

Received October 9, 2021, accepted October 27, 2021, date of publication November 1, 2021, date of current version November 11, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3124710

Overview on Binary Optimization Using Swarm-Inspired Algorithms

MARIANA MACEDO¹, HUGO SIQUEIRA², ELLIACKIN FIGUEIREDO^{3,4},
CLODOMIR SANTANA¹, RODRIGO C. LIRA⁴, ANU GOKHALE⁵, (Senior Member, IEEE), AND
CARMELO BASTOS-FILHO⁴, (Senior Member, IEEE)

¹Computer Science Department, University of Exeter, Exeter, Devon EX4 4PY, U.K.

²Electrical Engineering Department, Federal University of Technology–Paraná, Ponta Grossa 84017-220, Brazil

³Innovation Laboratory of Information Technology (STI Labs), Federal University of Pernambuco (UFPE), Recife 50670-901, Brazil

⁴Computer Engineering Department, University of Pernambuco, Recife 50100-010, Brazil

⁵Department of Technology, Illinois State University, Normal, IL 61761, USA

Corresponding authors: Mariana Macedo (mgmm@comp.poli.br) and Hugo Siqueira (hugosiqueira@utfpr.edu.br)

This work was supported in part by the Fundação do Amparo a Ciência e Tecnologia (FACEPE) under Grant BCT-0070-3.04/17, Grant BCT-0066-1.03/17, and Grant IBPG-0964-3.04/16; in part by the University of Pernambuco (Programa de Fortalecimento Acadêmico (PFA)-UPE); in part by the National Council for Scientific and Technological (CNPq) under Grant 405580/2018-5 and Grant 315298/2020-0; in part by the Araucaria Foundation Process under Grant 51497; and in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

ABSTRACT Swarm Intelligence is applied to optimisation problems due to its robustness, scalability, generality, and flexibility. Based on simple rules, simple reactive agents - swarm (e.g. fish, bird, and ant) - directly or indirectly exchange information to find an optimal solution. Among multiple nature inspirations and versions, the dilemma of choosing proper swarm-based algorithms for each type of problem prevents their recurrent application. This scenario gets even more challenging when considering binary optimisation because of the absence of overview papers that assembles the trends, benefits and limitations of swarm-based techniques. Based on 403 scientific papers, we describe the basis of the leading binary swarm-based algorithms presenting their rationales, equations, pseudocodes, and descriptions of their applications to tackle this research gap. We also define a new classification based on the mechanism to update the solutions and the displacements, indicating that the Binary-Binary approach - binary decision variables and binary search space - is more efficient for binary optimisation in accuracy and computational cost.

INDEX TERMS Binary optimisation, swarm intelligence, metaheuristics, fitness function.

I. INTRODUCTION

Optimisation is an essential task for several fields such as Computer Science, Economy, Engineering, Bioinformatics and Operational Research [1]–[4]. Classical or brute force approaches that are represented by linear or non-linear programming suffer from the “curse of dimensionality” and require high knowledge of the problem (e.g. objective function, parameters, preconditions) [5]–[7]. Computational Intelligence (CI) arises as an alternative to solve the optimisation problems as it decreases the computational cost considerably, and it does not require a complete understanding of the problem.

The development of the Genetic Algorithm (GA) in the 1970s can be defined as the beginning of the optimisation using metaheuristics. The methods which use the

evolution paradigm, such as the GA or the Differential Evolution (DE), are classified as Evolutionary Computation algorithms [8], [9], and they can be mapped to binary or continuous problems [9]. Just in recent years, the binary optimisation - in which the variables are binary, as “on/off” or “selected/not selected” problems - gained considerable attention in the literature. The increase in the complexity of the current binary/discrete tasks leads to the necessity to develop strategies to deal with more complex optimisation problems [10]–[12].

In the early 1990s, another area of CI became recognised: Swarm Intelligence (SI). Similarly to Evolutionary Algorithms, SI is inspired by mechanisms from nature. The term *Swarm Intelligence* represents the “coordinated” behaviour that animals such as birds (Particle Swarm Optimization - PSO [13], [14]), bees (Artificial Bee Colony - ABC [15]), fish (Fish School Search - FSS [16]) or ants (Ant Colony Optimization - ACO [17]) emerge from the individual and

The associate editor coordinating the review of this manuscript and approving it for publication was Jamshid Aghaei.

collective movements. These groups of animals, or simple reactive agents, behave mainly without any supervision, and their actions have a stochastic component according to the perception of their neighbourhood [6], [18]. The *intelligence* of the swarm lies in the interactions agent-agent and agent-environment, using collective characteristics (e.g. decentralisation and self-organisation) [2], [19], [20]. Evolutionary computation is mainly based on competition and natural selection, while swarm intelligence relies mainly on the cooperation of the agents.

Most of the swarm-based algorithms were developed for continuous optimisation, but the demand for better algorithms for binary optimisation inspired researchers to adapt continuous algorithms to binary ones. The first version that we located in the literature of the binary PSO (BPSO) [10] use the same pseudocode of their continuous versions with some adjustments to deal with binary data. The first version located in the literature for the bees, Binary Artificial Bee Colony (BABC) [21], was inspired not only by the continuous version of ABC but also by this first BPSO algorithm. Until now, many binary algorithms still use mapping functions to transform continuous information to binary solutions during the iterations. A few papers use binary strings either in the solutions or their displacement. However, binary optimisation differs from continuous optimisation in several aspects, and the application of continuous methods and mechanisms to binary problems may not be the most efficient approach.

In order to understand better the swarm-based algorithms for binary optimisation, we have analysed a high number of papers proposed in the field. Our work points out the characteristics of the binary optimisation problem for swarm intelligence techniques, which is rarely discussed in the literature. Also, we argue about the advantages and disadvantages of considering particular algorithms, operators, and functions. We identified four main categories of methodologies applied to binary optimisation: (a) the use of genetic operators, (b) the use of mapping functions (or transfer functions), (c) the use of logic gates and (d) the use of similarity or information theory metrics. Then, we grouped the algorithms into three classes: Continuous-Continuous, Binary-Continuous and Binary-Binary, which represents the type of decision variables and the search space. Each class shares several methods and operators regardless of the inspiration of the algorithms.

Finally, the paper is organised as follows: Section II details the process of selecting and analysing the literature; Section III presents the general notation used in this investigation and the most used transfer functions to transform a continuous vector into a binary string; Section IV discusses a new classification proposal to group the binary swarm-based algorithms and presents the nine most prominent binary swarm algorithms before mentioned; Section V describes the most relevant applications found in the literature regarding each algorithm; in Section VI are the other nature-inspired algorithms found in the search, while Section VII and VIII present a discussion about some hybrid proposals and

multiobjective approaches; in Section IX we perform a discussion and point some future directions, and Section X shows the main conclusions.

II. SYSTEMATIC METHODOLOGY

We selected the most relevant articles regarding swarm-based algorithms for binary optimisation from five of the most important scientific platforms: IEEE Xplore [22], ACM [23], Springer [24], and Science Direct [25]. The search criteria were the keywords “binary” and “swarm”; the papers are from 2006 until the first half of 2020, and their *relevance* were ranked using the search engine of each electronic library. We applied an experimental analysis on the impact of using different strings on the search engine, and we noticed that including more keywords resulted in a very specialised set of papers. Therefore, as our goal was to understand the generality of the area and not specific patterns on a subtopic, we did not select other keywords.

It is important to remark that we also double-checked if some of the most important papers were presented also on the Google Scholar [26] that is well-recognised on ranking. We searched for the first 400 most relevant papers in the Google Scholar [26] and we selected extra 39 papers to our list because of their relevance to our paper. Most of the results presented in the Google Scholar were also selected on the other four databases, confirming that we could indeed identify some of the most relevant papers in the literature. We also point out that using the five platforms together, we were able to cover a high number of papers from the field.

In Table 1, we present the total number of analysed papers using the keywords: binary and swarm. Notice that all platforms presented a large number of articles, so we considered only the first 1,200 papers in each case. The exception was the IEEE Xplore library which we analysed the first 500 occurrences (as the following articles were too specific in a particular topic). The number of most relevant articles were not a priori selected; in fact, each paper was equally likely to be selected on our investigation. We analysed the papers based on five iterations of reading: (i) title, (ii) abstract, (iii) conclusions, (iv) methods and results and (v) entire paper. Finally, we selected the papers that we considered important to the literature based on historical (first proposals), generality (proposal not dependable from a specific problem) and novelty (outstanding performance and recent proposal) perspectives.

TABLE 1. Number of selected papers in each scientific database.

Database	IEEE	Science Direct	Springer	ACM	Google Scholar	Total
Visualised	1091	14047	5219	8105	28000	56462
Analysed	500	1200	1200	1200	400	3600
Selected	174	119	41	30	39	403

Figure 1 shows the overall number of papers published per year considering the 403 papers selected. The number

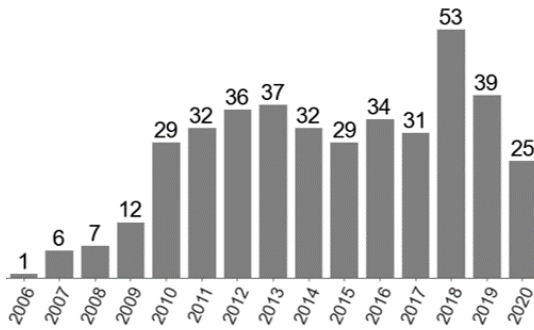


FIGURE 1. Number of selected papers according to their publication year.

of selected works in our investigation have been expressive since 2006. Almost half of the papers selected (233, or 58%) are from journals, while the others are from conferences (170). We identified 43 different swarm-based proposals which nine of them are well-known algorithms in Swarm Intelligence: Artificial Bee Colony (ABC), Ant Colony Optimisation (ACO), Bat Algorithm (BBA), Cat Swarm Optimisation (BCSO), Firefly Algorithm (FA), Flower Pollination Algorithm (FPA), Grey Wolf Optimiser (GWO), Gravitational Search Algorithm (GSA) and Particle Swarm Optimisation (PSO). A high number of papers addresses continuous benchmark functions as a case study. Most papers convert the continuous values into binary ones applying transfer functions (e.g. sigmoid and tangent), or mapping continuous values into bits [27], [28]. New algorithms are proposed in 237 papers, in which the authors usually present some modification or improvement in previous existing proposals. Additionally, 30 proposals was found for multi-objective approaches.

III. DEFINITIONS AND NOTATIONS

In this section, we present concepts, definitions, terminologies, and notations used throughout this paper. Table 2 summarises the most common notations. An agent (i.e. particle, bat, cat, etc.) has a vector with D elements, being each element correspondent to a variable in the problem domain. To exemplify this idea, lets consider the agent i , $\mathbf{x}_i = \{0, 1, 0, 0, 0, 1, 0, 1, 1, 1\}$ with $D = 10$. Therefore, the dimensions are $x_{i,1} = 0, x_{i,2} = 1, x_{i,3} = 0, \dots, x_{i,10} = 1$. This example is graphically presented in Figure 2:



FIGURE 2. Binary representation of the position of an agent in swarm-based algorithms.

It is very usual to apply continuous mathematical operations using the bits as real numbers. For example, consider Equation (1) [12], [29]:

$$\mathbf{z}_{ij} = \mathbf{x}_i - \mathbf{x}_j, \tag{1}$$

where $\mathbf{x}_i = \{1, 1, 0, 0\}$ and $\mathbf{x}_j = \{0, 1, 0, 1\}$. The resulting array is the subtraction of the value in each

TABLE 2. Representation of equations and formulas.

Notation	Definition
N	Size of the population
t	Current iteration
D	Number of dimensions
$d = 1, 2, 3, \dots, D$	Index of each dimension
$\vec{x}_i = \{x_{i,1}, \dots, x_{i,D}\}$	Agent i considering its D dimensions
$\vec{z}_i = \{z_{i,1}, \dots, z_{i,D}\}$	Agent i considering its D dimensions represented by continuous values
$fit(\mathbf{x}_i)$	Fitness function of the agent i
\mathbf{gbest}^t	Best agent until iteration t
$rand(\alpha, \beta)$	Random continuous value drawn from an uniform distribution between the limits α and β
$randbin(0, 1)$	Random binary value generated using an uniform distribution probability

corresponding dimension, generating $\mathbf{z}_{ij} = \{1, 0, 0, -1\}$; the elements of \mathbf{z} are assumed to be real numbers. Most of the binary swarm-based algorithms (but not all of them) use, during the intermediate steps, continuous information to update the displacement of an agent or its next position. Sometimes, it is also calculated as a continuous solution in the literature. In both cases, it is necessary to transform a continuous vector into a binary string before the evaluation of the fitness [30], [31]. There are some possibilities to transform (or map) a continuous array with D dimensions into a binary string with the same size. Often, the mapping process is performed in two steps. The first one is the application of the value of each dimension D into a nonlinear function. Then, the number achieved is used as a probability to determine if the respective binary dimension d is “0” or “1”; the conversion is done by some conditional criterion [10], [32]. This first proposal is the most common in the literature. Considering a general continuous (real) vector \mathbf{z}_i^{cont} [33], [34], for each dimension $z_{i,d}^{cont}$, it is possible to apply a sigmoid function S , according to (2):

$$S(z_{i,d}^{cont}) = \frac{1}{1 + e^{-z_{i,d}^{cont}}}. \tag{2}$$

Then, the value found $S(z_{i,d}^{cont})$ is used as a probability to define the bit in the respective dimension d , creating a binary element $x_{i,d}^{bin}$ of a vector as described in (3) [10], [35]:

$$x_{i,d}^{bin} = \begin{cases} 1 & \text{if } rand(0, 1) < S(z_{i,d}^{cont}) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

We can modify (2) by adding a constant b generating $S(bz_{i,d}^{cont})$ [7]. If $b = 1$, the equation stands as presented before, otherwise b changes the format of the function.

Figure 3A shows the graphic behaviour of a sigmoid function. Because of its format, it is usually called as S-shaped [30].

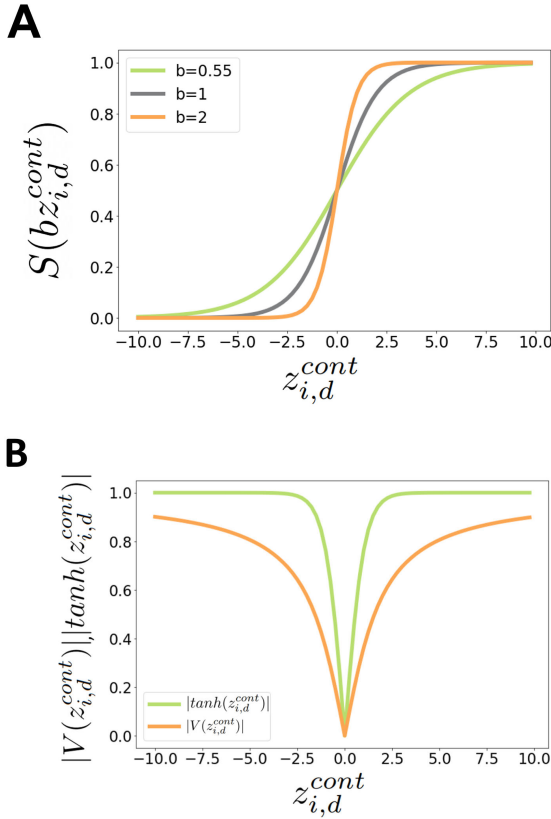


FIGURE 3. (A) Sigmoid and (B) V-Shaped transfer functions.

A different approach is to use “V-shaped” functions [30]. The first proposal is the utilisation of the modulus of the hyperbolic tangent in the same way as in the previous case, generating $|\tanh(z_{i,d}^{cont})|$. Equation 4 describes the mathematical formulation of this proposal [36], [37]:

$$|\tanh(z_{i,d}^{cont})| = \left| \frac{e^{-z_{i,d}^{cont}} - 1}{e^{-z_{i,d}^{cont}} + 1} \right| \quad (4)$$

Alternatively, other transfer function is addressed in (5) [38]–[40]:

$$|V(z_{i,d}^{cont})| = \left| \frac{2}{\pi} \arctan \left(\frac{2}{\pi} z_{i,d}^{cont} \right) \right| \quad (5)$$

We can resort (5) to perform the mapping of the continuous vector $z_{i,d}^{cont,t}$ in each dimension to generate $x_{i,d}^{bin}$ as in (6) [30], [38]:

$$x_{i,d}^{bin,t+1} = \begin{cases} \bar{x}_{i,d}^{bin,t} & \text{if } \text{rand}(0, 1) < |\tanh(z_{i,d}^{cont,t})| \\ x_{i,d}^{bin,t} & \text{otherwise} \end{cases} \quad (6)$$

where $\bar{x}_{i,d}^{bin}$ means that the current state of the variable d must be flipped and $x_{i,d}^{bin}$ means that the current bit has to be maintained in the next iteration. Note that in this case $|\tanh(z_{i,d}^{cont,t})|$ can be replaced by $|V(z_{i,d}^{cont,t})|$. Figure 3B shows

the graphics of the most used V-shaped functions. Observe the difference of both proposals.

We highlight that there are other mapping methods. Dahi *et al.* [41] describe mapping methods using nearest-integer, normalisation, angle modulation and search process. Mirjalili and Lewis [30] discuss the impact of S-Shaped and V-Shaped functions in the BPSO. In the following sections, we describe other proposed mapping methods which are used a few times in specific swarm-based methods. We highlight that one transfer function is not necessarily better for all types of swarm-based algorithms.

IV. ALGORITHMS

Figure 4 depicts the most frequent algorithms mentioned by the selected papers and, as we can see, the most prominent names (calculated accordingly to the number of papers that mentioned a particular algorithm) are mostly related to the PSO and some of its variations. Besides the PSO-based algorithms, ABC, ACO, GSA and BA are other prominent algorithms in Figure 4.

All the versions of the swarm-based algorithms in this section can be divided into four main categories based on their similar methodologies and operators:

- Use of genetic operators - the beginning of the natural algorithms for optimisation started considering the evolutionary approaches, mainly the Genetic Algorithm, which uses genetic operators (as crossover and mutation) [3], [9]. The use of genetic operators can be found in the algorithms as a kind of hybridisation;
- Conversion of decision variables from continuous to binary values - the majority of binary swarm-based algorithms was adapted from continuous versions. Then, the researches created adaptations to deal directly with binary decision variables, usually using some mapping function, like those described in Section III, like S-Shaped [1];
- Use of logic gates - all the variables and agents are in the binary domain and the operations consider the logic gates such as AND, OR, XOR and NOR [2];
- Use of similarity or information theory metrics - recently, a new trend appeared in the literature, the use of measurements of binary structure similarities, like the Jaccard’s coefficient [42]. Other metrics from information theory can also be used such as entropy [43].

The 403 analysed papers showed that almost all papers are included in the items: b) and c). Looking at the similar patterns from the most popular methodologies and operators in swarm-based algorithms, we introduce a new proposal to classify binary swarm-based algorithms using the representation of the decision variables and the search space. We observe three groups that summarise the typical patterns of swarm-based techniques for binary optimisation:

1) Binary-Continuous

The candidate solutions in this Binary-Continuous class are binary vectors from the start until the end of the iterations. However, the displacement (e.g. the velocity



FIGURE 4. Wordcloud of the swarm-based algorithms mentioned in the selected papers. Note that the size of the word is proportional to the frequency that it appears in the papers (i.e. the number of papers that mentions the algorithm). Therefore, PSO and BPSO appear more frequently in the majority of the papers.

in the PSO) is calculated and updated as a continuous array before acts in the agent. The information of the displacement vector is converted into binary numbers commonly using a transfer function, like the functions described in Section III.

2) Continuous-Continuous

The Continuous-Continuous class is the closest of the original versions of the algorithms, which were initially developed to solve continuous problems. In this case, all the steps are performed in the continuous (real) space. Therefore, the candidate’s solutions, as well as the displacement process, are continuous vectors. The conversion to binary occurs by applying some mapping process to the agents to evaluate their fitness. Interestingly, this proposal is rarely found.

3) Binary-Binary

In the Binary-Binary approach, the algorithms operate in the binary space. The candidate’s solutions and the displacement are binary vectors, and the interactions between the swarm occur using binary methods such as the logic gates. Consequently, there is no need to use transfer functions. Therefore, the algorithms in this class are usually more efficient and less computationally costly than the others because they treat the problem in the proper binary search space using binary decision variables.

Algorithms inside each group share similar patterns also concerning computational complexity and cost. This is the case because their mechanisms usually have common levels

of memory consumption and a number of methods and functions. In Swarm Intelligence, we argue that a fair comparison across algorithms from different inspirations requires the comparison to be based on fitness evaluation, number of agents and complexity of any called function and method.

For binary optimisation, there is more development in Swarm Intelligence on the Binary-Continuous approach. In the following sections, according to our classification rule, we describe the core insights of the nine most popular swarm-based algorithms applied for binary problems: Artificial Bee Colony (BABC), Ant Colony Optimisation (BACO), Bat Algorithm (BBA), Cat Swarm Optimisation (BCSO), Firefly Algorithm (BFA), Flower Pollination Algorithm (BFPA), Gravitational Search Algorithm (BGSA), Grey Wolf Optimiser (GWO), and Particle Swarm Optimisation (BPSO). The binary algorithms inspired by the behaviour of birds (PSO) and bees (ABC) have versions on all three categories. The binary versions inspired in the behaviour of cats (CSO) present versions on Binary-Continuous and Binary-Binary versions. The binary versions inspired on the movement of ants (ACO), bats (BA), wolves (GWO) and planets (GSA) present versions only in the Binary-Continuous category. Firefly (FA) and Flower Pollination (BFPA) are only adapted to Continuous-Continuous category.

A. CONTINUOUS-CONTINUOUS ALGORITHMS

Based on the original swarm-based algorithms, several binary versions were proposed mapping the positions using a transfer function of the swarm from a continuous to a

binary search space. Four different inspired algorithms have continuous-continuous algorithms for binary optimization:

- 1) Binary Artificial Bee Colony (BABC) [44] (explained in Appendix A-A);
- 2) Binary Firefly algorithm (BFA) [6], [45] (explained in Section A-B);
- 3) Binary Flower Pollination algorithm (BFPA) [41], [46] (explained in Section A-C);
- 4) Binary Particle Swarm Optimization (BPSO) [41], [47]–[50] (explained in Section A-D).

Rashedi *et al.* [51] and Mirjalili *et al.* [52] argue that some aspects should be considered when applying a transfer function to swarm-based techniques:

- The values mapped by the transfer function should range between 0 and 1.
- Small values (closer to 0) from the transfer function represent small probabilities of changing the position, or few changes on the position.
- High values (closer to 1) from the transfer function represent high probabilities of changing the position, or several changes on the position.
- The algorithm should have a mechanism to control how the transfer function influences the swarm based on the aforementioned points.

Even though it is a straightforward map between the two search spaces, it does not account for all the characteristics of binary optimization problems such as the emergence of agents (e.g. fish, bee, particle) with similar positions, which rarely happens in continuous optimization. In this way, the transfer function can indeed influence the emergence of several agents with the same position and premature convergence.

It seems that for some binary problems such as Feature Selection that the goal is to minimise the number of selected features, the transfer function can be biased to select more 0s than 1s [53], so the swarm-based algorithm chooses quickly the best feature vector with a small number of 1s.

As, in this class, it is common to use most of the methods similar to the ones created by the continuous versions, the core strategies of the algorithms are not changed. Therefore, the new versions here focused on finding the best step/way to map the continuous solution to a binary one. Sigmoid functions are widely applied for any swarm-based algorithm, but the V-shaped function seems to have better results for the version of Flower Pollination Algorithm, for example.

B. BINARY-CONTINUOUS ALGORITHMS

For the Binary-Continuous approach, the swarm-based algorithms continue to apply a transfer function to map continuous vectors to binary vectors, but the agents are represented by binary vectors (position). The majority of the mechanisms are also maintained from the original versions, but the operations may change depending on the version.

In this category, most swarm-based algorithms have a binary version such as the following:

- 1) Binary Artificial Bee Colony (BABC) [36], [54] (explained in Appendix B-A);
- 2) Binary Ant Colony Optimization (BACO) [55]–[57], [57]–[61] (explained in SectB-A1);
- 3) Binary Bat algorithm (BBA) [30], [39], [62]–[65] (explained in Section B-A2);
- 4) Binary Cat Swarm Optimization (BCSO) [28], [66]–[69] (explained in Section B-A3);
- 5) Binary Gravitational search algorithm (BGSA) [37], [51], [70]–[73] (explained in Section B-A4);
- 6) Grey Wolf Optimizer (GWO) [7], [74]–[79] (explained in Section B-A5);
- 7) Binary Particle Swarm Optimization (BPSO) [31], [38], [40], [74], [80]–[96] (explained in Section A-D).

As the key idea in swarm intelligence is to update a new position of an agent considering a previous position, a collective position or a neighbour position, the algorithms provide adaptations for the original equations to incorporate binary solutions. One adaptation commonly used is a transfer function (e.g. S and V-shaped) that maps a value of a continuous vector directly to a binary solution (such as the velocity vector in PSO). Other methods are the use of mathematical operations to create a binary solution such as frequency, average and probability of values. For example, an agent will flip a dimension to be similar to the majority of the swarm because more than half of the swarm have the dimension in a particular direction (0 or 1). Some algorithms also map each dimension 0 or 1 to a continuous value to apply a continuous mechanism and then return to a binary solution after the mechanism is performed. We see that even though the versions find optimal solutions for a set of binary problems, the excess of mapping values from continuous to binary and binary to continuous might not be the most efficient alternative.

C. BINARY-BINARY ALGORITHMS

Binary-binary algorithms are the only class in which the proposals are made for binary optimization as their mechanisms and operators, binary decision variables and binary search space. In this category, the proposals usually use logic gates, genetic operators (e.g. crossover and mutation) and binary stochastic processes. We found three kind of swarm-based algorithms in this category:

- 1) Binary Artificial Bee Colony (BABC) [54], [97]–[100] (explained in Appendix C-A);
- 2) Binary Cat Swarm Optimization (BCSO) [101], [102] (explained in Appendix C-A1);
- 3) Binary Particle Swarm Optimization (BPSO) [103]–[108] (explained in Appendix C-A2);

Taking into account the computational cost, Binary-Binary algorithms seems to be much faster than the other categories because it considers a much smaller search space. Small changes on the continuous search space can represent no change on a projected binary search space, causing unnecessary search. Moreover, binary methods such as Boolean operators and crossover are usually less computationally costly than methods that account for continuous search. Thus,

binary methods, search space and storage usually requires less memory and operations.

In this class, we also see that the operators follow different operations than the original ones, but the inspiration and goal of these strategies maintain the same. For example, the update of velocities and displacements will continue to provide convergence depending on the collective information at the moment. Moreover, mechanisms of local search will be adapted to wiser random changes depending on the individual or social information.

V. MAIN APPLICATIONS

In this section, we describe the main applications found in the literature. The results presented in Figure 5 shows the diversity of the problems used to assess and validate the performance of the algorithms. It is worth mentioning that the benchmark functions and the feature selection problem were the most common applications. PSO displays the most extensive set of applications, which covers most of the problems present in the literature.

A. APPLICATIONS OF BABC

- Wang *et al.* [109] proposes a hybrid approach using BABC and SVM to improve Intrusion Detection Systems (IDSs). The results showed that the proposal was able to overcome the results found by BPSO and GA. Similarly, Wei *et al.* [54] showed that BABC could overcome the BPSO and BGA to optimise benchmark functions.
- Pampara and Engelbrecht [21] developed the Angle Modulated ABC (AMABC). They showed that the algorithm could overcome the Angle Modulated PSO (AMPPO), Angle-Modulated DE (AMDE), binABC, normABC to search the optimal points in benchmark functions.
- A Novel BABC algorithm (disABC) was developed by Kashan *et al.*, in which the authors evoke the concept of dissimilarity to generate new solutions [12]. They applied the algorithm in the uncapacitated facility location problem (in 15 benchmark instances) and showed that the disABC could be better than BinDE and PSO.
- Jia *et al.* introduced the BABC using Bitwise Operation (BitABC) [98]. In this work, because of the binary nature of the variables, they suggest the use of binary operations as those performed by logic gates. The proposal was the best in comparative to DisABC, normABC, BinABC to optimise benchmark functions.
- Ozturk *et al.* proposed the use of evolutionary mechanisms, like the crossover, to improve the BABC, creating the BABC based on Genetic Operators (GB-ABC) [110]. The proposal overcame GA, PSO based dynamic image clustering (DCPSO) and a binary ABC model (DisABC) to solve dynamic clustering problems. The same authors optimised three problems: dynamic image clustering, 0/1 knapsack problems and benchmark functions from CEC2005 using

the Improved DisABC (IDisABC), which was proposed in this study [3]. The new approach overcame the GB-ABC, DisABC, Quantum Inspired BPSO (QBPSO), BPSO and GA.

