in a similar time. On the other hand, we can say that the DaIBA has shown a better exploration capacity than the ESA, since it obtains better results needing more computational time.

Finally, an additional factor that should be mentioned is the robustness of the DaIBA. In this sense, we can observe in Table 3 that the standard deviation shown by our proposed method is lower than the ones presented by the other alternatives in all of the cases, meaning that the solutions provided by our technique are more stable. Logically, this is an important characteristic for a metaheuristic, giving reliability to the algorithm. This fact is specially appreciated in real environments.

4.4. Statistical Analysis of the Results

In addition to the above shown outcomes, and following the guidelines given in (Derrac et al. (2011)), we have performed two statistical tests with the obtained results. It should be highlighted that we have used the obtained averages to perform both tests.

Figure 4: Partial representation of the best solution found by the DaIBA for the instance DaIBA_RVRP_120_2. Source: Open Street Maps, via uMap, accessed March 2017.

First, we have used the Friedman's non-parametric test, in order to check whether there are significant differences among all the methods. The results of this test can be seen in Table 5. In this sense, the obtained Friedman statistic has been 30.45. Considering that the confidence interval has been stated at the 99%, the critical point in a χ^2 distribution with 3 degrees of freedom is 11.345. Since 32.3>11.345, we can say that there are significant differences among the obtained results, being DaIBA the one with the lower rank. Finally, regarding this Friedman's test, the resulting p-value has been 0.000001.

After obtaining the above described results, and with the aim of evaluating the statistical significance of the better performance of DaIBA, we have carried out the Holm's post-hoc test. Logically, we have taken the DaIBA as control algorithm. The unadjusted and adjusted p-values obtained through the application of Holm's post-hoc procedure are shown in Table 6. Analyzing these results, and taking into account that all the p-values are lower than 0.05, we can definitely conclude that the DaIBA is significantly better than ESA, EA and FA at a 95% confidence level.

Algorithm	Average Ranking
DaIBA	1.4583
ESA	3.000
EA	3.3333
FA	2.2083

Table 5: Average rankings obtained using the Friedman's test

Algorithm	Unadjusted p	Adjusted p
ESA	0.	0.000001
EA	0.000035	0.00007
FA	0.044171	0.044171

Table 6: Results obtained using the Holm's post-hoc procedure. DaIBA used as control algorithm.

Additionally, and in order to perform a more thoughtful analysis of the DaIBAs performance, we have conducted an additional experimentation using a fixed number of function evaluations as termination criterion. Furthermore, for this new experimentation, the convergence has also been analyzed, showing the average of evaluations needed to reach the best solution of each run. For this comparison, the FA has been used since (as can be seen in Table 5 and 6) it is the best method among the rest of the alternatives. These results have been summarized in Table 7. In this sense, it should be highlighted that the maximum number of function evaluations has been fixed in $n * Pop$, being n the size of the problem, and Pop the population of each algorithm.