- In the paper from Hancer *et al.* [5], it is introduced the Modified Discrete ABC (MDisABC) integrating evolutionary-based similarity search mechanisms to the DisABC. The case study was to optimise benchmark functions, tasks in which MDisABC performed better than BPSO, New Velocity Based BPSO, Quantum Inspired BPSO, Discrete ABC, Modification Rate Based ABC, Angle Modulated ABC, and GA.
- Kiran and Gunduz presented the first BABC based on Boolean gates, the binABC [97]. They solve some benchmark problems, and the algorithm overcame BPSO, IBPSO and DISABC.
- Lu *et al.* [36] introduced Binary-Real Coded Bee Colony Optimization (IB-RBCO) to solve short-term hydro generation scheduling, achieving better results than BCO, IBPSO, and DE.
- Ozger *et al.* [111] presented eight versions of the BABC, being the bitABC the best one to perform feature selection in benchmark functions. They compare the final results with those achieved by BABC, XSABC, XBABC, among others.
- The work from He *et al.* introduced a Novel BABC algorithm to solve Set-Union Knapsack Problem (SUKP) [44]. The proposal was the best in comparative to ABCbin, DEbin, GA and A-SUKP algorithms, and the unique paper presenting the real encoding to all the operations during the search process.
- Zhang and Zhang used the BABC to construct spanning trees in vehicular ad hoc networks [112]. Despite using more computational time, BABC found similar results than Kruskal, a classical algorithm. Moreover, BABC could also produce candidate suboptimal spanning trees that could be useful when some nodes become unavailable.
- The work from Shunmugapriya [18] addresses a feature selection using benchmark functions as a case study to evaluate the hybrid BABC and ACO algorithm (AC-ABC). The new proposal overcame the ACO, ABC, ABC-DE, PSO, CatFish Binary PSO.
- In the article from Xu *et al.*, they solve a two-level distribution optimisation problem using a multiobjective approach named BEES-Binary Bees Algorithm (BBA), which adopt the concept of Pareto dominance to develop the search [113].

B. APPLICATIONS OF BACO

- The paper from Fernandes *et al.* [114] was the oldest we found about BACO, published in 2007. The basis of the algorithm is present in the paper, even though they called the proposal as Binary Ant Algorithm (BAA). The update in the pheromone is a little different from

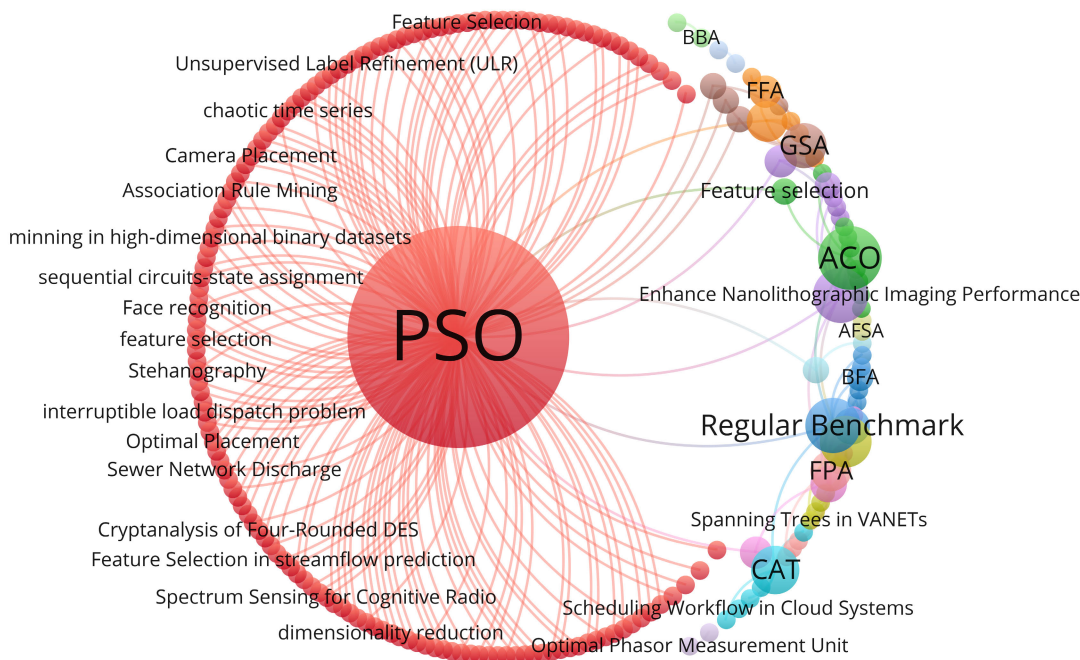


FIGURE 5. Network representation of the swarm-based algorithms applied to different problems. In this network, the nodes represent problems or algorithms, and the links indicate that an algorithm was assessed in a given problem. The node size is proportional to the number of connections it has, and the colours are a result of a clustering process. Note that the number of small red nodes correspond to a variety of problems in which only the PSO was applied. Moreover, we can see that continuous benchmark functions and feature selection are the most used problems.

those described in Section B-A1. They compared the algorithm with the GA using benchmark functions.

- Zhao and Yan [115] introduce the Bottleneck Assigned Binary Ant System (baBAS), based on the traffic organisation phenomenon in ants swarm, under a high level of crowded conditions. The proposal achieved better performance than Binary Ant System (BAS), Binary Ant System with Elitist Strategy (BASE) and Niche GA in benchmark problems.
- Kuo [61] use the BACO with a hypercube framework in pixelated source optimization for improving lithographic resolution. The numerical simulation showed that BACO effectively searches the optimal source shape in the problem.
- Wei *et al.* present the Adaptive and Commutative BACO (ACBACO) to optimise binary benchmark functions. The performance of the method was better than ACO, Modified ACO (MACO), Adaptive Random Search Technique (ARSET), Successive Zooming GA (SZGA) and ACO by Reduced Search Space (ACORSES) [58].
- The paper by Wang *et al.* [116] introduces a Hybrid Mutation-based Binary Ant Colony optimisation (MBACO) to solve facility-location problems. The algorithms BPSO, GA and Tabu Search, were used to compare the methods.
- Kashef *et al.* present the Advanced Binary ACO (ABACO) [117]. In a feature selection perspective of a binary benchmark problem, the algorithm overcame

the original BACO. On the other hand, Resendiz *et al.* applied the novel BACO (NBACO) in the application of the Mahalanobis–Taguchi System to a dimensional problem in the automotive industry [118], achieving better results than the BPSO.

- Changdar *et al.* [119] solved the 0/1 knapsack problem under fuzziness using a dedicated BACO, specially developed to this kind of problem, which was better than the BPSO, Modified BPSO, Genotype – Phenotype Modified Binary Particle Swarm Optimisation (GPMBPSO). This proposal is also inspired by genetic methods such as mutation and crossover.
- Sreeja and Sankar [59] introduces the BACO Search Point Pattern Matching Algorithm (ACOBSPPM) to point pattern-matching task. Some algorithms specifically developed for this application were addressed, like Fast Expected Time algorithm (FET), Approximate Input Sensitive Algorithm and binary search method. The final result was favourable to the BACO-based method.
- Wang *et al.* propose the Modified Coded ACO algorithm combined with GA (MBACO) to define the parameters of a Support Vector Machine (SVM) for remote sensing image classification [60]. The comparison was made with the GA, BPSO, BACO, BDE, and Binary-Coded Cuckoo Search (BCS).
- The work of Wan *et al.* [120] is the only that uses the η explicitly, employing hybridisation with the GA. They proposed some variations called Modified Binary

Coded Ant Colony Optimization Algorithm (MBACO), Visibility Density MBACO and Pheromone Density MBACO. The approaches were performed using binary benchmark problems and overcame GA, efficient GA (EGA), improved GA(IGA), BACO, Advanced BACO(ABACO), BPSO, BDE and a hybrid GA-ACO.

- Qing *et al.* proposed the Multi-Population Binary Ant Colony Algorithm with Concrete Behaviours (MPBACB) to deal with multi-objective problems [57]. They applied the algorithm in some benchmark functions and compared the achieved solutions with the NSGA-II and Ant Colony Algorithm for solving multi-objective problems.
- Zangari *et al.* proposed the Improved Decomposition-based Multi-objective BACO (MOEA/D-ACO) [121] to solve the multi-objective unconstrained binary quadratic programming (mUBQP). The result shows that the proposal can be better than the traditional multi-objective evolutionary algorithm based on decomposition (MOEA/D).

C. APPLICATIONS OF BBA

- Kaur *et al.* [122] address the BBA together with the DWT-SVD model to enhance the quality of underwater images. The simulation showed that the hybrid approach using BBA had overcome the existing technique when compared using various parameters, such as Bit Error Rate, Entropy, and Normalized Cross Correlation.
- Gupta *et al.* address the algorithm to perform feature selection to classify white blood cells using the KNN [123]. The results showed that the Optimized Binary Bat Algorithm proposed outperformed Optimized Cuttlefish Algorithm (OCFA), and Optimized Crow Search Algorithm (OCSA) in efficiency and accuracy.
- Mokhov *et al.* [124] created a mathematical formulation of the problem called “on space flights” from the film “Planet Ant: Life Inside the Colony (2012).” In summary, it is a discrete optimisation task that consists of finding the best spacecraft’s route between two planets. The results showed that BBA presented better results than the BPSO.
- Basetti *et al.* [63] suggest the application of the Taguchi method to perform the initial population of the bats. They give the acronym TBBA to the hybrid proposal. The application was the optimal PMU placement for the power system.
- Amine *et al.* [39] proposed a multi-objective bat algorithm (MBBA) using the concept of Pareto Dominance. They applied the algorithm to optimise benchmark functions, and the results overcame those achieved by NSGA-II.

D. APPLICATIONS OF THE BCSO

- Sharafi *et al.* [28] were one of the first to propose a binary version of the CSO algorithm, which they named

Discrete Binary CSO. They apply the BCSO to solve some instances of 0/1 knapsack problem and benchmark functions. The results were compared to the GA and two different versions of the BPSO.

- In the work of Mohamadeen *et al.* [66], the authors utilise the BCSO to define the parameters of an SVM. The goal was to select the tests that can be employed to classify transformer health indexes. The results were successfully compared to those achieved by the BPSO.
- In the same way, Srivastava and Maheswarapu applied the BCSO to solve the optimal PMU placement problem [69]. The algorithm presented better results than BPSO, Generalized Integer Linear Programming and Effective Data Structure Based Strategies.
- The paper developed by Li *et al.* [68] addresses the BCSO to the antenna selection problem. The authors analysed the algorithm changing the number of iterations, and agents (cats) in the swarm. Simulations showed that, for large scale MIMO systems, BCSO shows advantage in the antenna selection problem.
- Kumar *et al.* [67] solved scheduling workflow applications in cloud systems using the application of a Discrete Binary Cat Swarm Optimization (DBCOS) that they introduced. Their proposal achieved good results in comparison to BPSO.
- Pappula and Ghosh utilised the BCSO to planar thinned antenna array synthesis. In this case, they present a multi-objective version (MOBCSO), using the Pareto dominance criteria. The results were better than those obtained with the multi-objective BPSO (MBPSO) [125].
- Siqueira *et al.* [101] proposed a version of a Binary-Binary BCSO, the Boolean BCSO. In this algorithm, the authors compared the performance of the method with the BPSO, GA and the BCSO in the 0/1 knapsack problem. In 2020, Siqueira *et al.* [102] proposed the Simplified version of BCSO that overcame other swarm-based algorithms in the One max, Subset sum, 0/1 Knapsack, Multiple Knapsack and Feature Selection problems.
- A BCSO algorithm was also applied for manufacturing cell design problem [126] that optimises the transportation of manufactured parts between cells. Soto *et al.* include the Autonomous Search algorithm into the BCSO, improving the fitness on the middle-to-late stages of the simulation.

E. APPLICATIONS OF BFA

- Chandrasekaran *et al.* [6] address a unit commitment (UC) problem, achieving better results than PSO and GA, among others. Note that in this paper, the calculation of the parameter vector r_{ij} is specific to solve this problem.
- Liu *et al.* [45] solve a spectrum allocation optimisation for cognitive radio networks (CRN). The computational results showed that using network reward or average

user's fairness as fitness, the proposal performed better than BPSO and GA in the spectrum allocation problem.

- Almonacid *et al.* solved a manufacturing cell design problem (MDCP) using the BFA. The comparative analysis considers lots of other metaheuristics like Artificial Fish Swarm Algorithm (ASFA), Binary Bat Algorithm (BBA), Migrating Birds Optimization (MBO), Shuffle Frog Leaping Algorithm (SFLA), Invasive Weed Optimisation (IWO), Differential Evolution, Simulated Annealing, PSO and Egyptian Vulture Optimization Algorithm (EVOA) [127].

F. APPLICATIONS OF BFPA

- Rodrigues *et al.* [46] proposed the utilisation of it in an electroencephalogram (EEG)-based person identification process. The results were favourable to the BFPA in comparison to BGA, BPSO, Binary Firefly Algorithm (BFA), Binary Charged System Search and Binary Harmony Search (BHS).
- Dahi *et al.* [41] use the BFPA on the antenna positioning problem, presenting some improvements in the original proposal. They achieved better results in comparison to Population-based Incremental Learning (PBIL) and Differential Evolution (DE).
- Sayed *et al.* [128] introduce a hybrid methodology that uses the BFPA and the Clonal Selection Algorithm (CSA) to solve feature selection problems. The problem addressed was to optimise benchmark functions. The new algorithm overcame the Binary Cuckoo Search Algorithm (BCSA), Binary Bat Algorithm (BBA), Binary Differential Evolution (DE) and the standard Binary Flower Pollination Algorithm (BFPA).
- Shilaja and Ravi optimised the emission/economic dispatch in solar photovoltaic generation utilising the new Euclidean Affine Flower Pollination Algorithm (eFPA) and BFPA [129]. The final results were favourable to this proposal in comparative to PSO and FPA.

G. APPLICATIONS OF BGSA

- The work from Ji *et al.* [98] proposes an improvement in the algorithm using a quantum-inspired computing concept. This numerical computational method addresses the principle of quantum mechanics, creating the Quantum-inspired BGSA (QIBGSA). They solve an instance of the thermal unit commitment with wind power integration problem. The authors show that the QIBGSA can overcome the BPSO and BGA.
- The paper developed by Nezamabadi-Pour presented another proposal of Quantum-inspired BGSA. He addressed three different applications: 0/1 knapsack problem, Max-ones problem and the optimisation of Royal-Road functions. The novel proposal was superior to standard BGSA, GA and three versions of Quantum-Inspired Evolutionary Algorithms (QIEAs) [37].

- Barani *et al.* [71] introduced the Improved Binary Quantum-Inspired Gravitational Search Algorithm (QGSA-UC) to solve a unit commitment problem. The results obtained showed that the method could be better than GA, Evolution Programming (EP), Differential Evolution (DE), Simulated Annealing (SA) and some versions of BPSO.
- Chakraborti and Chatterjee [72] introduced a Binary Adaptive Weight GSA (BAW-GSA). The application addressed was a feature selection for face recognition. The BAW-GSA was a better optimiser than BPSO and BGA in this case.
- Rouhi and Nezamabadi-pour investigated the use of an improved version of the BGSA to perform feature selection in 5 high-dimensional microarray databases. The classifier addressed is the KNN [70].
- The work from Khanesar and Branson III [130] proposed the XOR BGSA to improve the capability of the method to solve instances of the knapsack problem and to optimise benchmark functions. It was shown that the proposal results overcome BGSA, NBPSO, IBPSO, and BPSO.

H. APPLICATIONS OF BGWO

- Emary *et al.* [7] and Hu *et al.* [76] solved feature selection using UCI datasets. Chantar *et al.* [131] employed BGWO to enhance a wrapper-based feature selection technique for Arabic text classification. The results showed that GWO-based process using elite-based crossover approach (55) revealed a superior performance in the problem. Lastly, modified binary grey wolf optimizer (MBGWO) was proposed by Alzubi *et al.* [77] to choose relevant features to intrusion detection system problem. The results showed that the proposal added significantly enhanced the performance of the IDS.
- Luo *et al.* [78] introduced a new BGWO proposal to solve the multidimensional knapsack problem. Compared with BFOA, Hybrid Harmony Search (HHS), Quantum PSO (QPSO) and bGWO-o, the proposal obtained the best performance on the two benchmark problems. Also, the results showed that the V-shaped function could achieve better results in the problem.
- A quantum-inspired binary grey wolf optimizer (QI-BGWO) was proposed by Srikanth *et al.* [79] to solve unit commitment problem. Statistical tests are performed to show the superior performance of the proposal.
- Jiang *et al.* [75] proposed the Improved Binary GWO to solve the dependent task scheduling problem in edge computing. The proposal showed faster when compared with BGWO and performed better than BBA and BPSO.
- Panwar *et al.* [132] showed that BGWO has superior results solving unit commitment problem when compared to classical and heuristic approaches. The algorithm has demonstrated better performance in small,

medium and large instances. Reddy *et al.* [133] used BGWO to solve profit unit commitment. The results showed that one of BGWO configurations reached the highest profit in all the cases.

I. APPLICATIONS OF BPSO

In our search, we found 263 papers addressing BPSO, about 65% of the total amount. Hence, different from the other algorithms discussed in this work, we chose to present the applications dividing the papers into main areas. We list below the number of works categorised in each case and the main subjects:

- a) Benchmark (54 papers) - feature selection in repositories, optimisation of mathematical functions, OneMax problem and 0/1 knapsack problem;
- b) Biology (34 papers) - drug design, DNA design, detection of diseases, cell layout, feature selection, classification in diseases dataset, among others;
- c) Computer Sciences (36 papers) - face recognition, image processing, software tests, computing security, coalitional games, neural networks design, cryptanalysis and software reliability;
- d) Engineering (21 papers) - allocation problems, scheduling, electrohysterogram signal, design resonators and structural topology;
- e) Electrical Engineering (69 papers) - unit commitment problems (UCP), phasor measurement unit (PMU) placement, control, power system reliability, distribution feeder scheduling, distributed generation interconnection, placement of generators, fault location, hydrothermal generation scheduling, microgrid and voltage regulation;
- f) Telecommunications (27 papers) - antenna problems, wireless sensor network, wireless local area network (WLAN) and cognitive radios;
- g) Others (22 papers) - NARX model structure, classification, time series, text mining, steganography, routing problem in VLSI circuits, among others.

Most of the papers used the BPSO as a previous step before applying another technique to solve the problem. In this sense, the algorithm is utilised, especially in feature selection. In Appendix B (Section D), we present all the acronyms of the version of the PSO, which is the most widely present in our study. We also suggest that the paper from Jordehi *et al.* [1] covers a great review about PSO for discrete optimisation problems.

VI. OTHER ALGORITHMS

We present in this section the summary of other 20 different swarm-inspired techniques that are not as popular as the ones previously discussed in Section IV, but they can present better results depending on the problem.

Algae: The Binary Artificial Algae Algorithm (BAAA) mimics the algae search for food [134], [135]. BAAA [134] is created in 2016 using three mechanisms: elite local search, transfer function and repair

operator. The repair operator that is something unique from BAAA minimises the search by removing the infeasible solutions. This version of BAAA applied to 94 benchmark problems is efficient when compared to MBPSO, BPSOTVAC, CBPSOTVAC, GADS, bAFSA, and lBAFSA. Then, in 2018, Korkmaz and Kiran [135] proposed a new version of BAAA using a stigmergic behaviour and XOR logic operator to solve uncapacitated facility location and benchmark problems. The stigmergic operator is a controlled mutation that allows the swarm to find better solutions. This proposal displayed efficient while comparing to GAs, BAAAs, BPSO, SBHS, HS, BLDE, BHTPSO-QI, GBABC, BQIGSA and SabDE.

Ant Lion: Emary *et al.* presented three versions of the Binary Ant Lion Optimization (BALO) [20], which is inspired in the ant lions hunt process. The case study is feature selection using the UCI repository. A similar task is solved by Mafarja and Mirjalili [43] using the Hybrid Binary Ant Lion Optimization (HBALO). This hybrid model combines rough set and conditional entropy-based methods (QuickReduct and CEBARKCC). This proposal overcame other BALO versions, as well as PSO and GA.

Bacteria: The Binary Bacterial Foraging Optimization (BBFO) is addressed in two papers [136] [137]. The biological inspiration comes from the bacterial foraging, which includes chemotaxis, swarming, reproduction and elimination-dispersal operations. The studies addressed instances of the 0/1 Knapsack Problem and Feature Selection, respectively.

Butterfly: Binary Butterfly Optimisation Algorithm (bBOA) is a mono-objective version that adapts the fitness function as a regression to solve multi-objective problem [138]. The BOA shows a high exploitation and convergence rate based on the employed of a random walk and elitism. The method is inspired by the food foraging behaviour of butterflies, which is based on the sense of the fragrance of flowers. In the investigation provided by Arora and Anand, the authors use 21 datasets from the UCI repository to prove the efficiency of the method [138]. Also, they use S- and V-shaped transfer functions.

Coyote: The Binary Coyote Optimization Algorithm (BCOA) was inspired by the intelligent social organisation of a group of coyotes. A binary approach was introduced by Souza *et al.*, which uses the hyperbolic tangent as a transfer function, to deal with binary data [139]. In this work, the authors investigated the effectiveness of the method using a Naive Bayes classifier and benchmark functions.

Crow: We identify two works that addressed the Binary Crow Search Algorithm (BCSA). The natural inspiration comes from flocks of crows that fly in a surface. These animals can hide food, memorise and protect their caches. Also, they follow each other to find a

better food source. The work from Souza *et al.* suggests a binary version of the continuous CSA that uses the V-Shaped transfer function [140]. The case study involves feature selection using benchmark problems. A second work proposes to solve two-dimensional packing problem with Fixed Orientation using a BCSA with S-Shaped transfer function [141]. The computational results showed the superiority of the method in comparison to the BPSO.

Cuckoo: The Binary Cuckoo Optimization Algorithm is inspired in the brood parasitism of cuckoo bird, which places their eggs in the nests of other species. Dalili and Karegar applied a modified version of the algorithm (the MBCOA) in some PMU placement problems [142]. Some instances of the Unit Commitment Problems are solved using the Improved Binary Cuckoo Search Algorithm (IBCSA) by Zhao *et al.* [143]. Garcia *et al.* applied the Binary Cuckoo Search Big Data Algorithm in scheduling tasks, providing a comparative investigation on different versions of the algorithm [144].

Dragonfly: The Binary Dragonfly Algorithm (BDA) was inspired by the social interactions of such animals when searching for food and escaping from enemies. The dragonflies are separated to avoid collisions and have an alignment (velocity matching of individuals with the other individuals in the neighbourhood), and cohesion (attraction of individuals towards the centre of the mass). The work from Mafarja *et al.* addresses the BDA for feature selection using V-shaped and S-shaped transfer functions with regression classification error. The case studies involve benchmark problems [145]. The work from Hariharan *et al.* proposes a combination of wavelet packet-based features and BDA for feature selection to classify the different types of infant cry signals [146].

Fish: The Breeding Artificial Fish Swarm Algorithm (BAFSA) is used to solve an optimal cluster head selection in Wireless Sensor Networks problem by Sengottuvelan and Prasath [147]. The method was inspired in the preying of a school of fish. The Binary Fish School Search Algorithm (BFSS) is another algorithm inspired on fish introduced by Sargo *et al.* [148]. In this investigation, the method was applied for feature selection in benchmark problems, and the intensive care unit readmission problem. The algorithm mimics the collective behaviour of fish schools, regarding their mechanisms of feeding and coordinated movement. Carneiro *et al.* [53] improved the BFSS to Feature Selection by adding a random flipping mechanism of variables and biasing the swarm to initialise closer to 0. Then, Santana *et al.* introduced the Simplified Binary Fish School Search (SBFSS), which present several modifications in comparison to the BFSS, such as a reduction in the number of free parameters [149]. Considering the KNN as a classifier, the SBFSS overcame other methods in feature selection, like ABC, GA, and PSO [149].

Fruitfly: The Binary Fruitfly Optimization Algorithm (BFOA) was inspired by the foraging behaviour of fruit flies, highlighting their sensitive vision and the smell of the food [150]. In the paper from Wang *et al.*, the authors present some modifications to solve some instances of the multidimensional knapsack problem (MKP).

Grasshopper: The Grasshopper Optimization Algorithm (GOA) was inspired by the behaviour of grasshopper swarms in nature, considering food search and social interactions [151]. In larval stages, the agents of a swarm perform slow movements through small steps. On the other hand, in adulthood, they perform long-range and abrupt movements. These combinations are used as the basis for the GOA [152]. The work from Hichem *et al.* [151] introduces a binary version of the algorithm to deal with feature selection in benchmark functions. The work from Mafarja *et al.* proposes the application of transfer functions to solve similar tasks. Pinto *et al.* use these premises to deal with some instances of the binary knapsack problem [153].

Glowworm: The ability of glowworms to change the intensity of the luciferin emission was the main inspiration to create the Binary Glowworm Swarm Optimization (BGSO). Mingwei *et al.* [154] addressed this idea to solve instances of the unit commitment problem. Xia *et al.* introduce a forecasting method based on improved binary glowworm swarm optimisation and multi-fractal dimension (IBGSOMFD) for feature selection. In this work, the proposal combined with an SVM overcame other methods, considering benchmark databases [155]. Also, the method is used to risk prediction of P2P lending investment.

Moth: A discounted knapsack problem is solved by using a binary version of the Moth Search Algorithm [156]. The method was inspired by Lévy flights and fly straightly of the moths. The authors proposed an investigation considering for the first time nine mutations procedures.

Owl: Once again, feature selection is the application addressed to test a swarm-based method. In work from Mandal *et al.*, the Binary Owl Search Algorithm (BOSA) is introduced considering six variants of transfer functions [157]. The method was able to improve the classification accuracy of an SVM, overcoming traditional approaches, as BPSO, BGA and BHS.

Pigeon: Rojas-Galeano proposed a modified version of the Urban Pigeon-Inspired Swarm Algorithm to deal with binary problems [158]. Urban pigeons present feeding habits that can be classified into two modes, flock feeding and solitary feeding (similar to the Cat Swarm Optimization). The solitary mode is used to explore the surface to avoid premature convergence. In the flock mode, the agents follow a pigeon that has found a food source. The authors used benchmark functions and real-world problems to evaluate the search capability of the method.

Salp: Salps are marine animals with cylindrical gelatinous bodies. They move by pumping water longitudinally through their bodies. Simultaneously, they filter such water through a set of internal structures to retain plankton. A group of these animals is named salp chain. Based on these premises, the Salp Swarm Algorithm (SSA) was developed, using as inspiration their food search. The work from Ahmed *et al.* proposed a binary version of the SSA with Chaotic maps to solve feature selection, using benchmark functions [159]. Antenna array synthesis problems are solved by Mondal and Saxena addressing the same algorithm and transfer functions to transform the output response into binary strings [160]. The work from Rizk-Allah *et al.*, proposed a new binary version of the algorithm, using modified transfer functions [161]. Other works have addressed the BSSA, like in [160], [162]–[164], mainly in feature selection tasks.

Spider: From Cuevas *et al.*, was inspired by groups of spiders that interact using as rules the biological laws of the cooperative colony [165]. From Yu and Li, is based in foraging strategy of social spiders, using the vibrations on the spider web to determine the positions of preys [166]. Shukla and Nanda used the Binary Social Spider Optimization (BSSO) algorithm [167], which was based on the proposal from Cuevas *et al.* [165]. Such a method was addressed for unsupervised band selection in compressed hyperspectral images. In 2020, Binary Social Spider algorithm (BinSSA) [168] was introduced by showing the effect of choosing different transfer functions and using or not a crossover operator. The authors stated that the usage of crossover is effective because it balanced the exploitation and exploration mechanisms. This algorithm follows Yu and Li's approach. Similar propositions were presented in [169] and [168].

Symbiotic Organism: The Binary Symbiotic Organism Search (BSOS) was proposed in the work from Han *et al.* [170]. They used the continuous version of the algorithm as the base, addressing S-Shaped functions to transform the agents into binary strings. The algorithm was inspired by the symbiotic relationship between two individuals from different populations in an ecosystem. The authors classify these relationships in mutualism, commensalism, and parasitism. Once again, the feature selection problem is addressed considering benchmark classification problems using the KNN as the classifier.

Vulture: Almonacid *et al.* solve the manufacturing cell design problem using the Egyptian Vulture Optimization Algorithm (EVOA) [127]. The inspiration of the method arises from the abilities of the Egyptian vulture to break eggs using pebbles and to rotate objects using twigs.