Instance		DaIBA		FA	
Name	Max eval.	Avg.	Conv.	Avg.	Conv.
DaIBA_RVRP_60_1_1	6k	**55823.1**	5.63k	55986.8	5.73k
DaIBA_RVRP_60_1_2	6k	**60226.5**	5.34k	60627.1	5.50k
DaIBA_RVRP_60_1_3	6k	**69535.7**	4.61k	69692.0	5.12k
DaIBA_RVRP_60_1_4	6k	**76657.5**	4.12k	77222.0	4.57k
DaIBA_RVRP_60_2_1	6k	53678.0	5.71k	**53642.7**	5.64k
DaIBA_RVRP_60_2_2	6k	**59027.7**	5.48k	59506.6	5.37k
DaIBA_RVRP_60_2_3	6k	**67497.2**	4.98k	67690.0	5.25k
DaIBA_RVRP_60_2_4	6k	81802.0	4.48k	**81758.4**	4.76k
DaIBA_RVRP_90_1	9k	**83156.5**	8.69k	83298.8	8.64k
DaIBA_RVRP_90_2	9k	**86188.7**	8.32k	86300.4	8.58k
DaIBA_RVRP_96_1	9.6k	**86803.0**	9.30k	86921.1	9.34k
DaIBA_RVRP_96_2	9.6k	**90178.5**	9.11k	90431.0	9.28k
DaIBA_RVRP_120_1	12k	109194.2	10.94k	**109099.9**	11.13k
DaIBA_RVRP_120_2	12k	**107403.3**	10.87k	107608.8	10.81k
DaIBA_RVRP_120_3	12k	**102308.9**	10.12k	102688.0	10.76k
DaIBA_RVRP_120_4	12k	**98140.4**	9.70k	98370.4	10.22k
DaIBA_RVRP_200_1	20k	**153222.3**	19.42k	153410.8	19.34k
DaIBA_RVRP_200_2	20k	151397.8	19.04k	**151392.0**	19.42k
DaIBA_RVRP_200_3	20k	150801.7	18.59k	**150744.9**	18.99k
DaIBA_RVRP_200_4	20k	**147422.9**	18.74k	147789.8	18.60k
DaIBA_RVRP_500_1	50k	**200297.1**	48.52k	200783.7	48.43k
DaIBA_RVRP_500_2	50k	196782.3	48.41k	**196737.9**	48.67k
DaIBA_RVRP_1000_3	100k	**277386.7**	98.04k	277671.3	98.42k
DaIBA_RVRP_1000_4	100k	**276183.4**	99.01k	276790.7	98.94k

Table 7: Results and convergence of DaIBA, and FA with a maximum of $n * Pop$ function evaluation.

These results are also coherent with the ones presented in the previous sections. This way, DaIBA reaches better outcomes in the 75% of the datasets (18 out of 24). Additionally, in terms of convergence, the DaIBA also presents a better behavior, reaching the returned results needing less evaluations in the 66.6% of the datasets.

Furthermore, to prove the significance of these second experimentation, the Wilcoxon Signed-Rank test has been applied. The confidence interval has been stated at the 99% also for these tests. In this sense, regarding the difference in the results quality, the obtained Z-value is -3.6143, with a p-value of 0.0003. These results support the significance of the difference at 99% confidence level. Besides that, the obtained W-value has been 23.5. The critical value of W for N=24 at p 0.01 is 61. Therefore, the result is significant at this confidence level. In relation to the convergence behavior, the obtained Z-value has been -2.6857, with a p-value of 0.00714. Finally, the W-value get has been 56. These results also support the significance between the difference at the 99% confidence level.

5. Conclusions and Future Work

In the present paper, a medical goods distribution system with pharmacological waste collection was described and solved. This system has been modeled as a rich vehicle routing problem, concretely, as a clustered vehicle routing problem with pickups and deliveries, asymmetric variable costs and forbidden paths. As far as authors know, this is the first time that this specific problem is addressed in the scientific community. For this reason, a benchmark composed of 24 different instances has been developed, using real-world geographical locations of drugstores, hospitals and health centers. With the aim of tackling such a complex problem, a discrete and improved Bat Algorithm has been proposed. In line with this, this real-world use case can be considered as the first application of the Bat Algorithm to any rich vehicle routing problem. In order to prove the quality of the presented DaIBA, we have compared its performance with three other famous approaches: an evolutionary simulated annealing, an evolutionary algorithm and a firefly algorithm.

We have planned to extend the application of the proposed DaIBA to other real-world problems. Furthermore, a wider comparison with additional nature-inspired methods will be performed. At this stage, some of the techniques that we have planned to use in this future comparison are the Cuckoo Search, the Imperialist Competitive Algorithm, and the Harmony Search. Finally, we will investigate diverse improvements in the algorithm so as to see if the results shown in this work for the C-VRP-P*C can be improved. In this sense, we have planned the development of different heuristic operators for their use in the movement performed by the bats.