Whale: The Whale Optimization Algorithm (WOA) was inspired in the behaviour of the whales in the oceans, based on their feeding. These marine mammals swim to prey in a unique spiral. This process was modelled considering three specs: the scope of hunting,

spiral trajectory, and random search. Some works have addressed binary versions of the WOA. The investigation conducted by Xu *et al.* proposed the use of an improved BWOA for Feature Selection of Network Intrusion Detection [171]. The research group from Hussien solved feature selection and discrete optimisation problems using this algorithm [172], [173]. Reddy K. *et al.* and Kumar and Kumar approached unit commitment problems [174], [175]. In all cases, they showed that the algorithm could overcome many other metaheuristics, mainly versions of the PSO.

A. COOPERATIVE ALGORITHMS

In this section, we present other algorithms that are not directly inspired by groups of animals' behaviour. However, their operators and forms of information spread between the agents are cooperative instead of evolutionary (generational). In this sense, the agents do not “die” along with the iterations, but change their position in the search space considering the collective information.

To exemplify the idea, nature-inspired algorithms, such as the Black Hole Algorithm or Tree-Seed Algorithm, mimic other kinds of natural phenomena. We can find a similar idea in the BGSA algorithm of Section B-A4. In contrast, the Fireworks Algorithm or the Open Source Development Model comes from artificial phenomena inspirations.

The Forest Optimization Algorithm (FOA) comes from the process of seeding of trees. Ghaemi *et al.* introduced this method to solve feature selection tasks using benchmark functions proposing a binary version, the FSFOA [176].

The binary version of the Tree-Seed Algorithm was proposed by Cinar and Kiran [42]. They introduced three ideas to implement the binary proposal: using logic gates (LogicTSA), similarity measurement techniques (SimTSA), and a hybrid variant (SimLogicTSA). As the name indicates, it mimics natural tree-seed behaviour too. The last approach was very competitive in comparison to binary versions of the ABC, PSO and DE.

The Binary Brain Storm Optimization (BBSO) was applied to medical data classification by Ogwo *et al.* [177]. The human brainstorm process inspired the BBSO. The authors used wrapper methodology and several classification methods. Similarly, the Imperialist Competitive Algorithm (ICA) emulates human social evolution. The work from Mirhosseini and Nezamabadi-pour address a binary version of such algorithm based on transfer functions to solve knapsack problems, the feature selection problems, and the Content-Based Image Retrieval [178].

The display of fireworks upon explosion inspired the Fireworks Algorithm (FWA). A binary version (BFWA) is introduced by Reddy *et al.* [179] to solve a profit based unit commitment (PBUC) problem. They execute an extensive investigation on the performance of the proposal, including versions of the PSO, GA and others, proving the efficiency of the BFWA. Xu *et al.* [180] also used the BFWA to solve instances of the knapsack problem.

The open-source software development mechanism and community's behaviours inspired the Open Source Development Model (ODMA) algorithm. The binary version of the method is introduced by Khormouji *et al.* [27]. In their study, the method overcame the BPSO and GA in benchmark functions optimisation.

The behaviour of black holes in outer space and the gravitational attraction inspired the Binary Black Hole Algorithm (BBHA). Pashei and Aydin investigated the application of the method in feature selection and classification on biological data [11]. They conclude that it can overcome versions of the BPSO, GA and others.

Binary Harmony Search Algorithm (BHSA) has an exciting inspiration: the improvisation process of jazz musicians. Three papers addressed this method in binary problems: i) from Wang *et al.* [181], to solve 0/1 knapsack problems; ii) from Gholami *et al.* [182], in which the authors use versions of the BHSA in feature selection for classification; iii) from Lee *et al.* in which the authors use it in feature classification in EEG signal [183].

The Binary Equilibrium Optimization (BEO) is a physics-based algorithm (as the GSA) inspired by dynamic controlled volume mass balance models to estimate equilibrium states. The paper from Gao *et al.* [184] presented a version considering the sigmoid as transfer function for feature selection in benchmark problems. Using similar problems, Zhao *et al.* changed the transfer functions to V-Shaped approaches [185]. In both cases, the comparative analysis involved the use of distinct parameters.

The Binary Water Wave Optimization (BWVO) for feature selection was introduced by Ibrahim *et al.* [186]. The algorithm was inspired by water waves phenomena, like propagation, refraction, and breaking. Using 17 datasets, the authors showed that the BWVO could overcome traditional approaches, as the PSO and GWO, considering the KNN classifier.

Binary Multi-Verse Optimizer (BMVO) [187] is applied on 13 benchmark functions (unimodal and multimodal), Feature Selection using four UCI datasets, and 0/1 knapsack problem. The BMVO was also compared to Binary Bat Algorithm, Binary Particle Swarm Optimization, Binary Dragon Algorithm, and Binary Grey Wolf Optimizer. The method was superior on the majority of functions with different dimensions with breakneck convergence speed.

VII. HYBRID APPROACHES

Some papers addressed in this work present significant improvements using the hybridisation of swarm-based algorithms with some other nature-inspired proposals. Figure 6 presents a visualisation of the most common algorithms/methods adopted in the hybridisation process. Note that the connections in the network mean that nodes connected were used in a hybrid algorithm. Also, the size of the node is proportional to the number of times that the algorithm was used. Note that, among the papers analysed, depending on the characteristic of the problem tackled by

the algorithm, the hybridisation can happen between an optimisation method and a problem-specific algorithm (e.g. the PSO combined with the K-Means to deal with clustering problems). Furthermore, we can see that the Bird, Bat, Wolf Pack and Firefly are the swarm inspirations that are most frequently used to produce hybrid versions.

Zhao *et al.* [188] proposed a priority planning and hierarchical learning (PHSO) adding the velocity updating from LLSO to BPSO. The algorithm proposed can produce high-quality solutions. The added mechanism divides particle into groups that where particles in inferior groups can learn from superior groups. Using learning steps, it is possible to select high-priority resources to become a candidate solution.

A Hybrid iBPSO and SFLA algorithm was proposed by Rajamohana and Umamaheswari [189] to solve feature selection to improve the accuracy of classification of fake reviewers. In the proposal, iBPSO population is provided as an input to SFLA algorithm that uses the pre-optimised solution as the initial population. Mafarja and Mirjalili [43] have created a hybrid approach using a similar approach. In their proposal, the binary ant lion starts using an improved population generated by two rough sets entropy reduct methods (QuickReduct and CEBARKCC).

Jia and Lu [190] have proposed a Taguchi binary particle swarm optimisation (HTBPSO) to optimise antennas designs. They have also included catfish operator in order to avoid premature convergence.

Kumar *et al.* [191] have proposed hybrid binary PSO and sine cosine algorithm (HBPSOSCA) to solve feature selection. The proposed algorithm uses SCA and PSO in order to improve exploration and exploitation, respectively.

Another hybrid technique with PSO is present in the literature. Using binary PSO with the decision tree pruning technique, Malik *et al.* applied the algorithm to network intrusion detection [192].

Lin *et al.* [193] have proposed a Hybrid Binary Particle Swarm Optimization (HBPSO) that adds a new position updating rule, the tabu-based mutation operators to generate diversity. Additionally, an iterated greedy local search procedure was proposed to repair infeasible solutions obnoxious p-median problem. In a different paper [194], they have proposed an HBPSO/TS a variation that uses a tabu search to intensify the search.

A home energy management system formulated as an MKP was proposed by Naz *et al.* [195]. Enhanced Differential Harmony Binary Particle Swarm Optimization is an hybrid algorithm that uses HSA, EDE and BPSO.

A Hybrid Binary Dragonfly Enhanced PSO (HBDEPSO) [196] and Hybrid Binary Bat Enhanced PSO (HBBEPSO) [197] were proposed by Tawhid and Dsouza to solve feature selection problem. Using the dragonfly algorithm, the HBDEPSO proposal can obtain diverse solutions, and the enhanced PSO increased the convergence to the global best solutions. As HBDEPSO, in HBBEPSO, the velocity vectors were updated independently for both algorithms. It was done

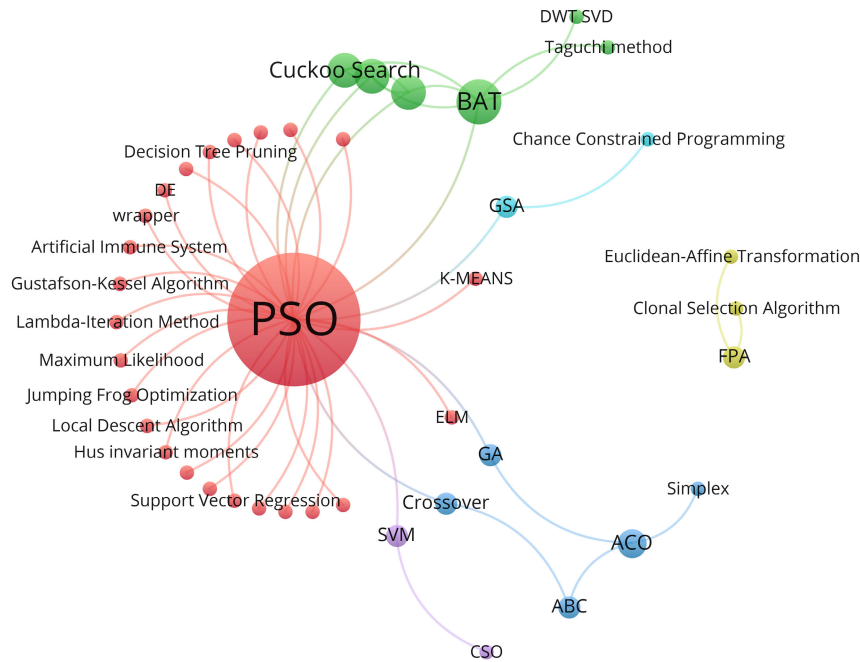


FIGURE 6. The network depicts examples of hybridisation proposed in some of the papers selected in this study. The nodes represent algorithms or techniques, while the links indicate that the nodes connected were used to produce a hybrid algorithm. Also, the nodes' size means the number of times that a node was a component of a hybrid algorithm.

to allow the algorithm to explore the search space in an alternative fashion.

Sarhani *et al.* [198] proposed BMPSOGSA and BPSOGSA, two approaches that combine BPSO and BGSA to solve the feature selection problem. BMPSOGSA differs from the other proposal because the authors have included a mutation operator to enhance population diversity. BMPSOGSA proposal has reached better results than other metaheuristics and other well-known methods for feature selection.

Too *et al.* [199] have proposed a new hybrid method called Binary Particle Swarm Optimization Differential Evolution (BPSODE) to solve feature selection problem in EMG signals classification. In the proposed approach, BPSO and BDE are computed in sequence. Hence, no extra computational cost is required. The proposal showed as a powerful feature selection tool overtaking other algorithms in the metrics used.

Al-Tashi *et al.* [200] proposed the BGWOPSO, a hybrid algorithm for feature selection using bGWO1 and PSO. Al-Tashi *et al.* also proposed the BMOGWO-S (Binary Multi-Objective Grey Wolf Optimizer) based in sigmoid transfer function [201] as proposed in (56) for feature selection.

Shunmugapriya *et al.* [18] developed the ACABC algorithm, a hybrid between the ACO and ABC to deal with binary problems. They applied it to optimise benchmark functions.

Rajamohana *et al.* [202] introduce the hybrid IBPSO with Cuckoo Search Optimization (CSO) in spam detection. Galvan *et al.* [203] and Ko *et al.* [83] apply a hybrid between

BPSO and Differential Evolution to solve feature selection problems.

The paper from Ruiz-Rodriguez *et al.* [204] uses the BPSO hybridised with the Jumping Frog Optimization to solve a voltage regulation problem.

Some versions of Artificial Immune System were utilised to perform hybrid methods. Sayed *et al.* [128] create the Binary Clonal Flower Pollination Algorithm mixing the BFPA and Clonal Selection Algorithm to solve an instance of the unit commitment problem. Pu *et al.* [105] and Zhai *et al.* [205], on the other hand, developed the BPSO hybridised with some versions of the Artificial Immune System.

Remarkably, most of the papers suggest the hybridisation between the most known evolutionary model, the Genetic Algorithm (GA). The BACO algorithm is hybridised with the GA to create the modified BACO (MBACO) by Wan *et al.* [120]. A similar proposal is developed by Wang *et al.* [60], the modified coded ACO algorithm combined with GA (MBACO). Both papers solved feature selection problems.

The works of Fathy *et al.* [206] and from Mirjalili *et al.* [30] present hybrid versions of the BPSO and GSA. The last introduced the BPSOGSA algorithm. The paper of Zeng *et al.* [207] introduces the Mixed-Binary Evolutionary PSO (MB-EPSo), using BPSO and GA to solve a scheduling problem.

In the paper from Ozturk *et al.* [110] the authors proposed the BABC Based on Genetic Operators (GB-ABC), utilising the idea of crossover from the GA. The same research group

in [3] introduces the Improved DisABC (IDisABC) using the same approach. In both papers, the problems addressed are benchmark datasets. In the same way, Suresh *et al.* [103] proposed the Hybrid Improved BPSO (IBPSO) and solved the generation maintenance scheduling problem. Besides, the BPSO with Crossover (BPSOC) was discussed by Singh *et al.* [4].

The mutation in GA is the main inspiration to create two distinct versions of the Modified BPSO, one from Lee *et al.* [208] and other from Luh *et al.* [209]. While in the first they solve a benchmark problem, in the second the proponents apply the new model in continuum structural topology optimisation.

Wei *et al.* [210] present the hybrid BPSO (HBPSO), which uses the GA. The application is related to find the free parameters of an SVM to classify Crohn's disease and Lung cancer. The same idea is used by Jin *et al.* [211] (hybrid BPSOGA), and Zhou *et al.* [212] (Differential BPSO-GA), which were applied in the 0/1 Multidimensional Knapsack Problem. Zouache *et al.* introduced the Quantum-Inspired Firefly Algorithm with Particle Swarm Optimization (QIFAPSO) to solve the same task.

In the same way, Pashei *et al.* created the hybrid Binary Black Hole Algorithm and Modified BPSO to solve some instances of the uncapacitated facility location problem [213]. They have shown that applying BBHA as a local optimiser for BPSO (4-2) it was possible to increase the local search capability, effectiveness significantly, and reliability of BPSO (4-2) solving gene selection problem.

VIII. MULTI-OBJECTIVE APPROACHES

Multi-objective approaches are more complex than mono-objective ones because their optimisation consider 2 or 3 fitness functions simultaneously. There are several operators to qualify multiple fitness functions and to adapt the mono-objective methods (operators for exploration, exploitation, convergence and diversity aspects). In mono-objective problems, one solution is either better or worse than another solution. However, in multi-objective problems, we can have indifferent solutions called non-dominated solutions that are the ones that are better than all other (dominated) solutions, but within their group, there is not any solution that is ultimately better than the other. For example, solutions i and j are better than all other possible solutions. A solution i is better than a solution j in the objective a , but the opposite happens to the objective b . In this way, we cannot say that the solution i is better than solution j neither solution j is better than solution i , so they are non-dominated solutions. Non-dominated solutions are a set of solutions that are better than all the other solutions found by the execution of an algorithm. However, no solution in the set is better than the others.

Some multi-objective algorithms stores the non-dominated solutions in the called External Archive (EA) [140], [214], [215]. As the solutions are represented by the values of each dimension that outputs the best fitness values for 2 or 3 functions, it means that the required memory storage is higher as

you increase the number of dimensions of the problem. In this way, the usage of External Archive (EA) requires high memory space, but it simplifies the mechanisms of optimization because they do not need to save any current information of the search space. The majority of multi-objective algorithms uses external archive, because the computational cost to allow the swarm to maintain good solutions in the execution time is usually higher than the cost to use an EA [216]. The EA also requires to be updated and sorted as new non-dominated solutions are found in the search space. Crowding distance is one method used to rank and limit solutions in the external archive [215], [217].

The vast majority of multi-objective algorithms are applied for feature selection regardless of the type of data or the inspiration [168], [201], [215], [218]–[221]. The applications are varied such as biological or healthy-related problems [92], [222], [223], antenna design [224], electric power problems [225]–[228] and computer sciences [192], [229], [230]. As PSO is one of the most applied algorithms in the literature, in multi-objective problems, it could not be different. The majority of versions for multi-objective problems are inspired in the PSO. Some algorithms are listed:

- 1) Gong *et al.* [223] introduced the Multiobjective Comprehensive Learning PSO with a representation scheme based on binary search (BS-MOCLPSO) to deal with bed allocation in the hospital.
- 2) Souza *et al.* [218] developed the Multi-Objective Particle Swarm Optimisation with Crowding Distance, Roulette Wheel and Local Search (BMOPSO-CDRLS) to optimise a feature selection benchmark problem. BMOPSO-CDRLS outperformed BMOPSO-CDR and NSGA-II. The same research group introduced the Novel Binary Multiobjective PSO with Crowding Distance and Roulette Wheel (NBMOPSO-CDR) [82] to solve an analogue problem; NBMOPSO-CDR overcame some variants of the MOBPSO.
- 3) The Mixed-Discrete PSO was proposed by Chowdhury *et al.* [231]. The computational results evolving benchmark functions proved that it is competitive since it was better than PSO and NSGA2 [217].
- 4) Mahmood *et al.* [232] introduced the Discrete Barebones Multi-Objective PSO (DBB-MOPSO) and applied it in two types of problems: pavement maintenance schemes and benchmark functions. The new algorithm overcame the DMOPSO, GA and NSGA in both tasks.
- 5) Inspired in PSO, Zhang *et al.* introduce the Binary Multiagent Coordination Optimization (BMCO) [233] to optimise benchmark functions. The proposal presented better results than the BPSO and novel BPSO (NBPSO).
- 6) The improved binary particle swarm optimisation (IBPSO) is proposed to solve the multi-objective operation mode optimisation of medium voltage distribution [234]. By introducing nonlinear dynamic adjustment learning factors and inertia weight, the

convergence and optimisation of the proposed algorithm are improved compared to BPSO.

Some algorithms were not only inspired on the PSO, but also in another metaheuristic. The Multiobjective Hybrid Real-Binary (MOHPSO) algorithm was inspired on the MOEA and PSO operators [235], and MOHPSO outperformed versions of GA, BPSO, and PSO. Xu *et al.* [219] proposed the crowding, mutation and dominance binary PSO for feature selection (CMDPSOFS), CMDPSOFS succeed a variant of the NSGA-II [217]. A New Modified BPSO (mBPSO) was presented by Fan *et al.* [29] to solve a multi-objective resource allocation problem (MORAP), it is better than versions of GA and ACO.

Another example is the Hybrid Improved Binary Quantum Particle Swarm Optimization (HI-BQPSO) [221] is a multi-objective algorithm applied for feature selection that reduces the number of selected features while maximising the classification performance. The results show that HI-BQPSO compared to ABC, SA, GA, and BQPSO has good overall performance, strong search capability, and was able to maintain high efficiency with a range of different classifiers. The Co-Operation of Biology Related Algorithms (COBRA-bm) [236] developed by Akhmedova *et al.* is also a hybrid binary method which involves the PSO, Wolf Pack Search, FA, Cuckoo Search Algorithm and Bat Algorithm. They showed that COBRA-bm could achieve better performances in comparison to each one separately, in benchmark problems.

Other inspirations are also present for binary multiobjective optimisation as follows:

Fish: Macedo *et al.* [215], [220] proposed several versions of the Multi-Objective Binary Fish School Search. Their goal was to reach the most effective version regarding computational cost and performance. The most effective versions are MOBFSS-1-LS and MOBFSS-3-LS. In both versions, they replace the original mechanism of turbulence on the external archive for the local search method inspired by the BMOPSOCDRLS [218]. The MOBFSS-1-LS version shows the best results considering hypervolume, spacing and maximum spread, but the MOBFSS-3-LS shows the smallest computational cost. In their proposal, they show that having a reasonable control on the collective movements turns the individual movement dispensable.

Wolf: Binary multi-objective grey wolf optimiser based-on sigmoid binary transfer function (BMOGW-S) [201] is a multi-objective algorithm inspired on the GWO. BMOGW-S uses an external archive, and three leaders from the archive perform the hunting mechanism. BMOGW-S is applied to feature selection using 15 datasets from the UCI repository. The results showed that the proposed BMOGWO-S could effectively determine a set of non-dominated solutions. The proposed method outperforms the existing multi-objective approaches in most cases in terms of features reduction as well as the classification error rate while benefiting

from a lower computational cost. The authors also show that handling of feature selection as a problem with more than objective, as BMOGWO-S can explore the space more efficiently to attain a set of non-dominated solutions better than treating the problem as a single solution. The results demonstrated that BMOGWO-S could attain better non-dominated solution in most cases. BMOGWO-S outperformed the benchmarking algorithms on both feature reduction and classification accuracy.

Firefly: Zouache *et al.* [237] propose the Multi-objective Firefly algorithm with Particle Swarm Optimization (MOFPA) that combines the firefly algorithm FA with PSO. For every two individuals i and j , if the individual j weakly dominates the individual i , the FA strategy is performed; otherwise, the PSO strategy is performed. This work compares the performance of MOFPA with NSGA-II [217], SPEA-II [214], MOQAIS, MOEA/D in the Knapsack problems showing that MOFPA provides a better spread of solutions with a better convergence behaviour.

Bee: The Binary Bees Algorithm (BBA) [113], Multi-Population Binary Ant Colony Algorithm with Concrete Behaviors (MPBACB) [57], Decomposition-Based BACO (MOEA/D-BACO) [121], Bottleneck Assigned Binary Ant System (baBAS) [115], Multi-Binary Bat Algorithm (MBBA) [39] and Multi-objective Binary Cat Swarm Optimisation (MOBSCO) [125] were previously described in their respective sections.

We notice that the literature in multi-objective algorithms is focused on versions of PSO. This is again the case because of the popularity and simplicity of PSO and the lack of computing power in the past. We see that the multi-objective and many-objective algorithms have been more present in the literature in the past years but still concentrated in a small number of research groups because of the need for high computational power and memory. Besides, the complexity of these two classes of problems is much higher than for mono-objective algorithms.

IX. DISCUSSION AND DIRECTIONS

This section presents our considerations on the binary swarm-based algorithms extracted from the 403 selected papers. The Genetic Algorithm was created using binary variables because of the computational power available at that time. However, the development of computer components and their cost/price reduction contributed to creating nature-inspired methods on continuous problems. This change originated several continuous new techniques, but it might also cause a decrease in the creation or improvement of binary techniques. Recently, binary optimisation has expanded because of the absence of robust solutions across many real problems.

Even though the Genetic Algorithm has been successfully solving several binary problems for years, after some of our works with Swarm Intelligence, we learned that SI might have

more capability of solving harder binary problems because of the smooth convergence operators from SI. In GA, two operators are mainly used: crossover and mutation. In the crossover operator, the chance of creating similar solutions are high because the mixing of already known individuals does not tend to create a highly diverse set of individuals for binary optimisation. A premature convergence can become quick by the high presence of similar individuals. Problems with high dimensionality are less likely to be susceptible to a high presence of similar individuals. For the case of mutation, flipping random features (0 to 1 or 1 to 0) can prevent the swarm to come back to this state over the iterations. For instance, some features can only increase the fitness value combined with other features, so random flips can drastically impact the fitness value. The lack of impact in the context can also show the irrelevance of a feature, but the strong negative or positive impact is uncertain.

Adapting continuous algorithms to binary problems can require more resources and rules than is needed. Binary search space has fewer combinatorial options and is more sensible than a continuous search space. Each feature for a binary problem should be whether 0 or 1, but, in a continuous problem, it exhibits many more options (e.g. 0.0001 to 1, 0.0001 to 0.01, or 0.0001 to 0.0002). Consequently, some variations in a continuous search space can represent no change on a binary search space, not allowing the agents to explore new solutions in a binary space. In addition, when a feature is somehow dependent on another one, small changes in the continuous search space can have less impact on the fitness value than for the case of binary search space because it is easier for the agent to return to previous regions.

Exemplifying, the fish, ant or bee is going to be moving around by little steps, and the change for a binary mapped vector is going to be rare. The inspired agent will need to move several times to influence the binary vector, which can be prevented by using Binary-Binary approaches. A binary problem demands a peculiar balance of convergence and diversity, which avoids similar solutions and premature convergence. Even though it appears that continuous optimisation is smooth, in fact, the projection of the optimisation for the binary space will not follow the same pattern. Thus, we need to treat binary problems using proper mechanisms that provide optimal convergence, small computational cost, and high accuracy.

We perceive that new binary operators, which are promising in swarm-based techniques, are being developed in the literature. These operators are ideal for binary optimisation also because it requires fewer parameter and mapping functions. In fact, a trend of parameterless techniques is the future. The adaptive versions of swarm-based algorithms are being published, and they will be soon replaced as more effective and fast algorithms are proposed. Unfortunately, for binary optimisation, we do not see as much this trend than on continuous problems, so here we encourage researches to work on this gap.

We observed that the number of relevant papers tends to increase, since 2006. It is noticeable an increase until 2013 and then a small reduction until 2017. This observation may be explained by the growth of Deep Learning and other classification techniques which attract everybody from Machine Learning to focus on them. The massive appearance of the Binary Particle Swarm Optimization (BPSO) algorithm is not a surprise, since, in the continuous case, this is the most prominent algorithm. It can be explained to its simplicity in implementation allied to good results achieved in many real problems. The importance of the BPSO is so high that almost all the papers which address another proposal as the main algorithm, use some PSO-based to perform comparative analysis.

We notice that a large number of papers, around half part, proposed new versions of existing algorithms. The researchers are still looking for the best swarm-binary algorithm version. Also, it is possible to say that the field is in constant evolution and, maybe, a definitive method can be developed soon. However, we cannot identify and point out the best binary version from all swarm-based algorithm because the area are still not organised regarding the comparisons, advantages and disadvantages. We notice that several new versions are only compared by one or two techniques along with one or two versions, which do not help us electing promising versions. Another aspect is that probably some of new versions or techniques might display similar behaviours after a better look at the rules process, but the literature lack from depth analysis on the social behaviour from the swarm. Some works argue that evaluating the social interaction of simple reactive agents. We can understand the peculiarities of the swarm-based algorithm and probably understand the differences between different versions, rules, and operators [238]–[243]. Besides, the extensive use of benchmark functions to evaluate the search capability of the methods does not help the field to determine the best models to deal with real-world problems.

We indicate in Table 3 some promising leading proposals that we believe have higher chances of being successful, being novel or having mechanisms that will be used in the future. The application of transfer functions will continue to appear in the literature not because of the computational cost or its efficiency, but because of the flexibility of applying any algorithm to a binary (or discrete) problem. The most used transfer function to transform a continuous vector into a binary string is the sigmoid function. The fact that the first binary version of the PSO uses this tool, as presented by Kennedy and Eberhart in 1997 [10] may be an explanation why the sigmoid function is widely adopted. It is possible to observe that some of the concepts addressed in the pioneer versions of the BPSO were utilised in many other swarm-based algorithms. Also, the use of the V-Shaped functions, especially the hyperbolic tangent, is very often [96]. In both cases, the main idea is to make the agent more similar to some best solution, as in the continuous case.

We highlight that each specific swarm-based technique or problem can display better results based on the different transfer function. However, the V-shaped seems to be more efficient than S-Shaped in general. The S-Shaped seems to be more effective at the beginning of the iterations, but not necessarily effective in the middle to the end of iterations. Moreover, adaptations of V-Shaped seem to be the most efficient choice or even a trend in the literature. For instance, NBBA shows good performance using multi-V-shaped transfer function [64].

Interestingly, NMBPSO [87] shows a hybrid transfer function of S and V-shaped that seems really effective because it provides local (V-shaped) and global (S-shaped) search over the iterations. We argue that new algorithms in the future might use hybrid transfer functions to better balance the exploration and exploitation for other swarm-based algorithms. Hybrid or adaptive mechanisms will definitely be more present in the literature in the next years because these mechanisms can provide a continuous search through better solutions that are generally better for most complex, high dimensional and multi-modal problems.

Looking at other binary operators, logic gates seem to be the most efficient ones in computational cost and performance. The simplicity of logic gates avoids unnecessary calculations, and provide efficient algorithms such as SBCSO [102], binABC [97] and BPSO [103], [104]. However, even though logic gates are very efficient in position displacement, logic gates are not the most efficient method to provide a high rate of diverse individuals. In this matter, we believe that methods inspired in similarity metrics are a great solution to provide diversity in the swarm such as the use of Jaccard's coefficient similarity in TSA [42]. The use of entropy is as rare as the use of similarity in swarm-based techniques. Entropy can be also use in the sense of providing diversity of individuals, but it is actually used for a filter and map purpose of minimising its complexity by focusing in reduced dimensional problem.

Depending on the inspiration, genetic operators showed high efficiency MBGWO [77], BABC [100], [110], BinSSA [168] and BPSO [100]. The methods of crossover and mutation have limitations on its efficiency for binary optimisation as they tend to generate similar individuals to the current population. In contrast to logic gates, genetic operators tend to have premature convergence because of the lack of diversity in the swarm. The advantage of Boolean gates compared to genetic operators is that they tend to create similar individuals in the direction of the boolean gate and not the swarm positions. For example, the use of an AND gate change more positions to 0 than to 1 indicating a higher probability of having vectors with more 0s than 1s.

The ant-inspired algorithms show less major changes from the original proposal because it already adapts pretty well to binary optimisation problems. We highlight the MBACO [60] algorithm because it outperformed a diverse set of algorithms as GA, BPSO, BACO, BDE, and Binary-Coded Cuckoo Search (BCS). In contrast, the algorithms inspired by bee,

cat and fish show very different techniques based on different operators and strategies. For example, NBABC [99] outperformed in accuracy and computational cost when compared to multiple algorithms: ABCBin [244], BABC [44], BitABC [98], GBABC [3] and XBABC [97], BCSO [245], BFSS [148], BGA [246] and MBPSO [247]. Other algorithms are not very popular in the literature but show good insights, such as the case of wolf-inspired algorithms that the QI-BGWO [79] appears as an efficient version to balance exploration and exploitation.

Hybrid and adaptive algorithms might be the next bets for Swarm Intelligence, as the field seems to be trying to create one single algorithm that solves perfectly any problem. Therefore, the only solution to have an algorithm good enough for several problems is making the swarm-based technique the most flexible and adaptive possible. The balance of exploration and exploitation are key for reaching optimal solutions.

Even though we constantly acknowledge that there is no free lunch, mixed algorithms comes with the hope that swarm-based algorithms will be as effective and flexible as mathematics and nature allows it. The three algorithms, HBBEPSO [197], HI-BQPSO [221] and ABPSO [248] seems to be interesting options to start studying this new branch of Swarm Intelligence. HBBEPSO [197] is applied for 20 feature selection problems, compared to six algorithms and assessed by six metrics. HI-BQPSO [221] divides the complexity of the problem into two steps (coarse and fine-grained), considers principles of cross-variation and learning, and compares its performance to nine gene expression datasets and 36 UCI datasets. ABPSO [248] highlights strategies to balance exploration and exploitation, analyses its time complexity and compares its performance to 150 benchmark instances. Taken together, the HBBEPSO [197], HI-BQPSO [221] and ABPSO [248], we are able to have a summary of advantages, drawbacks, data, metrics and methods for Particle Swarm Optimization. We argue here that even though PSO is the most popular and straightforward algorithm in the literature, it does not mean that PSO is the best choice to be transformed in the best adaptive algorithm. We still need to consider more other options in order to get in the consensus that one of the techniques is better for adaptive mechanisms.

Finally, for the case of multi-objective algorithms, the literature is still scarce in proposals but provides really promising techniques. In summary, the multi and many-objective approach evaluates more than two objectives at the same time. This balance of objectives powerfully enhances the computational cost of the techniques. However, most of the relevant problems recently have been described by more than one fitness function, and it was already proven by several papers that mono-objective approach is not as efficient in cases even when combining several mono-objective results. Consequently, the multi and many-objective areas should grow in order to help more and more our society.

TABLE 3. Current leading proposals based on the operators, transfer functions and inspirations. We also present the number of papers in which the popular options were proposed or applied in the literature. We observe that PSO continues to be popular and effective in the literature, but other swarm-based algorithms can be highlighted as effective while using logic gates, genetic operators and similarity metrics.

Methods	Popular Options	Leading Proposals	Number of Papers
Operators	Transfer functions	BPSOGSA [30]	50
	Genetic operators	MBGWO [77], BABC [100], BinSSA [168]	45
	Logic gates	SBCSO [102]	35
	Similarity	TSA [42]	7
	Entropy	HBALO [43], EBPSO [249]	7
Transfer function	S-shaped	BPSO [250]	24
	V-shaped	BFPA [41], BPSOGSA [30], NBBA [64]	14
	S & V-shaped	NMBPSO [87]	6
Mono-objective	Hybrid	HBBEPSO [197], HI-BQPSO [221]	45
	Adaptive	ABPSO [248]	26
	Bird	IBPSO [103]	263
	Ant	MBACO [60]	55
	Cat	SBCSO [102]	24
	Bee	NBABC [99]	18
	Bat	BCBA [65]	17
	Fish	SBFSS [149]	14
	Gravitational	XOR-BGSA [130]	10
	Salp	BSSA [164]	7
	Wolf	QI-BGWO [79]	5
	Flower	N-BFPA [41]	5
	Firefly	BFA [127]	4
Multi-objective	Bird	BMOPSOCDR-LS [218]	28
	Bee	BEES-BBA [113]	7
	Ant	MPBACB [57]	3
	Wolf	BMOGWO-S [201]	2
	Fish	MOBFSS-3-LS [215]	2

We identify in Table 3 leading multi-objective proposals inspired by bird, bee, ant, wolf and fish. BMOGWO-S [201] extensively compare its performance to the BMOPSO [229] and NSGA-II [217] for 15 Feature Selection datasets. BMOGWO-S improves the computational cost, the diversity of the leaders that guide the optimization, and the storage of solutions for next iterations. Because of its low memory space and straightforward mechanisms, BMOGWO-S shows a high performance for large datasets with high-dimensionality. There are other multi-objective algorithms that it would be useful to compare, such as BMOPSOCDR-LS [218] and MOBFSS [215]. All three swarm-based techniques use the mechanism of archive that it is widely used in the literature.

In summary, the leading proposals of the swarm-based methods - PSO, ACO, and ABC - are present in most papers as expected. However, it is remarkable that we found a total of 43 other approaches using different insights from nature. In the same way, it was unexpected that algorithms inspired by the cats and bats were so popular in the literature. The future opens a diverse set of inspirations to propose new efficient binary operators.

The idea to propose a new classification - Binary-Binary, Continuous-Binary, and Continuous-Continuous - in Section I arose when we observed that the binary algorithm follow some predetermined central concepts during their development. We notice that these ideas are not systematised and, sometimes, these various paradigms could hinder the understanding. Also, this unprecedented concept may help the researchers to increase the search power of the algorithms.

Looking at the Swarm Intelligence point of view, it seems that the area will be developing adaptive versions to prevent premature convergence by balancing exploration and exploitation. The big challenge of transforming the swarm-based approaches to parameterless is that it strongly impacts the balance between exploitation and exploration, the diversity of solutions and the convergence of the swarm. Moreover, differences on the landscape of fitness functions are also adapted by the use of different set of parameter values.

The Binary-Binary optimisation might not be the only approach, but it is the one that we believe it is the cheapest and most robust way of working with the binary optimisation

for Swarm Intelligence. We also believe that the future relies on the necessity of applying multi and many-objective approaches for real problems in which will be fundamental to understand both the swarm techniques and the type of problem because of the rise of the complexity of multiple objectives.

Our work brings awareness and understanding of many applications, algorithms and methods for Binary Optimisation from Swarm Intelligence. We understand that some problems might be better solved by other kinds of algorithms such as binary search trees, coevolutionary algorithms or binary neural networks. However, we argue that Swarm Intelligence brings robustness, flexibility, scalability, modularity, parallelism, and decentralization, which is a challenge to compete, especially in high dimensional and dynamic problems. Swarm Intelligence will be broadly used in the future as the literature converge to less volume of versions, fast and robust mechanisms, and mostly more understanding and organization in the field.

Some limitations and biases can be reflected in our analyses. Our analyses might not represent well papers that are not well-cited in the literature, not in English, and were published before and after our data collection. Moreover, some swarm-based inspirations are more popular than others. For instance, bird-inspired algorithms have a much larger volume of papers than those inspired by fireflies, fish, or wolves. In this way, our analyses follow this unbalanced distribution of papers across inspirations.

X. CONCLUSION

This paper presents an investigation of the most prominent swarm-based algorithms to deal with binary optimisation. This study carried out 403 papers from four important scientific databases: IEEE Xplore, ACM, Science Direct, and Springer, and some of them were also ranked by Google Scholar.

We propose a new way to categorise binary swarm-based algorithms: Binary-Binary, Binary-Continuous and Continuous-Continuous approaches. The difference between them lay in the intermediate steps to improve the solutions during the iterative process. Moreover, they differ generally in computational cost as they apply similar time-consuming mechanisms.

The final solution presented by the models has to be a binary vector, but, based on the previous continuous versions of the algorithms, the displacement (as the velocity in the BPSO) can be binary or continuous. The most used transfer functions to map continuous variables into binary vectors are the sigmoid function (S-Shaped) and two proposals of V-Shaped functions, which use the hyperbolic tangent and expression based on the arctangent. However, new adaptive and hybrid functions were proposed in the last years that balance better the exploration-exploitation necessary for effective optimisation.

It is important to remark that the binary swarm-algorithms found always come from previous continuous versions.

We describe in detail nine of them: Binary Artificial Bee Colony (BABC), Binary Ant Colony Optimization (BACO), Binary Bat Algorithm (BBA), Binary Cat Swarm Optimization (BCSO), Binary Firefly Algorithm (BFA), Binary Flower Pollination Algorithm (BFPA), Binary Grey Wolf Optimizer (BGWO), Binary Gravitational Search Algorithm (BGSA), and Binary Particle Swarm Optimization (BPSO). We also present some different versions of each, according to the most popular and efficient ones in the literature. Finally, we also list another 34 binary algorithms and their respective applications.

Around 7.5% of the papers address multiobjective methods, while another 10% presents hybrid proposals. Most of the hybrid algorithms address a swarm-based method together with the Genetic Algorithm. We consider the multi and many-objective methods the future of not only swarm intelligence but also for the general optimisation area because of the increasing necessity of applying more than one fitness function in real problems that usually requires at least the decrease of computational cost and the growth of any kind of profit (for example: money or accuracy of diseases).

We highlight that the use of binary versions inspired by the PSO algorithm continues to be the most popular in the literature. While the second most used proposal is the BACO, with 55 papers found, the BPSO appears in 263 papers. It is about 65% of the total amount of articles selected. The interest in this method may be related to the fact that the BPSO was the first swarm algorithm to be proposed in the literature, and also because of the simplicity in its implementation. Also, the popularity of the continuous PSO version can influence.

We encourage the field to focus on the Binary-Binary approach because it targets the problem using less complexity and more efficiency. Binary problems can be sensitive to small changes. Flipping some features can change the context entirely, and this may harm the convergence. Moreover, binary problems usually suffer from the issue of creating similar solutions which stagnate the system rapidly. Consequently, what seems to be an easy problem of flipping or not features becomes a hard problem to balance the convergence and diversity cautiously. We believe that binary problems should be solved slowly and using the randomness of operators carefully. Moreover, using continuous search space adds unnecessary work and time to the system because moving a little bit on the search space will not affect the solutions and fitness in the binary space, making a fake slow convergence and diversity.

We highlight that at least half of the papers are composed of new proposals or improvements in existing algorithms, increasing the applications of benchmark problems. In the same way, almost half of the papers were published in conference annals. We hope that this paper encourages researchers to work more in Swarm Intelligence in many ways: comparing existing techniques, understanding the nuances of operators, proposing new algorithms, and applying them to complex problems.

APPENDIX A CONTINUOUS-CONTINUOUS ALGORITHMS

A. BEE-INSPIRED

The Binary Artificial Bee Colony (BABC) versions were inspired by the behaviour of a honey bee swarm, being proposed by Karaboga in 2005 [3], [5], [97], [112], [251]. The bees are classified into three categories, each one playing a specific role during the process of finding a food source: employed, onlooker and scout bees [44]. The employed bees are in charge of bringing nectar from a known food source. Then, the employed bees share the information about the quality of this source (amount of nectar) to the onlooker bees, using a process called *waggle dance*. In nature, the duration of dance is proportional to the quality of a source [5], [12], [112].

The onlooker bees have to choose one of the food sources to explore. Their probability of selecting a source is proportional to its amount of nectar. Better food sources tend to attract more bees [5], [110]. The scout bees seek for unknown food sources in the vicinity of the hive, flying randomly. Whenever a source is exhausted (nectar is over), and employed becomes a scout. In the process of search, employed and onlooker bees perform exploitation (local search) whereas the scout bees perform exploration (global search) [3], [110].

Based on this metaphor, the food sources are the candidate solutions to the problem, and their nectar amount corresponds to their fitness. The solutions are not codified on the agent (bees) but in the environment (food source). The metaphor of different types of bee is used to select which food sources are more or less explored, as well as to define unexplored candidate solutions [18], [98].

The proposition of the BABC from He *et al.* [44] is a Continuous-Continuous method. This algorithm is initiated generating the food sources \mathbf{x}_i with real numbers according to (7):

$$x_{i,d}^{cont} = x_{d_{min}} + rand(0, 1)(x_{d_{max}}^{cont} - x_{d_{min}}^{cont}) \quad (7)$$

where $i = 1, \dots, N$ is the number of food sources, $d = 1, \dots, D$ is the dimension of the problem (the number of features), $x_{d_{max}}^{cont}$ and $x_{d_{min}}^{cont}$ are the lower and upper bounds of the d -th parameter defined by the user. At each iteration, new food sources are generated in the vicinity of the previous using (8):

$$v_{i,d}^{cont} = x_{i,d}^{cont} + rand(-1, 1)(x_{i,d}^{cont} - x_{j,d}^{cont}) \quad (8)$$

being j a random selected food source and $j \neq i$.

This idea is the same as the original ABC algorithm to deal with real problems. However, the fitness in the binary space must be calculated using binary strings. In this case, the authors suggest the use of the operation described in (9) to map the solution:

$$x_{i,d} = \begin{cases} 1, & \text{if } x_{i,d}^{cont} \geq 0 \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

After that, the evaluation of the binary solution $fit(\mathbf{x}_i)$ can be performed. Note that (9) can also be used to convert \mathbf{v}_i^{cont} into \mathbf{v}_i . The solutions and the process to change their positions occurs in the real space. The conversion into a binary string happens to calculate the fitness.

After the stage of employed bee, the onlooker bees choose a food source to explore based on the information found by the employed bees. The probability of selecting a food source is proportional to its fitness (amount of nectar), which is calculated using (10):

$$p_i = \frac{fit_i}{\sum_{j=1}^N fit_j}. \quad (10)$$

For a food source i , each onlooker draw a number $r = rand(0, 1)$ and if $r < p_i$, it explores the corresponding source \mathbf{x}_i , as in a *roulette wheel* scheme. Then, a new source \mathbf{v}_i is generated in the same way as in the employed bee. In the end, a greedy selection is performed.

The last stage of the BABC is the scout bee's phase. It just occurs if some food source is exhausted. An employed bee exceeds the maximum number of trials (*trial*) to improve the fitness of a source searching on its vicinity. In this case, a new food source is randomly generated at the beginning of the algorithm and the variable *limit* is set as zero. Algorithm 1 summarises the general steps to implement the BABC algorithm.

B. FIREFLY-INSPIRED

The Binary Firefly Algorithm (BFA), as the name suggests, is inspired by the social behaviour of fireflies [252]. We found two different implementation approaches, both using the continuous-continuous idea. There are around two million of species of fireflies, and the most part produces a short and rhythmic flashing light. These flashes are generated through a process of bioluminescence and exhibit two specific goals: a) to attract other fireflies and b) to attract a potential prey [127]. Based on this natural behaviour, the Firefly Algorithm was developed according to three idealised rules [6]:

- i) fireflies are unisex;
- ii) the degree of attractiveness of a firefly is proportional to its brightness. Hence, considering any two flashing fireflies, the less bright moves towards, the brighter. If two fireflies have the same brightness they move randomly;
- iii) the brightness of a firefly is proportional to its fitness.

To implement the BFA, each simple agent is a firefly $\bar{x}_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,d}, \dots, x_{i,D}\}$. The variation of the attractiveness is a vector β where the dimensions are proportional to the distance between the fireflies i and j . Each dimension is calculated separately as shown in (11):

$$\beta_d = \beta_0 e^{-\gamma r_{ij,d}^\gamma}, \quad (11)$$

where $\beta_0 \in [0, 1]$, $\gamma \in [0, 10]$ is the absorption coefficient, $n \geq 1$ is a user defined coefficient and the vector \mathbf{r}_{ij} has all positions equals to “-1,” “0” or “1” as presented in (1).

Algorithm 1 BABC Pseudocode

```

Initialise randomly all food sources positions  $\mathbf{x}_i^0$ ;
Evaluate the nectar amount (fitness) of them;
while stop criterion is not reached do
  //–Employed Bees–
  for each employed bee  $i = 1$  to  $N$  do
    Produce a new food source  $\mathbf{v}_i$  in the neighbourhood
    of  $\mathbf{x}_i$ ;
    if  $fit(\mathbf{v}_i) > fit(\mathbf{x}_i)$  then
      Replace  $\mathbf{x}_i$  by  $\mathbf{v}_i$  in the next generation (greedy
      selection);
       $trial = 0$ 
    else
       $trial = trial + 1$ 
    end if
  end for
  Calculate the probability  $p_i$  for each food source
  using (10);
  //–Onlooker Bees–
   $count\_onlooker = 1$ 
  while  $count\_onlooker \leq N$  do
    Draw  $r = rand(0, 1)$ 
    if  $r < p_i$  then
      if  $fit(\mathbf{v}_i) > fit(\mathbf{x}_i)$  then
        Replace  $\mathbf{x}_i$  by  $\mathbf{v}_i$  in the next generation (greedy
        selection);
         $trial = 0$ 
      else
         $trial = trial + 1$ 
      end if
    end if
     $count\_onlooker = count\_onlooker + 1$ 
  end while
  //–Scout Bees–
  if  $max(trial) > limit$  then
    Find a new food source randomly;
  end if
end while
Output the best solution

```

Then, the position is updated by (12):

$$x_{i,d}^{t+1} = x_{i,d}^t + \beta_d(x_{j,d}^t - x_{i,d}^t) + \alpha \left(rand(0, 1) - \frac{1}{2} \right) \quad (12)$$

in which α is a parameter defined by the user a priori, \mathbf{x}_j^t is some brighter firefly (higher fitness), $(x_{j,d} - x_{i,d})$ is equivalent to $r_{ij,d}$ and follow the rule defined in (1) too. Note that, in (12), the second component is related to attraction while the third one is a random step.

To determine the value of each dimension, the sigmoid function can be applied as in (2), creating $S(x_{i,d}^{t+1})$. Therefore, the determination of the binary position is given by (3).

Other possibility described in [6] is the application of the hyperbolic tangent function instead of the sigmoid, using (4)

and having $|tanh(x_{i,d})|$. Then, (3) is changed replacing $S(x_{i,d})$ by $|tanh(x_{i,d})|$ in the inequality.

Liu et al. [45] introduces two important modifications in BFA. They changed (1) by calculating the distance between two fireflies using their hamming distance - the total number of different bits of $x_{i,d}$ and $x_{j,d}$ -, which is a scalar, as in (13):

$$r_{ij}^{Hamm} = \sum_{d=1}^D (x_{i,d} \otimes x_{j,d}) \quad (13)$$

where \otimes is the logical function *XOR* which sets the output as “0” if the bits in the dimension d of both vectors are the same and “1” if they are different. Hence, β becomes a scalar too, being unique for all dimensions in (12).

Another difference between the proposals is that in (12) they replace $\left(rand(0, 1) - \frac{1}{2} \right)$ by a random generated vector ε . Finally, the steps of the Binary Firefly can be summarised in Algorithm 2.

Algorithm 2 BFA Pseudocode

```

1: Initialise randomly the parameters: number of fireflies  $N$ ,
   the initial position of the fireflies as binary vectors,  $\gamma$ ,  $\beta_0$ ,
    $\alpha$  and the stop criterion;
2: while stop criterion is not reached do
3:   for  $i = 1$  to  $N$  do
4:     for  $j = i + 1$  to  $N$  do
5:       if  $fit(\mathbf{x}_j) < fit(\mathbf{x}_i)$  then
6:         Calculate the attractiveness  $\beta$ 
7:         Update the position according to (12)
8:         Evaluate the fitness of each firefly
9:         Transform the position into a binary vector
10:      end if
11:    end for
12:    Ranking the fireflies according to their fitness and
    find the best solution;
13:  end for
14: end while
15: Output the best solution

```

C. FLOWER POLLINATION-INSPIRED

The Flower Pollination Algorithm (FPA) is a nature-inspired population-based algorithm which, depending on the definition, might not fit on the swarm group. The inspiration of the algorithm is different from the most swarm methods because it does not come from the behaviour of a collective of groups of animals [253]. However, as in the GSA case, the communication between the agents are cooperative, instead of generational or competitive.

The Binary Flower Pollination Algorithm (BFPA) was inspired in the natural pollination process of flowering plants and is the binary counterpart of the FPA introduced by Yang [254]. This algorithm is a case of the Continuous-Continuous method since the solution is first calculated as a real vector and then mapped into a binary

string. At the same time, the process of changing the position is performed using real numbers.

The main topics addressed are the optimal reproduction and the survival of the best-adapted plant. In this way, the algorithm follows four basic rules [46], [128]:

- i) Biotic cross-pollination is considered as global pollination. Also, pollen-carrying pollinators move to obey Lévy flights;
- ii) Self or abiotic-pollination are viewed as local pollination;
- iii) Pollinators, such as insects, can develop flower constancy. It means that the reproduction probability is proportional to the similarity of the two flowers involved;
- iv) The interaction or switching of local and global pollinations are controlled by the switching probability $p \in [0, 1]$. Due to the physical proximity and other factors, like the wind, local pollination has a more significant fraction p .

The rules *i* and *iii* are related to the global pollination. In this case, the pollens from the flowers are carried by pollinators (insects, wind, etc.) allowing the pollens to travel long distances. This process is described by (14):

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \alpha L(\lambda)(\mathbf{gbest} - \mathbf{x}_i^t) \quad (14)$$

where

$$L(\lambda) = \frac{\lambda \cdot \Gamma(\lambda) \cdot \sin(\lambda)}{\pi} \cdot \frac{1}{s^{1+\lambda}} \quad (15)$$

in which s is the step size and $s \gg s_0 > 0$, \mathbf{x}_i^t is the pollen i at iteration t , \mathbf{gbest} is the best position achieved so far, α is the variable which controls s , $L(\lambda)$ is the Lévy flight step size (strength of the pollination), $\Gamma(\lambda)$ stands for the gamma function, being $\lambda \in [1, 2]$.

In (15), observe that $\Gamma(\lambda)$ distribution is valid for large steps $s > 0$ and s_0 is suggested be 0.1. Indeed, some papers omits α , as in [129]. Nevertheless, the local pollination (rule *ii*) is defined by (16):

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \varepsilon(\mathbf{x}_j^t - \mathbf{x}_i^t) \quad (16)$$

being \mathbf{x}_j^t and \mathbf{x}_i^t the pollen of different flowers of the same species. Finally, rule *iv* is addressed to mimic the local and the global pollination, the switching probability p .

Following the basic rule proposed by the first BPSO, Rodrigues *et al.* [46] suggests again the application of the sigmoid function to transform the pollen in a binary vector, using (2) to generate $S(x_{i,d}^t)$, and the new position is given by (3). Algorithm 3 present the complete steps to implement BFPA.

The work from Dahi *et al.* [41] discusses the application of five techniques to map the continuous solution into a binary string: nearest-integer, normalisation, angle modulation, search process and the traditional sigmoid function. They conclude that the V-Shaped proposals can achieve better results.

Algorithm 3 BFPA Pseudocode

```

Initialise the  $N$  flower/pollen gametes randomly
Evaluate the population and define the best initial solution
gbest
Determine  $p \in [0, 1]$  as the switch probability
while stop criterion is not reached do
  for each pollen gamete do
    if  $\text{rand} < p$  then
      // Global pollination
      Draw a ( $D$ -dimensional) step vector  $L$  which obeys
      a Lévy flight distribution, as in (15);
      Proceed to the global pollination using (14)
    else
      // Local pollination
      Draw  $\varepsilon \sim U[0,1]$ 
      Choose two pollen gametes  $j$  and  $i$  randomly
      Perform the local pollination using (16)
    end if
  Transform the continuous solution into a binary vector.
  Evaluate the current solution
  if newly-generated solution is better then
    Proceed a greedy selection, replacing  $\mathbf{x}_i^t$  by  $\mathbf{x}_i^{t+1}$ 
  end if
  end for
  Update gbest
end while
Output the best solution

```

D. BIRD-INSPIRED

Undoubtedly, the Particle Swarm Optimization (PSO) is the most known and used swarm-based algorithm in the literature, proposed in 1995 by Kennedy and Eberhart [14]. The biological metaphor that inspired the algorithm was the collective intelligence of flocks of birds or school of fishes, simulating their social behaviour.

The BPSO is characterised by simple rules of information sharing between individual agents. The agent is called a particle, and a population of particles is named as a swarm. Each of them is assumed to be a location in the multi-dimensional search space or a candidate solution for the addressed problem. As usual, a particle is associated with a performance measure, the fitness.

Another critical remark is that the particle's position is changed based on its best position achieved so far (self-experience) and the best position found by some particle in its neighbourhood (the group experience). Sometimes, the entire swarm can be defined as the neighbourhood. Hence, an emergent complex global behaviour arises. The general steps of the BPSO are described in Algorithm 4.

In our search, we found just a few papers which address the Continuous-Continuous paradigm. In this case, the velocity and the position of the particles are vectors containing real

Algorithm 4 BPSO Pseudocode

```

1: Initialise all particle's position  $\mathbf{x}_i^0$  randomly with "0" and
   "1" as the values
2: Initialise the particle's velocity  $v_i^0$ 
3: Initialise the particles' best-known position  $\mathbf{pbest}_p^0$  with
   their initial position
4: Be  $fit(\mathbf{x}_i)$  the fitness of particle  $i$ : evaluate the fitness of
   the whole swarm
5: Set  $\mathbf{gbest}^0$  as the position of particle which has the best
   fitness
6: while stop criterion is not reached do
7:   for each particle  $i = 1$  to  $N$  do
8:     for each particle dimension  $d = 1$  to  $D$  do
9:       the particle's position
10:      Update particle's fitness  $fit(\mathbf{x}_i)$ 
11:      if  $fit(\mathbf{x}_i) < \mathbf{pbest}_i$  then
12:        Update the particle's best known position:
13:         $\mathbf{pbest}_i = \mathbf{x}_i$ 
14:      end if
15:      if  $fit(\mathbf{x}_i) < \mathbf{gbest}$  then
16:        Update the swarm's best known position:
17:         $\mathbf{gbest} = \mathbf{x}_i$ 
18:      end if
19:    end for
20:    Update the particle's velocity
21:  end for
22:  Output the best solution
23: end while

```

numbers. The conversion into binary strings occurs in the particles' position immediately before the fitness evaluation.

Two papers from Yassin *et al.* use the original PSO with both position and velocity as real vectors [47], [48]. They state that particles' positions $x_{i,d}^{cont}$ are the probabilities to flip a bit in a binary string. Therefore, the binary positions are generated from the continuous position using (17):

$$x_{i,d}^{t+1} = \begin{cases} \bar{x}_{i,d}^t & \text{if } x_{i,d}^{t,cont} \geq 0.5 \\ x_{i,d}^t & \text{otherwise} \end{cases} \quad (17)$$

where $\bar{x}_{i,d}^t$ means flip the bit in dimension d of $x_{i,d}$.

Babu *et al.* [49] proposed a similar way to solve the phasor measurement unit (PMU) problems. The normalisation method [41], [50] is another way to use the current paradigm. In this sense, the continuous positions $x_{i,d}^{t,cont}$ are transformed into a probability $x_{i,d}^{t,prob}$ as in (18):

$$x_{i,d}^{t,prob} = \frac{x_{i,d}^{t,cont} - x_{min}}{x_{max} - x_{min}} \quad (18)$$

being x_{max} and x_{min} the maximum and minimum values of the continuous solution $x_{i,d}^{t,cont}$, respectively. Then, the binary vector is created using (19):

$$x_{i,d}^{t+1} = \begin{cases} 1 & \text{if } x_{i,d}^{t,prob} \geq 0.5 \\ 0 & \text{otherwise} \end{cases} \quad (19)$$

Binary Coupled Spring Forced Multiagent Coordination Optimization (BCSFMCO) is an algorithm developed by Zhang and Hui [255]. They mixed a communication topology of PSO and multi-agent consensus protocols from control theory to create this method. The problems addressed as a case study are benchmark tasks and topology design for multi-agent formation control. The comparative analysis includes some PSO versions. The same authors developed the Binary Hybrid Multiagent Swarm Optimization Algorithm (BHMSO) [256], inspired by multi-agent consensus protocols from control theory. Benchmark problems were addressed and solved, being this proposal superior to BPSO versions.

APPENDIX B BINARY-CONTINUOUS ALGORITHMS

A. BEE-INSPIRED

There are different ways to represent solutions and displacements for bee-inspired algorithms. The approach from Lu *et al.* [36] uses an initial position \mathbf{x}_i in the binary space. However, the new food source \mathbf{v}_i initially is in the real domain - a Binary-Continuous representation.