Compliance with Ethical Standards

Eneko Osaba declares that he has no conflict of interest. Javier del Ser declares that he has no conflict of interest. Xin-She Yang declares that he has no conflict of interest. Iztok Fister Jr. declares that he has no conflict of interest. Pedro Lopez-Garcia declares that he has no conflict of interest. Alejo J. Vazquez-Pardavila declares that he has no conflict of interest. This article does not contain any studies with human participants or animals performed by any of the authors.

Amorim, P., & Almada-Lobo, B. (2014). The impact of food perishability issues in the vehicle routing problem. *Computers & Industrial Engineering*, *67*, 223–233.

de Armas, J., & Melián-Batista, B. (2015). Variable neighborhood search for a dynamic rich vehicle routing problem with time windows. *Computers & Industrial Engineering*, *85*, 120–131.

Avci, M., & Topaloglu, S. (2016). A hybrid metaheuristic algorithm for heterogeneous vehicle routing problem with simultaneous pickup and delivery. *Expert Systems with Applications*, *53*, 160–171.

Bansal, J. C., Sharma, H., Jadon, S. S., & Clerc, M. (2014). Spider monkey optimization algorithm for numerical optimization. *Memetic computing*, *6*, 31–47.

Belmecheri, F., Prins, C., Yalaoui, F., & Amodeo, L. (2013). Particle swarm optimization algorithm for a vehicle routing problem with heterogeneous fleet, mixed backhauls, and time windows. *Journal of intelligent manufacturing*, *24*, 775–789.

Bortfeldt, A., Hahn, T., Männel, D., & Mönch, L. (2015). Hybrid algorithms for the vehicle routing problem with clustered backhauls and 3d loading constraints. *European Journal of Operational Research*, *243*, 82–96.

Caceres-Cruz, J., Arias, P., Guimarans, D., Riera, D., & Juan, A. A. (2015). Rich vehicle routing problem: Survey. *ACM Computing Surveys (CSUR)*, *47*, 32.

Chakri, A., Khelif, R., Benouaret, M., & Yang, X.-S. (2017). New directional bat algorithm for continuous optimization problems. *Expert Systems with Applications*, *69*, 159–175.

Chen, S.-M., & Chien, C.-Y. (2011). Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques. *Expert Systems with Applications*, *38*, 14439–14450.

Conesa-Muñoz, J., Pajares, G., & Ribeiro, A. (2016). Mix-opt: A new route operator for optimal coverage path planning for a fleet in an agricultural environment. *Expert Systems with Applications*, *54*, 364–378.

Defryn, C., & Sörensen, K. (2017). A fast two-level variable neighborhood search for the clustered vehicle routing problem. *Computers & Operations Research*, *83*, 78–94.

Derrac, J., García, S., Molina, D., & Herrera, F. (2011). 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*, 3–18.

Doerner, K. F., & Schmid, V. (2010). Survey: matheuristics for rich vehicle routing problems. In *International Workshop on Hybrid Metaheuristics* (pp. 206–221). Springer.

Expósito-Izquierdo, C., Rossi, A., & Sevaux, M. (2016). A two-level solution approach to solve the clustered capacitated vehicle routing problem. *Computers & Industrial Engineering*, *91*, 274–289.

Fister, I., Yang, X.-S., Fister, D., & Fister Jr, I. (2014). Firefly algorithm: A brief review of the expanding literature. In *Cuckoo Search and Firefly Algorithm* (pp. 347–360). Springer.

Fister Jr, I., Yang, X.-S., Fister, I., Brest, J., & Fister, D. (2013). A brief review of nature-inspired algorithms for optimization. *Elektrotehniški vestnik*, *80*, 116–122.

Golden, B. L., Raghavan, S., & Wasil, E. A. (2008). *The vehicle routing problem: latest advances and new challenges* volume 43. Springer Science & Business Media.

Gutierrez, A., Dieulle, L., Labadie, N., & Velasco, N. (2016). A multi population memetic algorithm for the vehicle routing problem with time windows and stochastic travel and service times. *IFAC-PapersOnLine*, *49*, 1204–1209.

Kaveh, A., & Talatahari, S. (2010). A novel heuristic optimization method: charged system search. *Acta Mechanica*, *213*, 267–289.