The food sources are randomly generated by $randbin(0, 1)$ and \mathbf{v}_i^{cont} is calculated directly from (8). The resultant vectors \mathbf{v}_i^{cont} have the elements in the range $[-1, 1]$. Then, they use the hyperbolic tangent function $tanh(v_{i,d}^{cont})$ described in (4) to perform the mapping process. The binary \mathbf{v}_i is given by (20):

$$v_{i,d} = \begin{cases} 1, & \text{if } rand(0, 1) \leq |tanh(v_{i,d}^{cont})| \\ 0, & \text{otherwise.} \end{cases} \quad (20)$$

After \mathbf{v}_i be determined, a greedy selection is applied between it and \mathbf{x}_i , and those with better fitness value remains in the next iteration. A similar strategy was used by Wei *et al.* [54], but instead of the direct application of (20), they suggest to adopt the function *round*, or the nearest integer, to generate \mathbf{v}_i from \mathbf{v}_i^{cont} . As the employed bees, the onlooker and scout bees also apply the transfer function to their positions.

1) ANT-INSPIRED

The Binary Ant Colony Optimization (BACO) is a binary version of the previous ACO algorithm introduced by Dorigo [55], developed to solve integer problems like scheduling or routing [55], [56]. It is inspired by the behaviour of ant colonies searching for the shortest path to a food source [60]. In nature, ants randomly select a way to reach some food source. On finding it, they release pheromone trails on their way back to the nest as a manner to communicate with other members of the colony [116], [257]. The pheromone evaporates over time, and its concentration decreases with the path length. The ants are more likely to follow the trails that present the highest pheromone concentrations [59], [61].

BACO differs from several swarm-based algorithms because the optimum solution stands in the environment

(trails) instead of in the agent [116]. The movements of the ants occur in a graph that the vertices (nodes) consist of bits “0” and “1,” or the state transition of every bit [58]. Algorithm 5 presents the steps of the BACO. At each iteration, an ant travels all nodes to build a candidate solution. The ant departs from a randomly selected node and travels through the digraph along the arcs. Its trace will generate a binary string with D bits and, by this means, the colony constructs a group of candidate solutions. The ants perform D walks to form a complete solution. Figure 7 present the search space of an ant.

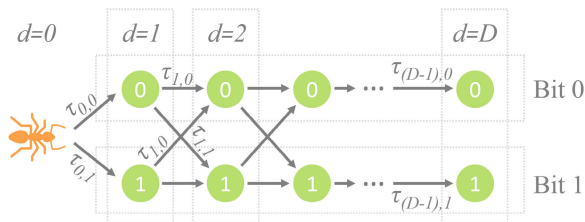


FIGURE 7. The BACO search space.

Note that changing (or flipping) a bit is also called *state transition* which means that a state 0 becomes a state 1 or vice versa [57]–[59]. The initial concentration of pheromone is usually the same in all edges. The index of an agent is i . Here, $d, 0$ or $d, 1$ are the edges which links the current node d , where the ant is located, to the next node $d + 1$. Some authors also consider the next node as a variable.

During the moving, each ant decides the next node (bit 0 or bit 1) depending on the amount of pheromone on the path and the visibility from the current node to the next. Therefore, the probability to go to “0” is calculated as in (21) and to “1” is given by (22) [120]:

$$p_{d,0}^i = \frac{[\tau_{d,0}]^\alpha \cdot [\eta_{d,0}]^\beta}{[\tau_{d,0}]^\alpha \cdot [\eta_{d,0}]^\beta + [\tau_{d,1}]^\alpha \cdot [\eta_{d,1}]^\beta} \quad (21)$$

and

$$p_{d,1}^i = 1 - p_{d,0}^i \quad (22)$$

being $\tau_{d,0}$ the *artificial pheromone trail* or the pheromone density of the side which leads to “0” and $\tau_{d,1}$ leads to “1,” $\eta_{d,0}$ and $\eta_{d,1}$ are the visibility densities of each edge and α and β are the relative importance of the pheromone and the visibility, respectively.

The majority of the selected papers in this work mention a heuristic function, which can be seen as the visibility densities in some problems. However, they disregard these parameters in their applications, as in [60] and [57]. Other works, like [61] and [58], define this probability without mentioning the visibility. In both cases, (21) can be rewritten as (23) [61]:

$$p_{d,0}^i = \frac{[\tau_{d,0}]^\alpha}{[\tau_{d,0}]^\alpha + [\tau_{d,1}]^\alpha} \quad (23)$$

The update of the pheromone in each edge at iteration t is given by (24) and (25), respectively:

$$\tau_{d,0}^{t+1} = (1 - \rho)\tau_{d,0}^t + \Delta\tau_{d,0}^t \quad (24)$$

$$\tau_{d,1}^{t+1} = (1 - \rho)\tau_{d,1}^t + \Delta\tau_{d,1}^t \quad (25)$$

in which $\rho \in [0, 1]$ is the evaporation rate and $\Delta\tau_{d,gbest}^t$ is the incremental amount of pheromone, calculated via (26) [120]:

$$\Delta\tau_{d,gbest}^t = \begin{cases} \frac{1}{fit(\mathbf{gbest})} & \text{if the arc from } d \text{ to } (0 \text{ or } 1) \\ & \text{is in the trace} \\ 0 & \text{otherwise} \end{cases} \quad (26)$$

where $fit(\mathbf{gbest})$ is the best fitness value.

The literature presents some important variations in the calculation of the parameters of the algorithm. Kuo et al. [61] introduce a convergence factor $cf^t \in [0, 1]$, calculated via (27). This step is performed after the update in the pheromone trail.

$$cf^t = \frac{\sum_{d=1}^D |\tau_{d,0}^t - \tau_{d,1}^t|}{D} \quad (27)$$

This factor is proportional to the difference between $\tau_{d,0}$ and $\tau_{d,1}$. The authors suggest that when the cf is near to 1 after t iterations, the ants are trapped in a local optimum. Therefore, pheromone values are reinitialised, and the algorithm is restarted. Then, the calculation of the pheromone update is given by (28) and (29).

$$\tau_{d,0}^{t+1} = (1 - \rho)\tau_{d,0}^t + \rho S_{d,0}^t \quad (28)$$

and

$$\tau_{d,1}^{t+1} = (1 - \rho)\tau_{d,1}^t + \rho S_{d,1}^t \quad (29)$$

in which

$$S_{d,gbest}^t = \begin{cases} w_{ib}s_{ib} + w_{rb}s_{rb} \\ + w_{gb}s_{gb} & \text{if the arc from } d \text{ to} \\ & (0 \text{ or } 1) \text{ is in the trace} \\ 0 & \text{otherwise} \end{cases} \quad (30)$$

where s_{ib} is the best solution achieved so far, s_{rb} is the best solution achieved during the current iteration and s_{gb} the best solution revealed since the last re-initialisation of the pheromone values. The variables w weights the importance of each component of $S_{d,gbest}^t$.

2) BAT-INSPIRED

The Bat Algorithm (BA) mimics the collective intelligence from a group of bats [39]. The Binary BA (BBA) is the adaptation of the original continuous BA, introduced by Yang in 2010 [62], to deal with binary variables. In Algorithm 6, we describe all steps to be followed in the BBA.

The key characteristic of bats is the advanced capability of echolocation. The species are divided into two subtypes: megabats and microbats. The natural echolocation search mechanism of the microbats is the main inspiration of the BA algorithm [123]. The echolocate via ultrasonic pulses produces the echo, which is used to define the location, the exact distance and the measurements and qualities [122].

Algorithm 5 BACO Pseudocode

```

1: Initialise randomly the pheromones trails with small positive values
2: while stop criterion is not reached do
3:   for all ant do
4:     Compute transition probabilities using (21) and (22)
5:     for all dimension do
6:       Draw where to go using the probabilities calculated;
7:     end for
8:   end for
9:   Update pheromone trails
10:  Evaporate pheromone trails
11: end while
12: Output the best solution (the way with the maximum pheromone levels)

```

Using the echolocation, the bats can differ the food (or a pray) from background barriers. An artificial bat flies randomly and presents a position (\mathbf{x}_i), a velocity (\mathbf{v}_i), a fixed frequency f_{min} and a loudness A_0 to search preys. The loudness varies from a significant (positive) value A_0 to a minimum constant number A_{min} . As in nature, the artificial bat can automatically tune the frequency of its emitted pulses and adjust the rate of pulse emission per , depending on the proximity of its target (notice that some papers name per as r). The frequency F of a bat i is defined according to (31):

$$F_i = F_{min} + \beta(F_{max} - F_{min}) \quad (31)$$

where F_{max} and F_{min} are the limits in which the frequency must lie and $\beta = rand(0, 1)$. The displacement of an agent is defined based on its velocity, described in (32):

$$\mathbf{v}_i^{t+1} = \mathbf{v}_i^t + F_i(\mathbf{x}_i^t - \mathbf{gbest}) \quad (32)$$

being \mathbf{gbest} the best solution obtained so far, as usual.

According to Amine *et al.* [39], the new position of a bat is defined using the sigmoid function applied to the velocity using (2), generating $S(v_{i,d}^t)$. Then, the position is given by (3). In the same way, Basetti *et al.* [63] proposed the use of the V-shaped transform function to update the position based on the velocity. They addressed (5) generating $V(v_{i,d}^t)$ and the position is updated according to (6). A local search scheme is evaluated using the loudness A and pulse emission rate per , both calculated using (33) and (34), respectively:

$$A_i^{t+1} = \alpha A_i^t \quad (33)$$

where, in general, $A_i^0 \in [1, 2]$ and α is a constant and

$$per_i^{t+1} = per_i^0 [1 - e^{-\gamma t}] \quad (34)$$

being $per_i^0 \in [0, 1]$ and γ a constant. These two quantities are used to determine if a local search will be performed during some iteration.

In 2018, a novel binary bat algorithm (NBBA) was proposed to solve the 0-1 knapsack problem [64]. Different from

Algorithm 6 BBA Pseudocode

```

1: Initialise randomly the bat population ( $\mathbf{x}^0$ ), velocities  $\mathbf{v}_i$ , pulse frequency  $F$ , pulse rates  $per_i$ , and loudness  $A_i$ 
2: Evaluate the bats calculating their fitness
3: Set  $\mathbf{gbest}$  as the particle with the lowest fitness
4: while stop criterion is not reached do
5:   for all bat do
6:     Be  $\mathbf{x}_i$  the current position of the bat
7:     if  $rand > per_i$  then
8:       Generate a local solution ( $\mathbf{x}_i^{new}$ ) around  $\mathbf{gbest}$  flipping one of its dimensions
9:     else
10:      Update velocity using (32)
11:      Adjust frequency by means of (31)
12:      Apply a binary transformation method to generate  $\mathbf{x}_i^{new}$ 
13:    end if
14:    Evaluate  $fit(\mathbf{x}_i^{new})$ 
15:    if  $rand < A_i$  and  $fit(\mathbf{x}_i^{new}) \leq fit(\mathbf{x}_i)$  then
16:      Set  $\mathbf{x}_i$  as  $\mathbf{x}_i^{new}$ 
17:      Apply 33 to increase pulse emission rate
18:      Apply 34 to decrease loudness
19:    end if
20:  end for
21:  if  $fit(\mathbf{x}_i^{new}) \leq fit(\mathbf{gbest})$  then
22:     $\mathbf{gbest} = \mathbf{x}_i^{new}$ 
23:     $fit(\mathbf{gbest}) = fit(\mathbf{x}_i^{new})$ 
24:  end if
25: end while

```

BA [30], NBBA applies rough set scheme (RSS), one-to-one strategy and multi-V-shaped transfer function. Using these three mechanisms, the diversity and convergence outcome the BBA.

In 2020, Zhang [65] published another version of binary bat-inspired algorithm called Binary Cooperative Bat searching Algorithm (BCBA) in which four different transfer functions are tested, and an optimal topology is proposed as a thread-off between effectiveness and convergence.

3) CAT-INSPIRED

The Binary Cat Swarm Optimization (BCSO) is inspired in the ability of domestic cats to hunt and to stay alert to possible dangers [66]. According to Sarafi *et al.* [28], cats spend most of their time resting when they are awake. In this case, they change their position carefully and slowly or do not move. However, for the rest of the time, the cats are tracing targets [125]. The first proposal that we found was from Chu *et al.* [258], which is adequate to solve continuous problems.

The BCSO defines two modes of behaviour: seeking mode and tracing mode. In the first, their motions are slow and near the original position. Biologically, it corresponds to the

resting state of the cats. In the mode, a cat moves according to its velocities.

The mixture ratio MR is a parameter that defines the percentage of cats that will perform the tracing mode, and the percentage of cats will be in the seeking mode. In each iteration, the cats are randomly selected based on the MR . The complete steps of the BCSO are summarized in Algorithm 7, and we describe both steps of the BCSO separately.

a: SEEKING MODE

The cats' displacement is usually slow and near the current position. A cat could look around and seek the next position. In this mode, there are four factors, which are [66]:

- i) *seeking memory pool (SMP)*: the size of seeking memory for each cat - a positive integer number $SMP \in (1, N)$. It can be seen as the number of identical copies clones generated from a given solution;
- ii) *probability muting operation (PMO)*: parameter which defines the mutation probability for the selected dimensions. In this case, $0\% < PMO \leq 100\%$;
- iii) *counts of dimension to change (CDC)*: present how many dimensions will be selected to possible mutation. Note that $1 \leq CDC < D$. Usually, it is presented as a percentage;
- iv) *self-position considering (SPC)*: a boolean flag which defines whether the current position of a cat will be one of the candidates in the seeking mode to or not. Its state can be *true* or *not true*.

Then, the seeking mode is performed according to the steps described below [69]:

Step 1: Consider the *SPC* flag. If it is set as *true*, produce $SMP - 1$ copies of the current position of each cat and take the current position as one of the candidates. If *SPC* flag is *not true*, make SMP copies of the current position of each cat;

Step 2: For each copy, select CDC dimensions and mutate (flip) them according to *PMO*;

Step 3: Evaluate all cats, calculating their fitness;

Step 4: Calculate the selecting probability of each cat according to (35) [67] and apply a *roulette wheel* method and replace the current position by the selected one:

$$P_i = \frac{fit_i - fit_b}{fit_{max} - fit_{min}} \quad (35)$$

where fit_i is the fitness of the i -th cat, $fit_b = fit_{max}$ if we are working on a minimisation problem and $fit_b = fit_{min}$ if we are working on a maximisation problem. Some authors [66], [68] propose a greedy selection between the copies, instead of applying the *roulette wheel*.

b: TRACING MODE

In tracing mode, a cat is metaphorically tracing targets, moving towards the best solution. Therefore, the next movement is determined based on the cat's velocity and the best position found by the rest of the swarm [67]. The notion of velocity in the BCSO is different from the PSO. In this case, the velocity is the probability of a dimension being changed (flipped).

This mode presents the following steps [68]. First, select the best-positioned cat (highest fitness) and name it as **gbest**. Define \mathbf{v}_i^0 and \mathbf{v}_i^1 as the vectors representing the *intermediate* velocities of each cat. The first is the probability of the bits changes to "0," and the other is the probability of flipping to "1." Note that they do not complement [69]. The update process for each dimension happens according to (36).

$$\begin{aligned} v_{i,d}^1 &= wv_{i,d}^1 + d_{i,d}^1 \\ v_{i,d}^0 &= wv_{i,d}^0 + d_{i,d}^0 \end{aligned} \quad (36)$$

where w is the inertia weight and $d_{i,d}^1$ and $d_{i,d}^0$ are temporary values, updated using the **gbest** as guide, as in (37).

$$d_{i,d}^1 \text{ and } d_{i,d}^0 = \begin{cases} r_1 c_1 \text{ and } -r_1 c_1 & \text{if } gbest_d = 1 \\ -r_1 c_1 \text{ and } r_1 c_1 & \text{if } gbest_d = 0 \end{cases} \quad (37)$$

being $r_1 = rand(0, 1)$ and c_1 a constant defined by the user. According to the position of cat \mathbf{x}_i , its velocity (probability of change d -th bit of i -th cat) is calculated by (38)

$$v'_{i,d} = \begin{cases} v_{i,d}^1 & \text{if } x_{i,d} = 0 \\ v_{i,d}^0 & \text{if } x_{i,d} = 1 \end{cases} \quad (38)$$

in which, the velocity is bounded by $[v_{min}, v_{max}]$. In the last step, update the positions of the swarm. As often happens, it is necessary to apply $v'_{i,d}$ on a sigmoid function defined in (2), generating the values $S(v'_{i,d})$. In the proposal from Sharafi et al. [28], the new value of each dimension is updated as in (39)

$$x_{i,d} = \begin{cases} gbest_d & \text{if } rand(0, 1) < S(v'_{i,d}) \\ x_{i,d} & \text{otherwise} \end{cases} \quad (39)$$

On the other hand, Mohamadeen et al. [66] suggest another way to perform this using (40).

$$x_{i,d}^{t+1} = \begin{cases} \bar{x}_{i,d}^t & \text{if } rand(0, 1) < S(v'_{i,d}) \\ x_{i,d}^t & \text{otherwise} \end{cases} \quad (40)$$

where $\bar{x}_{i,d}^t$ means flip the bit in dimension d or the 1's complement of $x_{i,d}$.

4) GRAVITATIONAL SEARCH ALGORITHM

The Binary Gravitational Search Algorithm (BGSA) was inspired by Newton's Laws of motion and gravity and is based on the metaphor of gravitational interaction between masses [70]. Note that the algorithm is not a classic swarm-based approach, since it is not inspired in the collective behaviour of groups of animals [130]. However, we decide to discuss the method since the mechanisms of collaboration between the agents are cooperative, as in the swarm approaches [37].

The BGSA is a nature-inspired proposal, introduced by Rashedi et al. [51]. BGSA algorithm is presented in Algorithm 8. In this case, the displacement is a real vector, while the position is always a binary string. The agent is a mass $\mathbf{x}_i = [x_{i,1}, x_{i,2}, \dots, x_{i,d}, \dots, x_{i,D}]$, a binary string with D dimensions. The gravitational force f_{ij}^t which acts on mass

Algorithm 7 BCSO Pseudocode

```

1: Initialize randomly all cats positions and velocities, the
   SPC flag of every cat, and MR, SMP, CDC and PMO
2: while stop criterion is not reached do
3:   Select MR% of the cats to perform the tracing mode
   and consider that the other cats are in seeking mode
4:   Evaluate each cat and save gbest
5:   for all cat do
6:     if  $x_i^t$  is in seeking mode then
7:       if SPC flag is true then
8:         Produce as many as SMP – 1 copies of the
         present position of each cat and take the current
         position as one of the candidates
9:       else
10:        Make SMP copies of the present position of
        each cat
11:      end if
12:      for all copy do
13:        Select as many as CDC dimensions
14:        Randomly mutate this CDC dimensions
        according to PMO
15:      end for
16:      Evaluate the fitness of all copies
17:      Apply roulette wheel method, select one candi-
        date and replace the current position by its
18:    end if
19:    if  $x_i^t$  is in tracing mode then
20:      Calculate the intermediate velocities using (36)
        and (37)
21:      Update cat’s velocity utilising (38)
22:      Update cat’s position as a binary vector
23:    end if
24:  end for
25: end while
26: Output the best solution (gbest)

```

i due to mass j at iteration t and considering dimension d , is described by 41 [72]:

$$f_{ij,d}^t = G^t \frac{M_i^t M_j^t}{R_{ij}^t + \epsilon} (x_{i,d}^t - x_{j,d}^t) \quad (41)$$

where M_i and M_j are the masses of agents i and j , respectively (updated according (47) ahead), ϵ is a small positive constant, R_{ij}^t is the distance between these agents, which follows the subtraction method described in (1) [73], and G_t is the gravitational constant, calculated by (42):

$$G^t = G^0 e^{\frac{\alpha t}{T_{max}}} \quad (42)$$

in which G^0 is the initial gravitational constant, T_{max} is the total number of iterations and α is an exponential decay constant. Observe that G gradually decreases over time. It is possible to define the distance R_{ij}^t as two other ways. The first is the Hamming distance, calculated in (43), similar

Algorithm 8 BGSA Pseudocode

```

1: Initialise randomly the N masses/agents
2: Evaluate the fitness and define best and worst using (49)
   and (48) respectively.
3: while stop criterion is not reached do
4:   Update G using (41)
5:   for all mass do
6:     Evaluate the fitness
7:     Calculate the gravitational mass by means of (47)
        and (48)
8:     Calculate the acceleration using (45) and (44)
9:     Update velocity utilising 46
10:    Update position using some mapping method to
        have a binary vector
11:  end for
12:  update best and worst using (49) and (50) respectively.
13: end while
14: Output the best solution

```

to (13) [37]:

$$R_{ij}^t = \sum_{d=1}^D (x_{i,d}^t \oplus x_{j,d}^t) \quad (43)$$

Another possibility is to divide the result of (43) by D , generating a normalised Hamming distance [71]. The total gravitational force on an individual i in the d -th dimension is the sum of all forces provided by the other masses, which is given by 44 [73]:

$$F_{i,d}^t = \sum_{j=1, j \neq i}^{N-1} rand(0, 1) f_{ij,d}^t \quad (44)$$

The acceleration $a_{i,d}^t$ of mass i in the dimension d at iteration t is given by (45):

$$a_{i,d}^t = \frac{F_{i,d}^t}{M_i^t} \quad (45)$$

and the velocity is updated using (46):

$$v_{i,d}^{t+1} = rand(0, 1) v_{i,d}^t + a_{i,d}^t \quad (46)$$

The last step updates the gravitational mass, which is done by (47):

$$M_i^t = \frac{m_i^t}{\sum_{j=1}^N m_j^t} \quad (47)$$

being

$$m_i^t = \frac{fit^t(\mathbf{x}_i) - worst^t}{best^t - worst^t} \quad (48)$$

where $fit^t(\mathbf{x}_i)$ is the current fitness of \mathbf{x}_i while the variables $best^t$ and $worst^t$ are the fitness of the best and worst masses, being selected according (49) and (50):

$$best^t = min(fit^t) \quad (49)$$

$$worst^t = max(fit^t) \quad (50)$$

We highlight the similarity between (48) and (35) from the BCSO. The first expression maps the velocity into a binary position vector and applies the sigmoid function described in (2), generating $S(v_{i,d})$ [72]. Then, the positions are updated using (3). Another possibility is discussed by Ji *et al.* [73] and by Nezamabadi [37] that performs the transformation by the use of (4) (creating $|tanh(v_{i,d})|$) and (6).

5) WOLF-INSPIRED

The Grey Wolf Optimizer (GWO) was inspired in the leadership hierarchy and hunting mechanism of grey wolves [52]. These animals have a strict social dominant hierarchy that is divided into alpha (α), betas (β), deltas (δ) and omegas (ω). The most dominant individual is the alpha, and the lowest one is the omega. The algorithm mimics the hunting mechanism of wolves that guided by the alpha, the pack recognise the location, encircle the location and attack the prey.

Emary *et al.* [7] introduced the algorithm in the binary domain proposing the Binary Grey Wolf Optimizer (bGWO). The general BGWO algorithm is shown in the Algorithm 9. They have proposed two different approaches to solve feature selection, bGWO1 and bGWO2. In bGWO1 the central update equation is shown in (51).

$$\mathbf{x}_i^{t+1} = Crossover(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) \quad (51)$$

where $Crossover(x_1, x_2, x_3)$ is a crossover between three solutions x_1, x_2 and x_3 that represent the effect of alpha, beta and delta in the current wolf, respectively. x_1, x_2 and x_3 effect is calculated using the (52).

$$x_{\{1,2,3\}}^d = \begin{cases} 1 & \text{if } (x_{\{\alpha,\beta,\delta\}}^d + bstep_{\{\alpha,\beta,\delta\}}^d) \geq 1 \\ 0 & \text{otherwise} \end{cases} \quad (52)$$

where $x_{\{\alpha,\beta,\delta\}}^d$ is the position of the α, β or δ wolves in the dimension d and $bstep_{\{\alpha,\beta,\delta\}}^d$ is calculated using (53).

$$bstep_{\{\alpha,\beta,\delta\}}^d = \begin{cases} 1 & \text{if } cstep_{\{\alpha,\beta,\delta\}}^d \geq rand(0, 1) \\ 0 & \text{otherwise} \end{cases} \quad (53)$$

where $cstep_{\{\alpha,\beta,\delta\}}^d$ is calculated using (54).

$$cstep_{\{\alpha,\beta,\delta\}}^d = \frac{1}{1 + e^{-10(A_{\{\alpha,\beta,\delta\}}^d D_{\{\alpha,\beta,\delta\}}^d - 0.5)}} \quad (54)$$

where $A_{\{\alpha,\beta,\delta\}}^d$ and $D_{\{\alpha,\beta,\delta\}}^d$ are equations defined for GWO. $Crossover$ operator is defined as shown in (55).

$$x^d = \begin{cases} x_1^d & \text{if } rand(0, 1) < \frac{1}{3} \\ x_2^d & \text{if } \frac{1}{3} \leq rand(0, 1) < \frac{2}{3} \\ x_3^d & \text{otherwise} \end{cases} \quad (55)$$

where a_d, b_d, c_d are the binary values of d -th dimension for each parameter.

Algorithm 9 BGWO Pseudocode

```

Initialise the  $N$  wolves randomly
Find the  $\alpha, \beta, \delta$  solutions based on fitness
while stop criterion is not reached do
  for all wolf do
    Update wolf position
  end for
  Update  $a, A$  and  $C$ 
  Evaluate the current position of individual wolves
  Update  $\alpha, \beta, \delta$ 
end while
Output the best solution

```

In bGWO2 is proposed a different approach where only the position is converted to binary using the (56).

$$x_d^{t+1} = \begin{cases} 1 & \text{if } sigmoid(\frac{x_1 + x_2 + x_3}{3}) \geq rand(0, 1) \\ 0 & \text{otherwise} \end{cases} \quad (56)$$

where x_d^{t+1} is the new binary position in dimension d at iteration t , and $sigmoid(a)$ is defined as follows:

$$sigmoid(a) = \frac{1}{1 + e^{-10(a-0.5)}} \quad (57)$$

Jiang *et al.* [75] have upgraded the convergence parameter (\bar{a}) as shown in (58) in order to improve the algorithm. Hu *et al.* [76] also proposed changes in the a parameter as described in (59) to improve feature selection using BGWO.

$$\bar{a} = 2 - 2 \times \left[\frac{(e^{\frac{it}{m}} - 1)}{(e - 1)} \right] \quad (58)$$

$$a = 2 \times \frac{it}{m} \quad (59)$$

where e is Euler number and m is the maximum number of iterations.

Alzubi *et al.* [77] proposed a modified binary grey wolf optimizer (MBGWO). The proposal used omega information to reduce the impact rate of the best solutions from 0.33 (α, β and δ) to 0.25 (α, β, δ and ω). In this new approach, the new crossover operator is defined as shown in (60).

$$x^d = \begin{cases} x_1^d & \text{if } rand(0, 1) < \frac{1}{4} \\ x_2^d & \text{if } \frac{1}{4} \leq rand(0, 1) < \frac{2}{4} \\ x_3^d & \text{if } \frac{2}{4} \leq rand(0, 1) < \frac{3}{4} \\ x_4^d & \text{otherwise} \end{cases} \quad (60)$$

Luo *et al.* [78] introduced a new mechanism to highlight the leadership hierarchy by using a differentiated position updating strategy. In this approach, the leader wolves can only move if the new position is better than the previous one and other wolves unconditionally move to their new position.

A quantum-inspired binary grey wolf optimizer (QI-BGWO) was proposed by Srikanth *et al.* [79]. The usage

of quantum concepts has improved the balance between exploration and exploitation in the problem.

6) BIRD-INSPIRED

Regarding the Binary PSO, because of a large number of papers, we found an elevated number of distinct proposals. Formally, consider a population of N particles (swarm). The agent i present its current position $\mathbf{x}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$ and a velocity $\mathbf{v}_i = (v_{i,1}, v_{i,2}, \dots, v_{i,D})$. In the majority of the versions of BPSO, while the position is a binary string, the velocity is a continuous (real) vector limited in the interval $[-v_{max}; +v_{max}]$, where v_{max} is defined by the user. The update of the velocity is performed according to (61):

$$v_{i,d}^{t+1} = wv_{i,d}^t + c_1 rand(0, 1)(pbest_{i,d}^t - x_{i,d}^t) + c_2 rand(0, 1)(gbest_d^t - x_{i,d}^t) \quad (61)$$

where w is the inertial weight, c_1 and c_2 are the cognitive and social coefficients, respectively, $pbest_{i,d}^t$ is the best position found by particle i until iteration t (the individual position that achieved the best performance index during the search process) and $gbest_d^t$ is the best position found by some neighbour of the same agent i accordingly the communication topology. The algorithm is often initialised by randomly spreading the particles over the search space. The same process is applied to generate the initial velocities, but, in some cases, the velocities are initialised equal to zero.

Some papers as in Phuangpornpitak and Tia [89], Unler and Murat [90] and Azad et al. [91] suggest a mechanism to limits the inertia weight in a linear decay way, as in (62):

$$w_t = w_{max} - \left(\frac{w_{max} - w_{min}}{t_{max}} \right) t \quad (62)$$

in which $w \in [w_{min}, w_{max}]$ and t_{max} is the maximum number of iterations [92]. The concept of inertia weight was in fact introduced by Shi and Eberhart [74] to balance the local and global search.

Liu et al. [31] performed an extensive parameter analysis of the use of the parameter w for BPSO. Large values of inertia tend to display exploitation, and smaller values of inertia tend to display exploration. As exploration is usually recommended at the beginning of the iterations, a linearly increasing inertia weight is most likely to show better results. The opposite is suggested by Shi and Eberhart [74] for continuous optimisation, highlighting that even using the same main mechanisms of PSO, the fact of being binary or continuous optimisation or being different fitness functions can require a different balance of parameters.

Another idea is presented in Pookpant et al. [93] and Chanthaphavong and Chetty [95]. The proposals suggest that the accelerations coefficients (c_1, c_2) linearly changes over the iterations, like in (63) and (64), respectively:

$$c_1 = c_{1i} - \left(\frac{c_{1f} - c_{1i}}{t_{max}} \right) t \quad (63)$$

where c_1 changes from $c_{1i} = 2.5$ to $c_{1f} = 0.5$ and:

$$c_2 = c_{2i} - \left(\frac{c_{2f} - c_{2i}}{t_{max}} \right) t \quad (64)$$

in which c_2 is modified from $c_{2i} = 0.5$ to $c_{2f} = 2.5$. The initial values are suggestions of the authors and can be adapted to each problem [94]. To transform the elements of the velocity into a binary vector, they proposed the utilisation of a logistic sigmoid function, using the current velocity as a parameter in (2) to generate $S(v_{i,d})$. Therefore, the update of the position is performed according to (3).

On the other hand, Mirjalili et al. [96] and Kumar et al. [38] present a map of the V-shaped function in the velocity $|V(v_{i,d}^t)|$. Hence, the update in the binary position is given by (5) and (6).

Modified versions of the V-shaped transfer function is also present in the literature, such as MBPSO [88]. Jiang et al. proposed a binary version of PSO (NMBPSO) using a hybrid transfer function of S and V-shaped [87] outperforming BPSO because NMBPSO provides local search capability even at later iterations. In the same way, Minzu et al. [86] suggested the application of the $|\tanh(v_{i,d}^t)|$ from (4).

Souza et al. [85] applied an alternative equation using the modulus of a sigmoid function, generating $|S^{alt}(v_{i,d}^t)|$. This value replaces the probability $|\tanh(v_{i,d}^t)|$ in (6).

A modification called modified BPSO is showed in the papers from Menhas et al. [84] and Ko et al. [83]. In this case, the velocity is calculated without the inertia weight and the acceleration coefficients via (65).

$$v_{i,d}^{t+1} = v_{i,d}^t + rand(0, 1)(pbest_{i,d}^t - x_{i,d}^t) + rand(0, 1)(gbest_d^t - x_{i,d}^t) \quad (65)$$

and the position is updated according to (66)

$$x_{i,d}^{t+1} = \begin{cases} x_{i,d}^t & \text{if } 0 \leq v_{i,d}^t \leq a \\ pbest_{i,d}^t & \text{if } a < v_{i,d}^t \leq \frac{1+a}{2} \\ gbest_d^t & \text{if } \frac{1+a}{2} < v_{i,d}^t \leq 1 \end{cases} \quad (66)$$

where $a \in [0, 1]$ is static probability fixed as a constant, being the usual set $a = 0.5$. Note that the velocity is constrained in the interval $[0, 1]$.

The Novel BPSO is presented in the works of de Sousa et al. [82] and Puri and Hsiao [40]. The noticeable fact about this version is that it uses a velocity equation with similarities to the BCSO algorithm. To do so, it is necessary to define four temporary values: $d_{i,d}^{1,pbest}$ and $d_{i,d}^{0,pbest}$ that are related to **pbest**, and $d_{i,d}^{1,gbest}$ and $d_{i,d}^{0,gbest}$ regarding **gbest**. These values are calculated using (67) and (68) below:

$$d_{i,d}^{1,pbest} \text{ and } d_{i,d}^{0,pbest} = \begin{cases} r_1 c_1 \text{ and } -r_1 c_1 & \text{if } pbest_d = 1 \\ -r_1 c_1 \text{ and } r_1 c_1 & \text{if } pbest_d = 0 \end{cases} \quad (67)$$

considering $r_1 = rand(0, 1)$, c_1 a constant defined by the user and:

$$d_{i,d}^{1,gbest} \text{ and } d_{i,d}^{0,gbest} = \begin{cases} r_2 c_2 \text{ and } -r_2 c_2 & \text{if } gbest_d = 1 \\ -r_2 c_2 \text{ and } r_2 c_2 & \text{if } gbest_d = 0 \end{cases} \quad (68)$$

being $r_2 = rand(0, 1)$ and c_2 another user-defined constant.

Equation 69 defines the *intermediate* velocities:

$$\begin{aligned} v_{i,d}^0 &= wv_{i,d}^0 + d_{i,d}^{0,pbest} + d_{i,d}^{0,gbest} \\ v_{i,d}^1 &= wv_{i,d}^1 + d_{i,d}^{1,pbest} + d_{i,d}^{1,gbest} \end{aligned} \quad (69)$$

where w is the same inertia weight. Here, the final velocity can be defined as the probability of flipping the d -th bit of i -th particle, which is performed using (70):

$$v'_{i,d} = \begin{cases} v_{i,d}^1 & \text{if } x_{i,d} = 0 \\ v_{i,d}^0 & \text{if } x_{i,d} = 1 \end{cases} \quad (70)$$

Usually, the velocity is bounded by v_{max} . Then, the new position is given by (71) [40]

$$x_{i,d}^{t+1} = \begin{cases} \bar{x}_{i,d}^t & \text{if } rand < v'_{i,d} \\ x_{i,d}^t & \text{otherwise} \end{cases} \quad (71)$$

in which $\bar{x}_{i,d}^t$ means flip the bit in dimension d of $x_{i,d}^t$.

Puri and Hsiao [40] and Sousa et al. [82] mention that 71 can be replaced by the use of the sigmoid function $S(v'_{i,d})$ or the V-shaped proposal $V(v'_{i,d})$, as in (2) and (5), respectively. Then, one can apply (3) or (6), as discussed.

Siqueira et al. [81] introduced the Double Swarm BPSO, which was inspired in the original BPSO and the steps of the BCSO. The authors divide the swarm into two sub-swarms, named *swarm of mutation* and *displacement swarm*. A half part of the particles is placed in the *displacement swarm* using the roulette wheel and the rest stands in the *swarm of mutation*. In the swarm of mutation, NC clones are created, and NM dimensions are flipped. Again, it is performed the roulette wheel to select the winner, comparing the original agent and the clones. On the other hand, considering the displacement swarm, the particles fly in the search space, following the velocity described in (72).

$$v_{i,d}^{t+1} = wv_{i,d}^t + rand(0, 1)c_1(2gbest_d^t - x_{i,d}^t - 1) \quad (72)$$

in which c_1 is a user-defined variable.

The Competitive Swarm Optimizer (CSO) is also a kind of PSO in which the particles learn from randomly selected competitors and not from the global or the personal best position. The idea is to perform better in large-scale optimisation problems. A binary approach was introduced by Gu et al. [80]. In this work, the authors dealt with feature selection problems, considering the KNN classifier and benchmark problems.

APPENDIX C BINARY-BINARY ALGORITHMS

A. BEE-INSPIRED

Kiran and Gunduz propose the first Binary-Binary ABC in 2013, the binABC [97]. In this case, the authors suggested generating the bees as binary strings, being the bit value according to a probability p . Then, the positions of employed and onlookers are updated following (73).

$$v_{i,d} = x_{i,d} \otimes [\odot(x_{i,d} \otimes x_{j,d})] \quad (73)$$

where \otimes is the “XOR” operator and \odot works as the NOT gate: if it is less than 0.5, the result obtained is inverted.

Another approach is in work from Wei and Hanning [54]. Initially, they set $x_{d_{min}} = 0$, $x_{d_{max}} = 1$ and $rand(0, 1)$ is replaced by $randbin(0, 1)$, which means that each draw generates a bit 0 or 1. Similarly, Santana et al. [99] initialises the agents as 0 or 1 depending on $rand(0, 1)$ being smaller or equal/higher to 0.5. Depending on the problem, the initialisation can speed up the results, for instance, if the binary problem is trying to minimise, having the majority of dimensions starting with zeros is generally beneficial.

Then, we can rewrite (7) as (74). We observe that the computational cost of applying (74), the most applied one, is much smaller than applying (7).

$$x_{i,d} = randbin(0, 1). \quad (74)$$

Once the food sources are generated, each employed bee moves to its food source and finds a new one in its neighbourhood v_i . To do so, Jia et al. [98] have adapted the of the continuous space using logic gates operations generating (75):

$$v_{i,d} = x_{i,d} \otimes \phi_{i,d} \odot (x_{i,d} \oplus x_{j,d}) \quad (75)$$

where \otimes is the “XOR” operator, \odot is the “AND” operator, \oplus is the “OR” operator and $j, j \neq i$, a randomly selected food source. The variable $\phi_{i,d}$ is defined by (76):

$$\phi_{i,d} = \begin{cases} 1 & \text{if } rand(0, 1) < r \\ 0 & \text{otherwise} \end{cases} \quad (76)$$

where the parameter r controls the generating probability of “0” and “1.” The binary value $\phi_{i,d}$ will have more probability to be a binary number “0” when r takes a small value. Therefore, the new food source will be more close to the older under the “XOR” operation. When using this procedure, we work in a binary space [98].

In Santana et al. [99], the employed and onlookers bees update their positions differently. If the chosen food sources have higher fitness than the bees and the chosen dimensions do not have the same position, the bees will change their dimensions to the value found on the food sources - otherwise, the bees do not move. Santana et al. [99] also show the impact on the algorithm when taking into account different strategies.

Aytimur et al. [100] compare three versions of binary ABC: using the sigmoid function, using crossover technique from Genetic Algorithm and using a Boolean technique based on the XOR operator. Aytimur et al. [100] show that the binary version of BABC using crossover technique from Genetic Algorithm performed better than the others.

1) CAT-INSPIRED

A Binary-Binary approach is introduced by Siqueira et al. [101], named Boolean BCSO, which presents some important differences in comparison with the BCSO. Firstly, in the seeking mode, the *SPC* flag is suppressed.

Then, in the tracing mode, the new velocity $v_{i,d}^{t+1}$ is defined according to (77).

$$v_{i,d}^{t+1} = v_{i,d}^t \oplus r_1 \otimes (x_{best,d}^t \odot x_{i,d}^t) \quad (77)$$

being $r_1 = rand(0, 1)$, \oplus the “XOR” operator, \otimes the “AND” operator and \odot the “OR” operator. The position updates as (78).

$$x_{i,d}^{t+1} = x_{i,d}^t \odot v_{i,d}^{t+1} \quad (78)$$

Finally, the last difference is the allocation of the *MR* cats in the tracing mode using the roulette wheel method.

In 2020, Siqueira *et al.* [102] proposed a version of the BCSO called Simplified Binary Cat Swarm Optimization (SBCSO). SBCSO proposes a new velocity and position strategy that overcome binary versions of CSO, FSS, ABC, GA and PSO not only in the accuracy but also in the computational cost.

2) BIRD-INSPIRED

Suresh *et al.* [103] present an improved version of the BPSO to solve maintenance schedule problem, being this a Binary-Binary algorithm. The same proposal is presented by Sedighizadeh *et al.* [104].

In this case, each particle is a binary string, and its position and velocity are updated using binary digital operation (Boolean gates), according to (79) and (80), respectively:

$$x_{i,d}^{t+1} = x_{i,d}^t \odot v_{i,d}^{t+1} \quad (79)$$

$$v_{i,d}^{t+1} = r_1 \otimes (pbest_{i,d}^t \odot x_{i,d}^t) \oplus r_2 \otimes (gbest_d^t \odot x_{i,d}^t) \quad (80)$$

where d is the current dimension, “ \odot ” is the *AND*, “ \otimes ” is the *XOR* and “ \oplus ” is the *OR* operators, $r_1 = randbin(0, 1)$ and $r_2 = randbin(0, 1)$ are two binary integer numbers randomly generated.

An exciting remark comes from the works from Pu *et al.* [105], Bin *et al.* [106], Pirhayati and Mazlumi [108] and Gomez *et al.* [107], in which the authors change the positions of the *AND* and *XOR* gates in (79) and (80).

APPENDIX D

BPSO APPLICATIONS

We present the applications of the versions of the BPSO found in our review. In bold are those which were first introduced by the respective work. It is necessary to mention that many authors use the same acronym or name like “modified BPSO (MBPSO)” or “improved BPSO (IBPSO)” to different proposals. Many of them are specific to the application addressed.

Benchmark databases: BPSO [259]–[263], CBPSO [264], 1-PSO [265], GPSO and HPSO [266], mPSO [90], **BHTPSO** and **BHTPSO-QI** [1], **CMDPSOFS** [267], **ABQPSO** [268], **BAPSA** [269], **BHPSOWM** [270], **IWBPSO** [31], **PSOFC** [262], **BPSO-HS** [106], **BPSOC** [4], **BPSOGSA** [30], **BPSO-MI** and **BPSO-E** [271], **CatfishBPSO** [35], **BCPSO** and **CBPSO** [272], **EPSO** [273], **EPSOq** [274], **BPSOWFSS** [275],

FPSBPSO [276], **HPSOGO** [277], **nBPSO** [278], **ICPSO** [279], **MBPSO** [247], **modifiedBPSO** [247], **M2BPSO** [280], **PSOFC** [281], **PPBO** [50], **PSO-PRSE** [282], **TVT-BPSO** [283], **VPSO** [96], **Memetic BPSO** [267], **DSBPSO** [81], **New BPSO** [284], **Probability BPSO** [285], **UBPSO** [286], **HBP-SOSCA** [191], **NBPSO-MSA** [287], **NBPSO and NMBPSO** [87], **MHBPSO** [288], **iBPSO** [289], **BPSO** [250], **CBPSO-MIWS** [290], **IBPSO** [291], **BPSO** [292], **RBPSO** [293], **INBPSO** [294], **SSBPSO** [295]

Biology and Medicine: BPSO [85], [263], [296]–[300], **ABPSO** [301], **BVEPSO** [92], **CBPSO** [302], **DE-BPSO** [203], **MOBPSO** [222], **QBPSO** [303], **BORG-Swarm** [95], **DBPSO** [304], **DEBPSO** [83], **EPSO** [305], **HDBPSO** [306], **HBPSO** [210], **IBPSO** [307], **mIBPSO** [32], **MSABPSO** [308], **NBPSO-HAP** [257], **TCBPSO** [19], **BPSO** [309], **IBPSO** [310], **MMBPSO** [311], **DPSO** [312], **PBPSO** [313], **BPSO** [314], **PHSO** [188], **iBPSO** [315], **BPSO** [316], **BPSO** [317]

Computer Sciences: BPSO [9], [318]–[331], **CBPSO** [332], **CPSO** [333], **DBPSO** [334], **EIBPSO** [335], **IBPSO** [336], **MBPSO** [337], **BMOPSO** [229], **BFIPS** and **MBFIPS** [230] **ABPSO** [338], **AdaptBPSO** [339], **BPSO-HD** [340], **BMOPSO-CDRLS** [218], **dtBPSO** [192], **AIS-BPSO** [105], **EBPSO** [249], **IBPSO** [202], **NBMOPSO-CDR** [82], **RBPSO** [341], **Time-varying BPSO** [342], **Wavelet Mutation BPSO** [343], **BPSO** [192]

Engineering: BPSO [344]–[350] **BPSO-CSP** [351], **GBPSO** [352], **MBPSO** [353], **BHTPSO-LDDA** [86], **MBPSO** [354] **mBPSO** [355], **BPSO-IP** [356], **BPSOr** [357], **mBPSO** [29], **MBPSO** [209], **BS-MOCLPSO** [223], **PBPSO** [84], **MBPS** [358], **BPSO** [358]

Electric Engineering: BPSO [49], [107], [225], [359]–[375], **rBPSO** [34], **MOBPSO** [226], [376], **BCPSO** [377], **BPSOGSA** [206], **BQPSO** [378], **CCBQPSO** [379], **BFIPS** and **MBFIPS** [228], **BPSO-TVAC** [93], [94], **BPSO-FD** [380], **vel-BPSO** [381], **CBPSO** [382], **Customized BPSO** [91], **Direct BPSO** [383], **EBPSO** [384], **Hybrid BPSO+GA** [385], **BPSOGA** [211], **Hybrid Improved BPSO** [103], **JFPSO** [204], **HPSO** [266], **IBPSO** [386], **IBPSO** [387], **IBPSO-DSC** [227], **IBPSO** [388], **IBPSO** [104], **IBPSO** [389], **IBPSO** [390], **IBPSO** [391], **intelligent BPSO** [108], **MB-EPPO** [392], **MBPSO** [393], **MBPSO** [394], **MBPSO** [395], **modified discrete BPSO** [396], **NBPSO** [397], **PSOFT** [40], **BQPSO** [398], **QBPSO** [399], **SHBPSO-TVAC** [89], **T2FABPSOM** [400], **V-shaped BPSO** [38], **BPSO** [360], **BPSO** [401], **BPSO** [402], **QBPSO** [403], **IBPSO** [404], **ABPSO** [405], **IBPSO** [234], **IBPSO** [406]

Telecommunications: BPSO [407]–[409] [410], [411], **mIBPSO** [412], **MBPSO** [413]–[415], **MOBPSO** [224],

- BPSO-LB [416], BQPSO [33], Two-step BPSO [417], HCPSO [418], AMPBPSO [419], **BILPSO** [420], **mrBPSO** [414], **MOHPSO** [235], **MBPSO** [88], **IBPSO** [421], IBPSO [422], DBPSO [423], BPSO [424], BPSO [425], BPSO [426], **HTBPSO** [190], BPSO [427]
- Other:** BPSO [47], [48], [428]–[434], **GK-MBPSO** [435], **ImmuneBPSO** [205], **SCO-FS-BPSO** [436], **HBPSO** [193], **Bounded BPSO** [437], **Jumping PSO** [231], BPSO [438], **FBPSO** [439], **ABPSO** [248], BPSO [440], HBPSO [194], HBPSO [441], BBPSO [437].
- ## REFERENCES
- [1] A. R. Jordehi and J. Jasni, "Particle swarm optimisation for discrete optimisation problems: A review," *Artif. Intell. Rev.*, vol. 43, no. 2, pp. 243–258, 2015.
 - [2] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: Artificial bee colony (ABC) algorithm and applications," *Artif. Intell. Rev.*, vol. 42, no. 1, pp. 21–57, 2014.
 - [3] C. Ozturk, E. Hancer, and D. Karaboga, "A novel binary artificial bee colony algorithm based on genetic operators," *Inf. Sci.*, vol. 297, pp. 154–170, Mar. 2015.
 - [4] V. Singh, D. Singh, R. Tiwari, and A. Shukla, "Discrete optimization problem solving with three variants of hybrid binary particle swarm optimization," in *Proc. 3rd Workshop Biologically Inspired Algorithms Distrib. Syst.*, 2011, pp. 43–48.
 - [5] E. Hancer, B. Xue, D. Karaboga, and M. Zhang, "A binary ABC algorithm based on advanced similarity scheme for feature selection," *Appl. Soft Comput.*, vol. 36, pp. 334–348, Nov. 2015.
 - [6] K. Chandrasekaran, S. P. Simon, and N. P. Padhy, "Binary real coded firefly algorithm for solving unit commitment problem," *Inf. Sci.*, vol. 249, pp. 67–84, Nov. 2013.
 - [7] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, Jan. 2016.
 - [8] U. K. Chakraborty, *Advances Differential Evolution*, 1st ed. Heidelberg, Germany: Springer-Verlag, Sep. 2008.
 - [9] A. Pal and J. Maiti, "Development of a hybrid methodology for dimensionality reduction in Mahalanobis–Taguchi system using Mahalanobis distance and binary particle swarm optimization," *Expert Syst. Appl.*, vol. 37, no. 2, pp. 1286–1293, Mar. 2010.
 - [10] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. Comput. Simulation*, vol. 5, Oct. 1997, pp. 4104–4108.
 - [11] E. Pashaei and N. Aydin, "Binary black hole algorithm for feature selection and classification on biological data," *Appl. Soft Comput.*, vol. 56, pp. 94–106, Jul. 2017.
 - [12] M. H. Kashan, N. Nahavandi, and A. H. Kashan, "DisABC: A new artificial bee colony algorithm for binary optimization," *Appl. Soft Comput.*, vol. 12, pp. 342–352, Jan. 2012.
 - [13] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro Mach. Hum. Sci.*, 1995, pp. 39–43.
 - [14] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE ICNN*, vol. 4, Nov./Dec., pp. 1942–1948.
 - [15] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," *Appl. Soft Comput.*, vol. 8, pp. 687–697, Jan. 2008.
 - [16] C. J. A. B. Filho, F. B. de Lima Neto, A. J. C. C. Lins, A. I. S. Nascimento, and M. P. Lima, "Fish school search," in *Nature-Inspired Algorithms for Optimisation* (Studies in Computational Intelligence), vol. 193, R. Chiong, Ed. Berlin, Germany: Springer, 2009, doi: [10.1007/978-3-642-00267-0_9](https://doi.org/10.1007/978-3-642-00267-0_9).
 - [17] M. Dorigo, "Optimization, learning and natural algorithms," Ph.D. dissertation, Dept. Syst. Inf. Eng., Politecnico Milano, Milan, Italy, 1992.
 - [18] P. Shunmugapriya and S. Kanmani, "A hybrid algorithm using ant and bee colony optimization for feature selection and classification (AC-ABC Hybrid)," *Swarm Evol. Comput.*, vol. 36, pp. 27–36, Oct. 2017.
 - [19] L.-Y. Chuang, C.-S. Yang, K.-C. Wu, and C.-H. Yang, "Gene selection and classification using Taguchi chaotic binary particle swarm optimization," *Expert Syst. Appl.*, vol. 38, no. 10, pp. 13367–13377, 2011.
 - [20] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary ant lion approaches for feature selection," *Neurocomputing*, vol. 213, pp. 54–65, Nov. 2016.
 - [21] G. Pampara and A. P. Engelbrecht, "Binary artificial bee colony optimization," in *Proc. IEEE Symp. Swarm Intell.*, Apr. 2011, pp. 1–8.
 - [22] *MS Windows NT Kernel Description*, IEEE Xplore, Piscataway, NJ, USA, 2017.
 - [23] *ACM Digital Library*, ACM DL, Chicago, IL, USA, 2017.
 - [24] Springer. (2021). *Technology, Medicine*. [Online]. Available: <https://link.springer.com/>
 - [25] *Science Direct*, Amsterdam, The Netherlands: Elsevier, 2017.
 - [26] *Google Scholar*, Google, Menlo Park, CA, USA, 2017.
 - [27] H. B. Khormouji, H. Hajipour, and H. Rostami, "BODMA: A novel Metaheuristic algorithm for binary optimization problems based on open source development model algorithm," in *Proc. 7th Int. Symp. Telecommun. (IST)*, Sep. 2014, pp. 49–54.
 - [28] Y. Sharafi, M. A. Khanesar, and M. Teshnehlab, "Discrete binary cat swarm optimization algorithm," in *Proc. 3rd IEEE Int. Conf. Comput., Control Commun. (IC4)*, Sep. 2013, pp. 1–6.
 - [29] K. Fan, W. You, and Y. Li, "An effective modified binary particle swarm optimization (mBPSO) algorithm for multi-objective resource allocation problem (MORAP)," *Appl. Math. Comput.*, vol. 221, pp. 257–267, Sep. 2013.
 - [30] S. Mirjalili, G.-G. Wang, and L. D. S. Coelho, "Binary optimization using hybrid particle swarm optimization and gravitational search algorithm," *Neural Comput. Appl.*, vol. 25, no. 6, pp. 1423–1435, 2014.
 - [31] J. Liu, Y. Mei, and X. Li, "An analysis of the inertia weight parameter for binary particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 666–681, Oct. 2016.
 - [32] M. S. Mohamad, S. Omatu, S. Deris, and M. Yoshioka, "A modified binary particle swarm optimization for selecting the small subset of informative genes from gene expression data," *IEEE Trans. Inf. Technol. Biomed.*, vol. 15, no. 6, pp. 813–822, Nov. 2011.
 - [33] J. Zhang, Z. Zhou, W. Gao, Y. Ma, and Y. Ye, "Cognitive radio adaptation decision engine based on binary quantum-behaved particle swarm optimization," in *Proc. 6th Int. ICST Conf. Commun. Netw. China (CHINACOM)*, Aug. 2011, pp. 221–225.
 - [34] T. O. Ting, M. V. C. Rao, and C. K. Loo, "A novel approach for unit commitment problem via an effective hybrid particle swarm optimization," *IEEE Trans. Power Syst.*, vol. 21, no. 1, pp. 411–418, Feb. 2006.
 - [35] L. Y. Chuang, S. W. Tsai, and C. H. Yang, "Improved binary particle swarm optimization using catfish effect for feature selection," *Expert Syst. Appl.*, vol. 38, no. 10, pp. 12699–12707, 2011.
 - [36] P. Lu, J. Zhou, C. Wang, Q. Qiao, and M. Li, "Short-term hydro generation scheduling of Xiluodu and Xiangjiaba cascade hydropower stations using improved binary-real coded bee colony optimization algorithm," *Energy Convers. Manage.*, vol. 91, pp. 19–31, Feb. 2015.
 - [37] H. Nezamabadi-pour, "A quantum-inspired gravitational search algorithm for binary encoded optimization problems," *Eng. Appl. Artif. Intell.*, vol. 40, pp. 62–75, Apr. 2015.
 - [38] R. H. Kumar and S. Ushakumari, "Optimal management of islanded microgrid using binary particle swarm optimization," in *Proc. Int. Conf. Adv. Green Energy (ICAGE)*, Dec. 2014, pp. 251–257.
 - [39] L. M. Amine and K. Nadjet, "A multi-objective binary bat algorithm," in *Proc. Int. Conf. Intell. Inf. Process., Secur. Adv. Commun.*, Nov. 2015, p. 75.
 - [40] P. Puri and M. S. Hsiao, "Fast stimuli generation for design validation of RTL circuits using binary particle swarm optimization," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, Jul. 2015, pp. 573–578.
 - [41] Z. A. El Moiz Dahi, C. Mezioud, and A. Draa, "On the efficiency of the binary flower pollination algorithm: Application on the antenna positioning problem," *Appl. Soft Comput.*, vol. 47, pp. 395–414, Oct. 2016.
 - [42] A. C. Cinar and M. S. Kiran, "Similarity and logic gate-based tree-seed algorithms for binary optimization," *Comput. Ind. Eng.*, vol. 115, pp. 631–646, Jan. 2018.
 - [43] M. M. Mafarja and S. Mirjalili, "Hybrid binary ant lion optimizer with rough set and approximate entropy reduces for feature selection," *Soft Comput.*, vol. 23, no. 15, pp. 6249–6265, Aug. 2019.
 - [44] Y. He, H. Xie, T.-L. Wong, and X. Wang, "A novel binary artificial bee colony algorithm for the set-union knapsack problem," *Future Gener. Comput. Syst.*, vol. 78, pp. 77–86, Jan. 2018.
 - [45] Q. Liu, W. Lu, and W. Xu, "Spectrum allocation optimization for cognitive radio networks using binary firefly algorithm," in *Proc. Int. Conf. Innov. Design Manuf. (ICIDM)*, Aug. 2014, pp. 257–262.