Leggieri, V., & Haouari, M. (2016). A matheuristic for the asymmetric capacitated vehicle routing problem. *Discrete Applied Mathematics*, .

Li, C.-L., Simchi-Levi, D., & Desrochers, M. (1992). On the distance constrained vehicle routing problem. *Operations research*, *40*, 790–799.

Li, M., Ma, J., Zhang, Y., Zhou, H., & Liu, J. (2015). Firefly algorithm solving multiple traveling salesman problem. *Journal of Computational and Theoretical Nanoscience*, *12*, 1277–1281.

Liu, R., Xie, X., Augusto, V., & Rodriguez, C. (2013). Heuristic algorithms for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care. *European Journal of Operational Research*, *230*, 475–486.

Liu, R., Xie, X., & Garaix, T. (2014). Hybridization of tabu search with feasible and infeasible local searches for periodic home health care logistics. *Omega*, *47*, 17–32.

Mahdavi, S., Shiri, M. E., & Rahnamayan, S. (2015). Metaheuristics in large-scale global continues optimization: A survey. *Information Sciences*, *295*, 407–428.

Mancini, S. (2016). A real-life multi depot multi period vehicle routing problem with a heterogeneous fleet: formulation and adaptive large neighborhood search based matheuristic. *Transportation Research Part C: Emerging Technologies*, *70*, 100–112.

Männel, D., & Bortfeldt, A. (2016). A hybrid algorithm for the vehicle routing problem with pickup and delivery and three-dimensional loading constraints. *European Journal of Operational Research*, *254*, 840–858.

Montané, F. A. T., & Galvao, R. D. (2006). A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers & Operations Research*, *33*, 595–619.

Moscato, P. et al. (1989). On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech concurrent computation program, C3P Report*, *826*, 1989.

Osaba, E., Carballedo, R., Yang, X.-S., & Diaz, F. (2016a). An evolutionary discrete firefly algorithm with novel operators for solving the vehicle routing problem with time windows. In *Nature-Inspired Computation in Engineering* (pp. 21–41). Springer.

Osaba, E., Yang, X.-S., Carballedo, R., & Lopez-Garcia, P. (2018). Good practice proposal for the implementation, presentation, and comparison of metaheuristics for solving routing problems. *Neurocomputing*, *271*.

Osaba, E., Yang, X.-S., Diaz, F., Lopez-Garcia, P., & Carballedo, R. (2016b). An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems. *Engineering Applications of Artificial Intelligence*, *48*, 59–71.

Ouaarab, A., Ahiod, B., & Yang, X.-S. (2014). Discrete cuckoo search algorithm for the travelling