- [46] D. Rodrigues, G. F. A. Silva, J. P. Papa, A. N. Marana, and X.-S. Yang, "EEG-based person identification through binary flower pollination algorithm," *Expert Syst. Appl.*, vol. 62, pp. 81–90, Nov. 2016.
- [47] I. M. Yassin, M. N. Taib, R. Adnan, M. K. M. Salleh, and M. K. Hamzah, "Effect of swarm size parameter on binary particle swarm optimization-based NARX structure selection," in *Proc. IEEE Symp. Ind. Electron. Appl.*, Sep. 2012, pp. 219–223.
- [48] I. M. Yassin, M. N. Taib, R. Adnan, M. K. M. Salleh, and M. K. Hamzah, "MySQL database lookup table for binary particle swarm optimization-based system identification," in *Proc. IEEE Symp. Ind. Electron. Appl.*, Sep. 2012, pp. 263–269.
- [49] R. Babu and B. Bhattacharyya, "Optimal placement of phasor measurement unit using binary particle swarm optimization in connected power network," in *Proc. IEEE UP Sect. Conf. Electr. Comput. Electron. (UPCON)*, Dec. 2015, pp. 1–5.
- [50] L. Zhen, L. Wang, X. Wang, and Z. Huang, "A novel PSO-inspired probability-based binary optimization algorithm," in *Proc. Int. Symp. Inf. Sci. Eng.*, Dec. 2008, pp. 248–251.
- [51] E. Rashedi, H. Nezamabadi-Pour, and S. Saryzadi, "GSA: A gravitational search algorithm," *J. Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, 2009.
- [52] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
- [53] R. F. Carneiro and C. J. A. Bastos-Filho, "Improving the binary fish school search algorithm for feature selection," in *Proc. IEEE Latin Amer. Conf. Comput. Intell. (LA-CCI)*, Nov. 2016, pp. 1–6.
- [54] L. Wei and C. Hanning, "BABC: A binary version of artificial bee colony algorithm for discrete optimization," *Int. J. Adv. Comput. Technol.*, vol. 4, no. 14, pp. 307–314, 2012.
- [55] M. Dorigo and T. Stützle, *Ant Colony Optimization*. Cambridge, MA, USA: MIT Press, 2004.
- [56] M. Dorigo, "Ant colony optimization," *Scholarpedia*, vol. 2, no. 3, p. 1461, 2007.
- [57] Y. Qing, X. Wei-Qing, and J. Bao-Chuan, "Multi-population binary ant colony algorithm with concrete behaviors for multi-objective optimization problem," in *Proc. 2nd IEEE Int. Conf. Inf. Manage. Eng.*, Jun. 2010, pp. 274–281.
- [58] K. Wei, H. Tuo, and Z. Jing, "Improving binary ant colony optimization by adaptive pheromone and commutative solution update," in *Proc. IEEE 5th Int. Conf. Bio-Inspired Comput., Theor. Appl. (BIC-TA)*, Sep. 2010, pp. 565–569.
- [59] N. K. Sreeja and A. Sankar, "Ant colony optimization based binary search for efficient point pattern matching in images," *Eur. J. Oper. Res.*, vol. 246, no. 1, pp. 154–169, Oct. 2015.
- [60] M. Wang, Y. Wan, Z. Ye, and X. Lai, "Remote sensing image classification based on the optimal support vector machine and modified binary coded ant colony optimization algorithm," *Inf. Sci.*, vol. 402, pp. 50–68, Sep. 2017.
- [61] H.-F. Kuo, "Ant colony optimization-based freeform sources for enhancing nanolithographic imaging performance," *IEEE Trans. Nanotechnol.*, vol. 15, no. 4, pp. 599–606, Jul. 2016.
- [62] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature-Inspired Cooperative Strategies for Optimization*. Berlin, Germany: Springer, 2010, pp. 65–74.
- [63] V. Basetti and A. K. Chandel, "Optimal PMU placement for power system observability using Taguchi binary bat algorithm," *Measurement*, vol. 95, pp. 8–20, Jan. 2017.
- [64] R. M. Rizk-Allah and A. E. Hassanien, "New binary bat algorithm for solving 0–1 knapsack problem," *Complex Intell. Syst.*, vol. 4, no. 1, pp. 31–53, Mar. 2018.
- [65] H. Zhang, "A binary cooperative bat algorithm based optimal topology design of leader–follower consensus," *ISA Trans.*, vol. 96, pp. 51–59, Jan. 2020.
- [66] K. I. Mohamadeen, R. M. Sharkawy, and M. M. Salama, "Binary cat swarm optimization versus binary particle swarm optimization for transformer health index determination," in *Proc. Int. Conf. Eng. Technol. (ICET)*, Apr. 2014, pp. 1–5.
- [67] B. Kumar, M. Kalra, and P. Singh, "Discrete binary cat swarm optimization for scheduling workflow applications in cloud systems," in *Proc. 3rd Int. Conf. Comput. Intell. Commun. Technol. (CICIT)*, Feb. 2017, pp. 1–6.
- [68] S. Li, J. Ma, M. Zhuang, and Y. Chen, "Research of antenna selection based on binary cat swarm optimization," in *Proc. 10th IEEE Int. Conf. Anti-Counterfeiting, Secur., Identificat. (ASID)*, Sep. 2016, pp. 131–134.
- [69] A. Srivastava and S. Maheswarapu, "Optimal PMU placement for complete power system observability using binary cat swarm optimization," in *Proc. Int. Conf. Energy Econ. Environ. (ICEEE)*, Mar. 2015, pp. 1–6.
- [70] A. Rouhi and H. Nezamabadi-pour, "Filter-based feature selection for microarray data using improved binary gravitational search algorithm," in *Proc. 3rd Conf. Swarm Intell. Evol. Comput. (CSIEC)*, Mar. 2018, pp. 1–6.
- [71] F. Barani, M. Mirhosseini, H. Nezamabadi-pour, and M. M. Farsangi, "Unit commitment by an improved binary quantum GSA," *Appl. Soft Comput.*, vol. 60, pp. 180–189, Nov. 2017.
- [72] T. Chakraborti and A. Chatterjee, "A novel binary adaptive weight GSA based feature selection for face recognition using local gradient patterns, modified census transform, and local binary patterns," *Eng. Appl. Artif. Intell.*, vol. 33, pp. 80–90, Aug. 2014.
- [73] B. Ji, X. Yuan, X. Li, Y. Huang, and W. Li, "Application of quantum-inspired binary gravitational search algorithm for thermal unit commitment with wind power integration," *Energy Convers. Manage.*, vol. 87, pp. 589–598, Nov. 2014.
- [74] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE Int. Conf. Evol. Comput.*, Jun. 1998, pp. 69–73.
- [75] K. Jiang, H. Ni, R. Han, and X. Wang, "An improved multi-objective grey wolf optimizer for dependent task scheduling in edge computing," *Int. J. Innov. Comput.*, vol. 15, no. 6, pp. 2289–2304, 2019.
- [76] P. Hu, J.-S. Pan, and S.-C. Chu, "Improved binary grey wolf optimizer and its application for feature selection," *Knowl.-Based Syst.*, vol. 195, May 2020, Art. no. 105746.
- [77] Q. M. Alzubi, M. Anbar, Z. N. M. Alqattan, M. A. Al-Betar, and R. Abdullah, "Intrusion detection system based on a modified binary grey wolf optimisation," *Neural Comput. Appl.*, vol. 32, pp. 6125–6137, Feb. 2019.
- [78] K. Luo and Q. Zhao, "A binary grey wolf optimizer for the multidimensional knapsack problem," *Appl. Soft Comput.*, vol. 83, Oct. 2019, Art. no. 105645.
- [79] K. Srikanth, L. K. Panwar, B. Panigrahi, E. Herrera-Viedma, A. K. Sangaiah, and G.-G. Wang, "Meta-heuristic framework: Quantum inspired binary grey wolf optimizer for unit commitment problem," *Comput. Electr. Eng.*, vol. 70, pp. 243–260, Aug. 2018.
- [80] S. Gu, R. Cheng, and Y. Jin, "Feature selection for high-dimensional classification using a competitive swarm optimizer," *Soft Comput.*, vol. 22, no. 3, pp. 811–822, 2018.
- [81] H. Siqueira, E. Figueiredo, M. Macedo, C. J. Santana, P. Santos, C. J. Bastos-Filho, and A. A. Gokhale, "Double-swarm binary particle swarm optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Dec. 2018, pp. 1–8.
- [82] L. S. de Souza, R. B. Prudêncio, and F. D. A. Barros, "A comparison study of binary multi-objective particle swarm optimization approaches for test case selection," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2014, pp. 2164–2171.
- [83] G. M. Ko, A. S. Reddy, S. Kumar, R. Garg, B. A. Bailey, and A. R. Hadaegh, "Differential evolution-binary particle swarm optimization algorithm for the analysis of aryl β -diketo acids for hiv-1 integrase inhibition," in *Proc. IEEE Congr. on Evol. Comput. (CEC)*, Feb. 2012, pp. 1–7.
- [84] M. I. Menhas, L. Wang, M.-R. Fei, and C.-X. Ma, "Coordinated controller tuning of a boiler turbine unit with new binary particle swarm optimization algorithm," *Int. J. Automat. Comput.*, vol. 8, no. 2, pp. 185–192, May 2011.
- [85] T. A. Souza, M. A. Souza, C. D. A. Washington, S. C. Costa, S. E. Correia, and J. and V. Vieira, "Feature selection based on binary particle swarm optimization and neural networks for pathological voice detection," in *Proc. Latin Amer. Congr. Comput. Intell. (LA-CCI)*, 2015, pp. 1–6.
- [86] V. Minzu, M. Barbu, and C. Nichita, "A binary hybrid topology particle swarm optimization algorithm for sewer network discharge," in *Proc. 19th Int. Conf. Syst. Theory, Control Comput. (ICSTCC)*, Oct. 2015, pp. 627–634.
- [87] L. Jiang, J. Liu, D. Cui, G. Bu, D. Zhang, and R. Hu, "A binary particle swarm optimization with the hybrid S-shaped and V-shaped transfer function," in *Proc. Adv. Intell. Syst. Comput.*, vol. 1107, 2020, pp. 69–77.
- [88] Z. Qin and Y.-G. Liang, "Sensor management of LEO constellation using modified binary particle swarm optimization," *Optik*, vol. 172, pp. 879–891, Nov. 2018.

- [89] N. Phuangpornpitak and S. Tia, "Optimal photovoltaic placement by self-organizing hierarchical binary particle swarm optimization in distribution systems," *Energy Proc.*, vol. 89, pp. 69–77, Jun. 2016.
- [90] A. Unler and A. Murat, "A discrete particle swarm optimization method for feature selection in binary classification problems," *Eur. J. Oper. Res.*, vol. 206, no. 3, pp. 528–539, 2010.
- [91] S. A. Azad, A. M. Oo, and M. F. Islam, "A low complexity residential demand response strategy using binary particle swarm optimization," in *Proc. 22nd Australas. Univ. Power Eng. Conf. (AUPEC)*, Dec. 2012, pp. 1–6.
- [92] Z. Ibrahim, N. K. Khalid, K. S. Lim, S. Buyamin, and J. A. A. Mukred, "A binary vector evaluated particle swarm optimization based method for DNA sequence design problem," in *Proc. IEEE Student Conf. Res. Develop.*, Dec. 2011, pp. 160–164.
- [93] S. Pookpant and W. Ongsakul, "Optimal placement of wind turbines within wind farm using binary particle swarm optimization with time-varying acceleration coefficients," *Renew. Energy*, vol. 55, pp. 266–276, Jul. 2013.
- [94] S. Pookpant and W. Ongsakul, "Design of optimal wind farm configuration using a binary particle swarm optimization at Huasai district, Southern Thailand," *Energy Convers. Manage.*, vol. 108, pp. 160–180, Jan. 2016.
- [95] S. S. Chanthaphavong and M. Chetty, "Binary-organoid particle swarm optimisation for inferring genetic networks," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Dec. 2010, pp. 1–10.
- [96] S. Mirjalili and A. Lewis, "S-shaped versus V-shaped transfer functions for binary particle swarm optimization," *Swarm Evol. Comput.*, vol. 9, pp. 1–14, Apr. 2013.
- [97] M. S. Kiran and M. Gündüz, "XOR-based artificial bee colony algorithm for binary optimization," *Turkish J. Elect. Eng. Comput. Sci.*, vol. 21, pp. 2307–2328, Nov. 2013.
- [98] D. Jia, X. Duan, and M. K. Khan, "Binary artificial bee colony optimization using bitwise operation," *Comput. Ind. Eng.*, vol. 76, pp. 360–365, Oct. 2014.
- [99] C. J. Santana, M. Macedo, H. Siqueira, A. Gokhale, and C. J. A. Bastos-Filho, "A novel binary artificial bee colony algorithm," *Future Gener. Comput. Syst.*, vol. 98, pp. 180–196, Sep. 2019.
- [100] A. Aytimur and B. Babayigit, "Binary artificial bee colony algorithms for 0-1 advertisement problem," in *Proc. 6th Int. Conf. Electr. Electron. Eng. (ICEEE)*, Apr. 2019, pp. 91–95.
- [101] H. Siqueira, E. Figueiredo, M. Macedo, C. J. Santana, C. J. A. Bastos-Filho, and A. A. Gokhale, "Boolean binary cat swarm optimization algorithm," in *Proc. IEEE Latin Amer. Conf. Comput. Intell. (LA-CCI)*, Nov. 2018, pp. 1–8.
- [102] H. Siqueira, C. Santana, M. Macedo, E. Figueiredo, A. Gokhale, and C. Bastos-Filho, "Simplified binary cat swarm optimization," *Integr. Comput.-Aided Eng.*, vol. 1, pp. 33–50, Dec. 2020.
- [103] K. Suresh and N. Kumarappan, "Hybrid improved binary particle swarm optimization approach for generation maintenance scheduling problem," *Swarm Evol. Comput.*, vol. 9, pp. 69–89, Apr. 2013.
- [104] M. Sedighzadeh, M. Dakhem, M. Sarvi, and H. H. Kordkheili, "Optimal reconfiguration and capacitor placement for power loss reduction of distribution system using improved binary particle swarm optimization," *Int. J. Energy Environ. Eng.*, vol. 5, no. 1, p. 73, Apr. 2014.
- [105] X. Pu, Z. Yi, and Z. Fang, "Holistic and partial facial features fusion by binary particle swarm optimization," *Neural Comput. Appl.*, vol. 17, nos. 5–6, pp. 481–488, Oct. 2008.
- [106] W. Bin, P. Qinke, Z. Jing, and C. Xiao, "A binary particle swarm optimization algorithm inspired by multi-level organizational learning behavior," *Eur. J. Oper. Res.*, vol. 219, no. 2, pp. 224–233, Jun. 2012.
- [107] M. Gómez, A. López, and F. Jurado, "Optimal placement and sizing from standpoint of the investor of photovoltaics grid-connected systems using binary particle swarm optimization," *Appl. Energy*, vol. 87, no. 6, pp. 1911–1918, Jun. 2010.
- [108] Y. Pirhayati and K. Mazlumi, "An intelligent binary particle swarm optimization for unit commitment problem," in *Proc. IEEE Int. Conf. Power Energy*, Nov. 2010, pp. 248–252.
- [109] J. Wang, T. Li, and R. Ren, "A real time IDSSs based on artificial bee colony-support vector machine algorithm," in *Proc. 3rd Int. Workshop Adv. Comput. Intell. (IWACI)*, Dec. 2010, pp. 91–96.
- [110] C. Ozturk, E. Hancer, and D. Karaboga, "Dynamic clustering with improved binary artificial bee colony algorithm," *Appl. Soft Comput.*, vol. 28, pp. 69–80, Mar. 2015.
- [111] Z. B. Ozger, B. Bolat, and B. Diri, "A comparative study on binary artificial bee colony optimization methods for feature selection," in *Proc. Int. Symp. Innov. Intell. Syst. Appl. (INISTA)*, Aug. 2016, pp. 1–4.
- [112] X. Zhang and X. Zhang, "A binary artificial bee colony algorithm for constructing spanning trees in vehicular ad hoc networks," *Ad Hoc Netw.*, vol. 58, pp. 198–204, Apr. 2017.
- [113] S. Xu, Z. Ji, D. T. Pham, and F. Yu, "Bio-inspired binary bees algorithm for a two-level distribution optimisation problem," *J. Bionic Eng.*, vol. 7, no. 2, pp. 161–167, Jun. 2010.
- [114] C. M. Fernandes, A. C. Rosa, and V. Ramos, "Binary ant algorithm," in *Proc. 9th Annu. Conf. Genetic Evol. Comput.*, 2007, pp. 41–48.
- [115] J. Zhao and C. Yan, "A bottleneck assigned binary ant system for multimodal optimization," in *Proc. 48th IEEE Conf. Decis. Control (CDC) held jointly with 28th Chin. Control Conf.*, Dec. 2009, pp. 6195–6200.
- [116] S. Wang, J. Watada, and W. Pedrycz, "Recourse-based facility-location problems in hybrid uncertain environment," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 40, no. 4, pp. 1176–1187, Aug. 2010.
- [117] S. Kashef and H. Nezamabadi-pour, "A new feature selection algorithm based on binary ant colony optimization," in *Proc. 5th Conf. Inf. Knowl. Technol.*, May 2013, pp. 50–54.
- [118] E. Reséndiz, L. A. Moncayo-Martínez, and G. Solís, "Binary ant colony optimization applied to variable screening in the Mahalanobis–Taguchi system," *Expert Syst. Appl.*, vol. 40, no. 2, pp. 634–637, Feb. 2013.
- [119] C. Changdar, G. S. Mahapatra, and R. K. Pal, "An ant colony optimization approach for binary knapsack problem under fuzziness," *Appl. Math. Comput.*, vol. 223, pp. 243–253, Oct. 2013.
- [120] Y. Wan, M. Wang, Z. Ye, and X. Lai, "A feature selection method based on modified binary coded ant colony optimization algorithm," *Appl. Soft Comput.*, vol. 49, pp. 248–258, Dec. 2016.
- [121] M. Zangari, A. Pozo, R. Santana, and A. Mendiburu, "A decomposition-based binary ACO algorithm for the multiobjective UBQP," *Neurocomputing*, vol. 246, pp. 58–68, Jul. 2017.
- [122] K. Kaur and N. Bawa, "Performance evaluation of image enhancement using hybrid binary bat optimization," in *Proc. 2nd Int. Conf. Next Gener. Comput. Technol. (NGCT)*, Oct. 2016, pp. 754–758.
- [123] D. Gupta, J. Arora, U. Agrawal, A. Khanna, and V. H. C. de Albuquerque, "Optimized binary bat algorithm for classification of white blood cells," *Measurement*, vol. 143, pp. 180–190, Sep. 2019.
- [124] V. A. Mokhov, D. V. Grinchenkov, and I. A. Spiridonova, "Research of binary bat algorithm on example of the discrete optimization task," in *Proc. 2nd Int. Conf. Ind. Eng., Appl. Manuf. (ICIEAM)*, 2016, pp. 1–6.
- [125] L. Pappula and D. Ghosh, "Planar thinned antenna array synthesis using multi-objective binary cat swarm optimization," in *Proc. IEEE Int. Symp. Antennas Propag.*, Jul. 2015, pp. 2463–2464.
- [126] R. Soto, B. Crawford, A. Aste Toledo, H. D. L. Fuente-Mella, C. Castro, F. Paredes, and R. Olivares, "Solving the manufacturing cell design problem through binary cat swarm optimization with dynamic mixture ratios," *Comput. Intell. Neurosci.*, vol. 2019, pp. 1–16, Feb. 2019.
- [127] B. Almonacid, F. Aspée, R. Soto, B. Crawford, and J. Lama, "Solving the manufacturing cell design problem using the modified binary firefly algorithm and the Egyptian vulture optimisation algorithm," *IET Softw.*, vol. 11, no. 3, pp. 105–115, Jun. 2017.
- [128] S. A.-F. Sayed, E. Nabil, and A. Badr, "A binary clonal flower pollination algorithm for feature selection," *Pattern Recognit. Lett.*, vol. 77, pp. 21–27, Jul. 2016.
- [129] C. Shilaja and K. Ravi, "Optimization of emission/economic dispatch using Euclidean affine flower pollination algorithm (eFPA) and binary FPA (BFPA) in solar photo voltaic generation," *Renew. Energy*, vol. 107, pp. 550–566, Jul. 2017.
- [130] M. A. Khanesar and D. Branson, "XOR binary gravitational search algorithm," in *Proc. IEEE Int. Conf. Syst., Man Cybern. (SMC)*, Oct. 2019, pp. 3269–3274.
- [131] H. Chantar, M. Mafarja, H. Alsawalqah, A. A. Heidari, I. Aljarah, and H. Faris, "Feature selection using binary grey wolf optimizer with elite-based crossover for Arabic text classification," *Neural Comput. Appl.*, vol. 32, no. 16, pp. 12201–12220, Aug. 2020.

- [132] L. K. Panwar, S. Reddy, A. Verma, B. K. Panigrahi, and R. Kumar, "Binary grey wolf optimizer for large scale unit commitment problem," *Swarm Evol. Comput.*, vol. 38, pp. 251–266, Feb. 2018.
- [133] S. K. Reddy, L. K. Panwar, B. K. Panigrahi, R. Kumar, and A. Alsumaiti, "Binary grey wolf optimizer models for profit based unit commitment of price-taking GENCO in electricity market," *Swarm Evol. Comput.*, vol. 44, pp. 957–971, Feb. 2019.
- [134] X. Zhang, C. Wu, J. Li, X. Wang, Z. Yang, J.-M. Lee, and K.-H. Jung, "Binary artificial algae algorithm for multidimensional knapsack problems," *Appl. Soft Comput.*, vol. 43, pp. 583–595, Jun. 2016.
- [135] S. Korkmaz and M. S. Kiran, "An artificial algae algorithm with stigmergic behavior for binary optimization," *Appl. Soft Comput.*, vol. 64, pp. 627–640, Mar. 2018.
- [136] B. Niu and Y. Bi, "Binary bacterial foraging optimization for 0/1 knapsack problem," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2014, pp. 647–652.
- [137] Y.-P. Chen, Y. Li, G. Wang, Y.-F. Zheng, Q. Xu, J.-H. Fan, and X.-T. Cui, "A novel bacterial foraging optimization algorithm for feature selection," *Expert Syst. Appl.*, vol. 83, pp. 1–17, Oct. 2017.
- [138] S. Arora and P. Anand, "Binary butterfly optimization approaches for feature selection," *Expert Syst. Appl.*, vol. 116, pp. 147–160, Feb. 2019.
- [139] R. C. T. de Souza, C. A. de Macedo, L. dos Santos Coelho, J. Pierezan, and V. C. Mariani, "Binary coyote optimization algorithm for feature selection," *Pattern Recognit.*, vol. 107, Nov. 2020, Art. no. 107470.
- [140] R. C. T. De Souza, L. D. S. Coelho, C. A. De Macedo, and J. Pierezan, "A V-shaped binary crow search algorithm for feature selection," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2018, pp. 1–8.
- [141] S. Laabadi, M. Naimi, H. E. Amri, and B. Achchab, "A binary crow search algorithm for solving two-dimensional bin packing problem with fixed orientation," *Proc. Comput. Sci.*, vol. 167, pp. 809–818, Jul. 2020.
- [142] M. Dalali and H. K. Karegar, "Optimal PMU placement for full observability of the power network with maximum redundancy using modified binary cuckoo optimisation algorithm," *IET Gener., Transmiss. Distrib.*, vol. 10, no. 11, pp. 2817–2824, Aug. 2016.
- [143] J. Zhao, S. Liu, M. Zhou, X. Guo, and L. Qi, "An improved binary cuckoo search algorithm for solving unit commitment problems: Methodological description," *IEEE Access*, vol. 6, pp. 43535–43545, 2018.
- [144] J. García, F. Altimiras, A. Peña, G. Astorga, and O. Peredo, "A binary cuckoo search big data algorithm applied to large-scale crew scheduling problems," *Complexity*, vol. 2018, Jul. 2018, Art. no. 8395193.
- [145] M. Mafarja, I. Aljarah, A. A. Heidari, H. Faris, P. Fournier-Viger, X. Li, and S. Mirjalili, "Binary dragonfly optimization for feature selection using time-varying transfer functions," *Knowl.-Based Syst.*, vol. 161, pp. 185–204, Dec. 2018.
- [146] M. Hariharan, R. Sindhu, V. Vijejan, H. Yazid, T. Nadarajaw, S. Yaacob, and K. Polat, "Improved binary dragonfly optimization algorithm and wavelet packet based non-linear features for infant cry classification," *Comput. Methods Programs Biomed.*, vol. 155, pp. 39–51, Mar. 2018.
- [147] P. Sengottuvelan and N. Prasath, "BAFSA: Breeding artificial fish swarm algorithm for optimal cluster head selection in wireless sensor networks," *Wireless Pers. Commun.*, vol. 94, no. 4, pp. 1979–1991, 2017.
- [148] J. A. G. Sargo, S. M. Vieira, J. M. C. Sousa, and C. J. A. B. Filho, "Binary fish school search applied to feature selection: Application to ICU readmissions," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2014, pp. 1366–1373.
- [149] C. J. Santana, C. J. Bastos-Filho, M. MacEdo, and H. Siqueira, "SBFSS: Simplified binary fish school search," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2019, pp. 2595–2602.
- [150] L. Wang, X. L. Zheng, and S. Y. Wang, "A novel binary fruit fly optimization algorithm for solving the multidimensional knapsack problem," *Knowl.-Based Syst.*, vol. 48, no. 2, pp. 17–23, 2013.
- [151] H. Hichem, M. Elkamel, M. Rafik, M. T. Mesaouad, and C. Ouahiba, "A new binary grasshopper optimization algorithm for feature selection problem," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 33, no. 9, pp. 1319–1578, Nov. 2019.
- [152] M. Mafarja, I. Aljarah, H. Faris, A. I. Hammouri, A. M. Al-Zoubi, and S. Mirjalili, "Binary grasshopper optimisation algorithm approaches for feature selection problems," *Expert Syst. Appl.*, vol. 117, pp. 267–286, Mar. 2019.
- [153] H. Pinto, A. Pe na, M. Valenzuela, and A. Fernández, "A binary grasshopper algorithm applied to the knapsack problem," in *Proc. Comput. Sci. Conf. Cham, Switzerland: Springer*, 2018, pp. 132–143.
- [154] M. Li, X. Wang, Y. Gong, Y. Liu, and C. Jiang, "Binary glowworm swarm optimization for unit commitment," *J. Mod. Power Syst. Clean Energy*, vol. 2, no. 4, pp. 357–365, Dec. 2014.
- [155] P. Xia, Z. Ni, X. Zhu, J. Zhang, and Y. Jin, "A novel prediction method based on improved binary glowworm swarm optimization and multifractal dimension for P2P lending investment risk," *IEEE Access*, vol. 8, pp. 23232–23245, 2020.
- [156] Y.-H. Feng and G.-G. Wang, "Binary moth search algorithm for discounted 0-1 knapsack problem," *IEEE Access*, vol. 6, pp. 10708–10719, 2018.
- [157] A. K. Mandal, R. Sen, and B. Chakraborty, "Binary owl search algorithm for feature subset selection," in *Proc. IEEE 10th Int. Conf. Awareness Sci. Technol. (iCAST)*, Oct. 2019, pp. 1–6.
- [158] S. Rojas-Galeano, "Binary optimisation with an urban pigeon-inspired swarm algorithm," *Commun. Comput. Inf. Sci.*, vol. 1052, pp. 190–201, Oct. 2019.
- [159] S. Ahmed, M. Mafarja, H. Faris, and I. Aljarah, "Feature selection using salp swarm algorithm with chaos," in *Proc. 2nd Int. Conf. Intell. Syst., Metaheuristics Swarm Intell.*, 2018, pp. 65–69.
- [160] A. K. Mondal and P. Saxena, "Thinning of concentric circular antenna array using binary salp swarm algorithm," in *Proc. IEEE Conf. Inf. Commun. Technol.*, Dec. 2019, pp. 2–5.
- [161] R. M. Rizk-Allah, A. E. Hassanien, M. Elhoseny, and M. Gunasekaran, "A new binary salp swarm algorithm: Development and application for optimization tasks," *Neural Comput. Appl.*, vol. 31, no. 5, pp. 1641–1663, May 2019.
- [162] H. Faris, M. M. Mafarja, A. A. Heidari, I. Aljarah, A. M. Al-Zoubi, S. Mirjalili, and H. Fujita, "An efficient binary salp swarm algorithm with crossover scheme for feature selection problems," *Knowl.-Based Syst.*, vol. 154, pp. 43–67, Aug. 2018.
- [163] Y. Meraihi, A. Ramdane-Cherif, M. Mahseur, and D. Achelia, "A chaotic binary salp swarm algorithm for solving the graph coloring problem," *Lect. Notes Netw. Syst.*, vol. 64, pp. 106–118, Dec. 2019.
- [164] A. E. Hegazy, M. Makhoulouf, and G. S. El-Tawel, "Improved salp swarm algorithm for feature selection," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 32, no. 2, pp. 335–344, 2018.
- [165] E. Cuevas, M. Cienfuegos, D. Zaldivar, and M. Pérez-Cisneros, "A swarm optimization algorithm inspired in the behavior of the social-spider," *Expert Syst. Appl.*, vol. 40, no. 16, pp. 6374–6384, 2013.
- [166] J. J. Q. Yu and V. O. K. Li, "A social spider algorithm for global optimization," *Appl. Soft Comput.*, vol. 30, pp. 614–627, May 2015.
- [167] U. P. Shukla and S. J. Nanda, "A binary social spider optimization algorithm for unsupervised band selection in compressed hyperspectral images," *Expert Syst. Appl.*, vol. 97, pp. 336–356, May 2018.
- [168] E. Ülker, "An efficient binary social spider algorithm for feature selection problem," *Expert Syst. Appl.*, vol. 146, May 2020, Art. no. 113185.
- [169] E. Bas and E. Ulker, "A binary social spider algorithm for continuous optimization task," *Soft Comput.*, vol. 24, no. 17, pp. 12953–12979, 2020.
- [170] C. Han, G. Zhou, and Y. Zhou, "Binary symbiotic organism search algorithm for feature selection and analysis," *IEEE Access*, vol. 7, pp. 166833–166859, 2019.
- [171] H. Xu, Y. Fu, C. Fang, Q. Cao, J. Su, and S. Wei, "An improved binary whale optimization algorithm for feature selection of network intrusion detection," *Proc. IEEE 4th Int. Symp. Wireless Syst.*, Dec. 2018, pp. 10–15.
- [172] A. G. Hussien, A. E. Hassanien, E. H. Houssein, S. Bhattacharyya, and M. Amin, "S-shaped binary whale optimization algorithm for feature selection," in *Recent Trends Signal Image Processing*, S. Bhattacharyya, A. Mukherjee, H. Bhaumik, S. Das, and K. Yoshida, Eds. Singapore: Springer, 2019, pp. 79–87.
- [173] A. G. Hussien, A. E. Hassanien, E. H. Houssein, M. Amin, and A. T. Azar, "New binary whale optimization algorithm for discrete optimization problems," *Eng. Optim.*, vol. 52, no. 6, pp. 945–959, Jun. 2020.
- [174] S. K. Reddy, L. Panwar, B. K. Panigrahi, and R. Kumar, "Binary whale optimization algorithm: A new metaheuristic approach for profit-based unit commitment problems in competitive electricity markets," *Eng. Optim.*, vol. 51, pp. 369–389, May 2018.
- [175] V. Kumar and D. Kumar, "Binary whale optimization algorithm and its application to unit commitment problem," *Neural Comput. Appl.*, vol. 32, no. 7, pp. 2095–2123, Apr. 2020.
- [176] M. Ghaemi and M.-R. Feizi-Derakhshi, "Feature selection using forest optimization algorithm," *Pattern Recognit.*, vol. 60, pp. 121–129, Dec. 2016.