salesman problem. *Neural Computing and Applications*, *24*, 1659–1669.
Parpinelli, R. S., & Lopes, H. S. (2011). New inspirations in swarm intelligence: a survey. *International Journal of Bio-Inspired Computation*, *3*, 1–16.
Penna, P. H. V., Afsar, H. M., Prins, C., & Prodhon, C. (2016). A hybrid iterative local search algorithm for the electric fleet size and mix vehicle routing problem with time windows and recharging stations. *IFAC-PapersOnLine*, *49*, 955–960.
Perez, J., Valdez, F., & Castillo, O. (2017a). Modification of the bat algorithm using type-2 fuzzy logic for dynamical parameter adaptation. In *Nature-Inspired Design of Hybrid Intelligent Systems* (pp. 343–355). Springer.
Perez, J., Valdez, F., Castillo, O., Melin, P., Gonzalez, C., & Martinez, G. (2017b). Interval type-2 fuzzy logic for dynamic parameter adaptation in the bat algorithm. *Soft Computing*, *21*, 667–685.
Pham, S., Gascon, V., & Dinh, T. (2014). Heuristics to solve a real-world asymmetric vehicle routing problem with side constraints. *Fundamenta Informaticae*, *130*, 231–246.
Precup, R.-E., Angelov, P., Costa, B. S. J., & Sayed-Mouchaweh, M. (2015). An overview on fault diagnosis and nature-inspired optimal control of industrial process applications. *Computers in Industry*, *74*, 75–94.
Precup, R.-E., David, R.-C., Petriu, E. M., Preitl, S., & Rădac, M.-B. (2014). Novel adaptive charged system search algorithm for optimal tuning of fuzzy controllers. *Expert Systems with Applications*, *41*, 1168–1175.
Rogdakis, I., Marinaki, M., Marinakis, Y., & Migdalas, A. (2017). An island memetic algorithm for real world vehicle routing problems. In *Operational Research in Business and Economics* (pp. 205–223). Springer.
Saji, Y., & Riffi, M. E. (2016). A novel discrete bat algorithm for solving the travelling salesman problem. *Neural Computing and Applications*, *27*, 1853–1866.
Shi, Y., Boudouh, T., & Grunder, O. (2017). A hybrid genetic algorithm for a home health care routing problem with time window and fuzzy demand. *Expert Systems with Applications*, *72*, 160–176.
Sicilia, J. A., Quemada, C., Royo, B., & Escuín, D. (2016). An optimization algorithm for solving the rich vehicle routing problem based on variable neighborhood search and tabu search metaheuristics. *Journal of Computational and Applied Mathematics*, *291*, 468–477.
Taha, A., Hachimi, M., & Moudden, A. (2015). Adapted bat algorithm for capacitated vehicle routing problem. *International Review on Computers and Software (IRECOS)*, *10*, 610–619.
Taha, A., Hachimi, M., & Moudden, A. (2017). A discrete bat algorithm for the vehicle routing problem with time windows. In *2017 International Colloquium on Logistics and Supply Chain Management* (pp. 65–70). IEEE.
Talarico, L., Springael, J., Sörensen, K., & Talarico, F. (2017). A large neighbourhood metaheuristic for the risk-constrained cash-in-transit vehicle routing problem. *Computers & Operations Research*, *78*, 547–556.
Tharakeshwar, T., Seetharamu, K., & Prasad, B. D. (2017). Multi-objective optimization using bat algorithm for shell and tube heat exchangers. *Applied Thermal Engineering*, *110*, 1029–1038.
Wang, J., Jagannathan, A. K. R., Zuo, X., & Murray, C. C. (2017). Two-layer simulated annealing and tabu search heuristics for a vehicle routing problem with cross docks and split deliveries. *Computers & Industrial Engineering*, *112*, 84–98.
Wang, P., Sanin, C., & Szczerbicki, E. (2015). Evolutionary algorithm and decisional dna for multiple travelling salesman problem. *Neurocomputing*, *150*, 50–57.
Wari, E., & Zhu, W. (2016). A survey on metaheuristics for optimization in food manufacturing industry. *Applied Soft Computing*, *46*, 328–343.
Yang, X.-S. (2009). Firefly algorithms for multimodal optimization. In *Stochastic algorithms: foundations and applications* (pp. 169–178). Springer.
Yang, X.-S. (2010a). Firefly algorithm, levy flights and global optimization. *Research and development in intelligent systems XXVI*, (pp. 209–218).
Yang, X.-S. (2010b). A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization* (pp. 65–74). Springer.
Yang, X.-S., & He, X. (2013). Bat algorithm: literature review and applications. *International Journal of Bio-Inspired Computation*, *5*, 141–149.
Yip, P. P., & Pao, Y.-H. (1995). Combinatorial optimization with use of guided evolutionary simulated annealing. *IEEE Transactions on Neural Networks*, *6*, 290–295.
Zhang, Z., Che, O., Cheang, B., Lim, A., & Qin, H. (2013). A memetic algorithm for the multiperiod vehicle routing problem with profit. *European Journal of Operational Research*, *229*, 573–584.
Zhou, Y., Luo, Q., Xie, J., & Zheng, H. (2016). A hybrid bat algorithm with path relinking for the

capacitated vehicle routing problem. In *Metaheuristics and Optimization in Civil Engineering* (pp. 255–276). Springer.