- [177] O. Ogwo, H. Turabieh, A. Sheta, and S. King, "Medical data classification using binary brain storm optimization algorithm," in *Proc. Int. Conf. Artif. Intell., Robot. Control*, Dec. 2019, pp. 44–52.
- [178] M. Mirhosseini and H. Nezamabadi-pour, "BICA: A binary imperialist competitive algorithm and its application in CBIR systems," *Int. J. Mach. Learn. Cybern.*, vol. 9, no. 12, pp. 2043–2057, Dec. 2018.
- [179] K. Srikanth Reddy, L. K. Panwar, R. Kumar, and B. K. Panigrahi, "Binary fireworks algorithm for profit based unit commitment (PBUC) problem," *Int. J. Electr. Power Energy Syst.*, vol. 83, pp. 270–282, Dec. 2016.
- [180] J. Xue, J. Xiao, and J. Zhu, "Binary fireworks algorithm for 0-1 knapsack problem," in *Proc. Int. Conf. Artif. Intell. Adv. Manuf. (AIAM)*, Oct. 2019, pp. 218–222.
- [181] L. Wang, R. Yang, Y. Xu, Q. Niu, P. M. Pardalos, and M. Fei, "An improved adaptive binary harmony search algorithm," *Inf. Sci.*, vol. 232, pp. 58–87, May 2013.
- [182] J. Gholami, F. Pourpanah, and X. Wang, "Feature selection based on improved binary global harmony search for data classification," *Appl. Soft Comput.*, vol. 93, Aug. 2020, Art. no. 106402.
- [183] T.-J. Lee, S.-M. Park, K.-E. Ko, and K.-B. Sim, "Feature classification of EEG signal with binary heuristic optimization algorithms," in *Proc. 13th Int. Conf. Control, Autom. Syst. (ICCAS)*, Oct. 2013, pp. 237–240.
- [184] Z.-M. Gao, J. Zhao, and S.-R. Li, "The binary equilibrium optimization algorithm with sigmoid transfer functions," in *Proc. 12th Int. Conf. Mach. Learn. Comput.*, Feb. 2020, pp. 193–197.
- [185] J. Zhao and Z. M. Gao, "Simulation research on the binary equilibrium optimization algorithm," in *Proc. ACM Int. Conf. Proc. Ser.*, 2020, pp. 140–144.
- [186] A. M. Ibrahim, M. A. Tawhid, and R. K. Ward, "A binary water wave optimization for feature selection," *Int. J. Approx. Reasoning*, vol. 120, pp. 74–91, May 2020.
- [187] N. Al-Madi, H. Faris, and S. Mirjalili, "Binary multi-verse optimization algorithm for global optimization and discrete problems," *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 12, pp. 3445–3465, Dec. 2019.
- [188] T.-F. Zhao, W.-N. Chen, A. W.-C. Liew, T. Gu, X.-K. Wu, and J. Zhang, "A binary particle swarm optimizer with priority planning and hierarchical learning for networked epidemic control," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 51, no. 8, pp. 5090–5104, Aug. 2021.
- [189] S. P. Rajamohana and K. Umamaheswari, "Hybrid approach of improved binary particle swarm optimization and shuffled frog leaping for feature selection," *Comput. Electr. Eng.*, vol. 67, pp. 497–508, Apr. 2018.
- [190] X. Jia and G. Lu, "A hybrid Taguchi binary particle swarm optimization for antenna designs," *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 8, pp. 1581–1585, Aug. 2019.
- [191] L. Kumar and K. K. Bharti, "A novel hybrid BPSO-SCA approach for feature selection," *Natural Comput.*, vol. 20, no. 1, pp. 39–61, Mar. 2021.
- [192] A. J. Malik and F. A. Khan, "A hybrid technique using binary particle swarm optimization and decision tree pruning for network intrusion detection," *Cluster Comput.*, vol. 21, no. 1, pp. 667–680, Mar. 2018.
- [193] G. Lin and J. Guan, "A hybrid binary particle swarm optimization for the obnoxious p -median problem," *Inf. Sci.*, vol. 425, pp. 1–17, Jan. 2018.
- [194] G. Lin, J. Guan, Z. Li, and H. Feng, "A hybrid binary particle swarm optimization with Tabu search for the set-union knapsack problem," *Expert Syst. Appl.*, vol. 135, pp. 201–211, Nov. 2019.
- [195] A. Naz, N. Javaid, T. N. Qureshi, M. Imran, M. Ali, and Z. A. Khan, "EDHBPPO: Enhanced differential harmony binary particle swarm optimization for demand side management in smart grid," in *Proc. 32nd Int. Conf. Adv. Inf. Netw. Appl. Workshops (WAINA)*, May 2018, pp. 218–225.
- [196] M. A. Tawhid, "Hybrid binary dragonfly enhanced particle swarm optimization algorithm for solving feature selection problems," *Math. Found. Comput.*, vol. 1, no. 2, pp. 181–200, 2018.
- [197] M. A. Tawhid and K. B. Dsouza, "Hybrid binary bat enhanced particle swarm optimization algorithm for solving feature selection problems," *Appl. Comput. Informat.*, vol. 16, nos. 1–2, pp. 117–136, Apr. 2018.
- [198] M. Sarhani, A. El Afia, and R. Faizi, "Facing the feature selection problem with a binary PSO-GSA approach," in *Recent Developments in Metaheuristics*. Cham, Switzerland: Springer, 2018, pp. 447–462.
- [199] J. Too, A. R. Abdullah, and N. Mohd Saad, "Hybrid binary particle swarm optimization differential evolution-based feature selection for EMG signals classification," *Axioms*, vol. 8, no. 3, p. 79, Jul. 2019.
- [200] Q. Al-Tashi, S. J. A. Kadir, H. M. Rais, S. Mirjalili, and H. Alhussian, "Binary optimization using hybrid grey wolf optimization for feature selection," *IEEE Access*, vol. 7, pp. 39496–39508, 2019.
- [201] Q. Al-Tashi, S. J. Abdulkadir, H. M. Rais, S. Mirjalili, H. Alhussian, M. G. Ragab, and A. Alqushaibi, "Binary multi-objective grey wolf optimizer for feature selection in classification," *IEEE Access*, vol. 8, pp. 106247–106263, 2020.
- [202] S. P. Rajamohana and K. Umamaheswari, "Hybrid optimization algorithm of improved binary particle swarm optimization (iBPSO) and cuckoo search for review spam detection," in *Proc. 9th Int. Conf. Mach. Learn. Comput.*, Feb. 2017, pp. 238–242.
- [203] R. A. Galvan, A. R. Hadaegh, and M. K. Moghaddam, "Improving pharmacological research of HIV-1 integrase inhibition using differential evolution–binary particle swarm optimization and nonlinear adaptive boosting random forest regression," in *Proc. IEEE Int. Conf. Inf. Reuse Integr.*, Aug. 2015, pp. 485–490.
- [204] F. J. Ruiz-Rodriguez, M. Gomez-Gonzalez, and F. Jurado, "Improving voltage profile in radial distribution systems using binary particle swarm optimization and probabilistic load flow," in *Proc. Int. Conf. Power Eng., Energy Electr. Drives*, May 2011, pp. 1–6.
- [205] T. Zhai and Z. He, "Instance selection for time series classification based on immune binary particle swarm optimization," *Knowl.-Based Syst.*, vol. 49, pp. 106–115, Sep. 2013.
- [206] A. Fathy, M. El-Arini, and O. El-Baksawy, "An efficient methodology for optimal reconfiguration of electric distribution network considering reliability indices via binary particle swarm gravity search algorithm," *Neural Comput. Appl.*, vol. 30, no. 9, pp. 2843–2858, 2017.
- [207] G.-Q. Zeng, K.-D. Lu, Y.-X. Dai, Z.-J. Zhang, M.-R. Chen, C.-W. Zheng, D. Wu, and W.-W. Peng, "Binary-coded extremal optimization for the design of PID controllers," *Neurocomputing*, vol. 138, pp. 180–188, Aug. 2014.
- [208] S. Lee, S. Soak, S. Oh, W. Pedrycz, and M. Jeon, "Modified binary particle swarm optimization," *Prog. Natural Sci.*, vol. 18, no. 9, pp. 1161–1166, 2008.
- [209] G.-C. Luh, C.-Y. Lin, and Y.-S. Lin, "A binary particle swarm optimization for continuum structural topology optimization," *Appl. Soft Comput.*, vol. 11, no. 2, pp. 2833–2844, 2011.
- [210] B. Wei, Q. Peng, C. Li, and X. Kang, "A hybrid of binary particle swarm optimization and estimation distribution algorithm for feature selection," in *Proc. 6th Int. Conf. Natural Comput.*, vol. 5, Aug. 2010, pp. 2510–2514.
- [211] T. Jin and H. Li, "Fault location method for distribution lines with distributed generators based on a novel hybrid BPSOGA," *IET Gener., Transmiss. Distrib.*, vol. 10, no. 10, pp. 2454–2463, Jul. 2016.
- [212] Y. Zhou, Z. Li, and C. Hu, "A differential BPSO-GA hybrid algorithm for discrete combination optimization," in *Proc. 36th Chin. Control Conf. (CCC)*, Jul. 2017, pp. 2686–2690.
- [213] E. Pashaei, E. Pashaei, and N. Aydin, "Gene selection using hybrid binary black hole algorithm and modified binary particle swarm optimization," *Genomics*, vol. 111, no. 4, pp. 669–686, Jul. 2019.
- [214] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," TIK, Culver City, CA, USA, Tech. Rep. 103, 2001.
- [215] M. Macedo, C. Bastos-Filho, and R. Menezes, "Improved multi-objective binary fish school for feature selection," in *Proc. 31st Int. Flairs Conf.*, 2018, pp. 189–192.
- [216] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 32–49, 2011.
- [217] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
- [218] L. S. D. Souza, R. B. C. Prudencio, and F. D. A. Barros, "A hybrid binary multi-objective particle swarm optimization with local search for test case selection," in *Proc. Brazilian Conf. Intell. Syst.*, Oct. 2014, pp. 414–419.
- [219] B. Xue, M. Zhang, and W. N. Browne, "Multi-objective particle swarm optimisation (PSO) for feature selection," in *Proc. 14th Int. Conf. Genetic Evol. Comput. Conf.*, 2012, pp. 81–88.
- [220] M. Macedo, C. Bastos-Filho, S. Vieira, and J. Sousa, "Multi-objective binary fish school search," in *Critical Developments and Applications of Swarm Intelligence*. Hershey, PA, USA: IGI Global, 2018, pp. 53–72.
- [221] Q. Wu, Z. Ma, J. Fan, G. Xu, and Y. Shen, "A feature selection method based on hybrid improved binary quantum particle swarm optimization," *IEEE Access*, vol. 7, pp. 80588–80601, 2019.
- [222] N. Elkhani and R. C. Muniyandi, "Membrane computing to model feature selection of microarray cancer data," in *Proc. ASE BigData SocialInfor.*, 2015, p. 13.

- [223] Y.-J. Gong, J. Zhang, and Z. Fan, "A multi-objective comprehensive learning particle swarm optimization with a binary search-based representation scheme for bed allocation problem in general hospital," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Oct. 2010, pp. 1083–1088.
- [224] N. Jin and Y. Rahmat-Samii, "Advances in particle swarm optimization for antenna designs: Real-number, binary, single-objective and multiobjective implementations," *IEEE Trans. Antennas Propag.*, vol. 55, no. 3, pp. 556–567, Mar. 2007.
- [225] A. Soroudi and M. Afrasiab, "Binary PSO-based dynamic multi-objective model for distributed generation planning under uncertainty," *IET Renew. Power Gener.*, vol. 6, no. 2, pp. 67–78, 2012.
- [226] Z. Chong, Z. Dai, S. Wang, X. Liu, Y. Jiao, and L. Kong, "The application of binary particle swarm optimization in power restoration," in *Proc. 10th Int. Conf. Natural Comput. (ICNC)*, Aug. 2014, pp. 349–353.
- [227] P. Li and D. Xu, "Optimal operation of microgrid based on improved binary particle swarm optimization algorithm with double-structure coding," in *Proc. Int. Conf. Power Syst. Technol.*, Oct. 2014, pp. 3141–3146.
- [228] R. Taormina and K. W. Chau, "Data-driven input variable selection for rainfall-runoff modeling using binary-coded particle swarm optimization and Extreme Learning Machines," *J. Hydrol.*, vol. 529, pp. 1617–1632, Oct. 2015.
- [229] Q. Wei and Y. Wang, "Binary multi-objective particle swarm optimization for channel selection in motor imagery based brain-computer interfaces," in *Proc. 4th Int. Conf. Biomed. Eng. Informat. (BMEI)*, Oct. 2011, pp. 667–670.
- [230] R. Taormina, K.-W. Chau, and B. Sivakumar, "Neural network river forecasting through baseflow separation and binary-coded swarm optimization," *J. Hydrol.*, vol. 529, pp. 1788–1797, Oct. 2015.
- [231] S. Chowdhury, W. Tong, A. Messac, and J. Zhang, "A mixed-discrete particle swarm optimization algorithm with explicit diversity-preservation," *Struct. Multidisciplinary Optim.*, vol. 47, no. 3, pp. 367–388, Mar. 2013.
- [232] M. Mahmood, S. Mathavan, and M. Rahman, "A parameter-free discrete particle swarm algorithm and its application to multi-objective pavement maintenance schemes," *Swarm Evol. Comput.*, vol. 43, pp. 69–87, Dec. 2018.
- [233] H. Zhang and Q. Hui, "Binary multiagent coordination optimization with application to formation control design," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Dec. 2013, pp. 1968–1975.
- [234] C. He and D. Li, "Multi-objective operation mode optimization of medium voltage distribution networks based on improved binary particle swarm optimization," in *Proc. 4th Int. Conf. Intell. Green Building Smart Grid (IGBSG)*, Sep. 2019, pp. 639–644.
- [235] N. Jin and Y. Rahmat-Samii, "Hybrid real-binary particle swarm optimization (HPSO) in engineering electromagnetics," *IEEE Trans. Antennas Propag.*, vol. 58, no. 12, pp. 3786–3794, Dec. 2010.
- [236] S. Akhmedova and E. Semenkin, "Co-operation of biology related algorithms for multi-objective binary optimization," in *Proc. 4th Int. Congr. Adv. Appl. Informat. (IAI-AAI)*, Dec. 2015, pp. 580–585.
- [237] D. Zouache, A. Moussaoui, and F. Ben Abdelaziz, "A cooperative swarm intelligence algorithm for multi-objective discrete optimization with application to the knapsack problem," *Eur. J. Oper. Res.*, vol. 264, no. 1, pp. 74–88, Jan. 2018.
- [238] M. Oliveira, C. J. A. Bastos-Filho, and R. Menezes, "Towards a network-based approach to analyze particle swarm optimizers," in *Proc. IEEE Symp. Swarm Intell.*, Dec. 2014, pp. 1–8.
- [239] M. Oliveira, D. Pinheiro, B. Andrade, C. Bastos-Filho, and R. Menezes, "Communication diversity in particle swarm optimizers," in *Proc. Int. Conf. Swarm Intell.* Cham, Switzerland: Springer, 2016, pp. 77–88.
- [240] M. Oliveira, D. Pinheiro, M. Macedo, C. Bastos-Filho, and R. Menezes, "Better exploration-exploitation pace, better swarm: Examining the social interactions," in *Proc. IEEE Latin Amer. Conf. Comput. Intell. (LACCI)*, Nov. 2017, pp. 1–6.
- [241] M. Oliveira, D. Pinheiro, M. Macedo, C. Bastos-Filho, and R. Menezes, "Uncovering the social interaction network in swarm intelligence algorithms," *Appl. Neww. Sci.*, vol. 5, no. 1, pp. 1–20, Dec. 2020.
- [242] N. Gurrupadi, L. Taw, M. Macedo, M. Oliveira, D. Pinheiro, C. Bastos-Filho, and R. Menezes, "Modelling the social interactions in ant colony optimization," in *Proc. Int. Conf. Intell. Data Eng. Autom. Learn.* Cham, Switzerland: Springer, 2019, pp. 216–224.
- [243] L. Taw, N. Gurrupadi, M. Macedo, M. Oliveira, D. Pinheiro, C. Bastos-Filho, and R. Menezes, "Characterizing the social interactions in the artificial bee colony algorithm," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Feb. 2019, pp. 1243–1250.
- [244] M. S. Kiran, "The continuous artificial bee colony algorithm for binary optimization," *Appl. Soft Comput.*, vol. 33, pp. 15–23, Aug. 2015.
- [245] B. Crawford, R. Soto, N. Berrios, F. Johnson, and F. Paredes, "Binary cat swarm optimization for the set covering problem," in *Proc. 10th Iberian Conf. Inf. Syst. Technol. (CISTI)*, Jun. 2015, pp. 1–4.
- [246] R. L. Haupt, S. E. Haupt, and S. E. Haupt, *Practical Genetic Algorithms*, vol. 2. New York, NY, USA: Wiley, 1998.
- [247] J. C. Bansal and K. Deep, "A modified binary particle swarm optimization for knapsack problems," *Appl. Math. Comput.*, vol. 218, no. 22, pp. 11042–11061, Jul. 2012.
- [248] G. Lin and J. Guan, "Solving maximum set k -covering problem by an adaptive binary particle swarm optimization method," *Knowl.-Based Syst.*, vol. 142, pp. 95–107, Feb. 2018.
- [249] M. R. Ganesh, R. Krishna, K. Manikantan, and S. Ramachandran, "Entropy based binary particle swarm optimization and classification for ear detection," *Eng. Appl. Artif. Intell.*, vol. 27, pp. 115–128, Jan. 2014.
- [250] Z. Beheshti, "A time-varying mirrored S-shaped transfer function for binary particle swarm optimization," *Inf. Sci.*, vol. 512, pp. 1503–1542, Feb. 2020.
- [251] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Dept. Comput. Eng. Fac., Erciyes Univ., Kayseri, Turkey, Tech. Rep. tr06, 2005.
- [252] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*. London, U.K.: Luniver Press, 2010.
- [253] W. Cui, Y. He, D. Zhao, and J. Zhu, "A novel binary chaotic cloud flower pollination algorithm for analog fault diagnosis," in *Proc. IEEE 3rd Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, Oct. 2018, pp. 1175–1179.
- [254] X.-S. Yang, "Flower pollination algorithm for global optimization," in *Proc. Int. Conf. Unconventional Comput. Natural Comput.* Berlin, Germany: Springer, 2012, pp. 240–249.
- [255] H. Zhang and Q. Hui, "New multiagent coordination optimization algorithms for mixed-binary nonlinear programming with control applications," in *Proc. IEEE Symp. Comput. Intell. Control Autom. (CICA)*, Dec. 2014, pp. 1–8.
- [256] H. Zhang and Q. Hui, "Modified hybrid multiagent swarm optimization algorithms for mixed-binary nonlinear programming," in *Proc. 46th Hawaii Int. Conf. Syst. Sci.*, Jan. 2013, pp. 1412–1421.
- [257] B. Wei and J. Zhao, "Haplotype inference using a novel binary particle swarm optimization algorithm," *Appl. Soft Comput.*, vol. 21, pp. 415–422, Aug. 2014.
- [258] S.-C. Chu, P.-W. Tsai, and J.-S. Pan, "Cat swarm optimization," in *Proc. Pacific Rim Int. Conf. Artif. Intell.* Berlin, Germany: Springer, 2006, pp. 854–858.
- [259] D. Sudholt and C. Witt, "Runtime analysis of a binary particle swarm optimizer," *Theor. Comput. Sci.*, vol. 411, no. 21, pp. 2084–2100, May 2010.
- [260] S. G. Dastider, H. Kashyap, S. Mandal, A. Ghosh, and S. Goswami, "Feature subset selection for clustering using binary particle swarm optimization," in *Proc. Int. Conf. Inf. Technol. (ICIT)*, Dec. 2015, pp. 159–164.
- [261] D. S. Severo, E. Verissimo, G. D. C. Cavalcanti, and T. I. Ren, "Hybrid feature selection and weighting method based on binary particle swarm optimization," in *Proc. IEEE 25th Int. Conf. Tools Artif. Intell.*, Nov. 2013, pp. 433–438.
- [262] B. Xue, M. Zhang, and W. N. Browne, "New fitness functions in binary particle swarm optimisation for feature selection," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Dec. 2012, pp. 1–8.
- [263] I. Babaoglu, O. Findik, and E. Ülker, "A comparison of feature selection models utilizing binary particle swarm optimization and genetic algorithm in determining coronary artery disease using support vector machine," *Expert Syst. Appl.*, vol. 37, no. 4, pp. 3177–3183, 2010.
- [264] F. Wang, Y. Yang, X. Lv, J. Xu, and L. Li, "Feature selection using feature ranking, correlation analysis and chaotic binary particle swarm optimization," in *Proc. IEEE 5th Int. Conf. Softw. Eng. Service Sci.*, Jun. 2014, pp. 305–309.
- [265] D. Sudholt and C. Witt, "Runtime analysis of binary PSO," in *Proc. 10th Annu. Conf. Genetic Evol. Comput. - GECCO*, 2008, pp. 135–142.
- [266] C. Di Chio, A. Moraglio, and R. Poli, "Geometric particle swarm optimization on binary and real spaces: From theory to practice," in *Proc. Conf. Companion Genetic Evol. Comput.*, 2007, pp. 2659–2666.

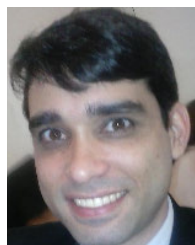
- [267] Z. Beheshti, S. M. Shamsuddin, and S. Hasan, "Memetic binary particle swarm optimization for discrete optimization problems," *Inf. Sci.*, vol. 299, pp. 58–84, Apr. 2015.
- [268] H. Lin, X. Maolong, and S. Jun, "An improved quantum-behaved particle swarm optimization with binary encoding," in *Proc. Int. Conf. Intell. Syst. Design Eng. Appl.*, vol. 1, Oct. 2010, pp. 243–249.
- [269] Z. Beheshti, S. M. Shamsuddin, and S. S. Yuhaniz, "Binary accelerated particle swarm algorithm (BAPSA) for discrete optimization problems," *J. Global Optim.*, vol. 57, no. 2, p. 549, 2013.
- [270] F. Jiang, H. Xia, Q. Anh Tran, Q. Minh Ha, N. Quang Tran, and J. Hu, "A new binary hybrid particle swarm optimization with wavelet mutation," *Knowl.-Based Syst.*, vol. 130, pp. 90–101, Aug. 2017.
- [271] L. Cervante, B. Xue, M. Zhang, and L. Shang, "Binary particle swarm optimisation for feature selection: A filter based approach," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Dec. 2012, pp. 1–8.
- [272] L.-Y. Chuang, C.-H. Yang, and J.-C. Li, "Chaotic maps based on binary particle swarm optimization for feature selection," *Appl. Soft Comput.*, vol. 11, no. 1, pp. 239–248, 2011.
- [273] E. Chen, Z. Pan, Y. Sun, and X. Liu, "A binary particle swarm optimization based on proportion probability," in *Proc. 3rd Int. Conf. Bus. Intell. Financial Eng.*, Aug. 2010, pp. 15–19.
- [274] E. Chen, J. Li, and X. Liu, "In search of the essential binary discrete particle swarm," *Appl. Soft Comput.*, vol. 11, no. 3, pp. 3260–3269, Apr. 2011.
- [275] X. Liu and L. Shang, "A fast wrapper feature subset selection method based on binary particle swarm optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2013, pp. 3347–3353.
- [276] X. Liu and L. Shang, "Fitness proportionate selection based binary particle swarm optimization," in *Proc. Annu. Conf. Genetic Evol. Comput.*, Jul. 2014, pp. 11–12.
- [277] L. F. M. López, N. G. Blas, and A. A. Albert, "Multidimensional knapsack problem optimization using a binary particle swarm model with genetic operations," *Soft Comput.*, vol. 22, pp. 2567–2582, Feb. 2017.
- [278] J. Liu and X. Fan, "The analysis and improvement of binary particle swarm optimization," in *Proc. Int. Conf. Comput. Intell. Secur.*, Dec. 2009, pp. 254–258.
- [279] S. Strasser, R. Goodman, J. Sheppard, and S. Butcher, "A new discrete particle swarm optimization algorithm," in *Proc. Genetic Evol. Comput. Conf.*, Jul. 2016, pp. 53–60.
- [280] J. Zhao, C. Han, and B. Wei, "Binary particle swarm optimization with multiple evolutionary strategies," *Sci. China Inf. Sci.*, vol. 55, no. 11, pp. 2485–2494, Nov. 2012.
- [281] B. Xue, M. Zhang, and W. N. Browne, "Single feature ranking and binary particle swarm optimisation based feature subset ranking for feature selection," in *Proc. 25th Australas. Comput. Sci. Conf.*, vol. 122, 2012, pp. 27–36.
- [282] L. Cervante, B. Xue, L. Shang, and M. Zhang, "Binary particle swarm optimisation and rough set theory for dimension reduction in classification," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Mar. 2013, pp. 2428–2435.
- [283] M. J. Islam, X. Li, and Y. Mei, "A time-varying transfer function for balancing the exploration and exploitation ability of a binary PSO," *Appl. Soft Comput.*, vol. 59, pp. 182–196, Oct. 2017.
- [284] H. Nezamabadi-pour, M. Rostami-Shahrbabaki, and M. Maghfoori-Farsangi, "Binary particle swarm optimization: Challenges and new solutions," *CSI J. Comput. Sci. Eng.*, vol. 6, no. 1, pp. 21–32, 2008.
- [285] L. Wang, X. Wang, J. Fu, and L. Zhen, "A novel probability binary particle swarm optimization algorithm and its application," *J. Softw.*, vol. 3, no. 9, pp. 28–35, Dec. 2008.
- [286] S. Mirjalili, H. Zhang, S. Mirjalili, S. Chalup, and N. Noman, "A novel u-shaped transfer function for binary particle swarm optimisation," *Adv. Intell. Syst. Comput.*, vol. 1138, pp. 241–259, Jun. 2020.
- [287] Y. Du, J. He, and C. Du, *A Novel Binary Particle Swarm Optimization for Multiple Sequence Alignment*, vol. 11644. Cham, Switzerland: Springer, 2019.
- [288] S. Gunasundari, S. Janakiraman, and S. Meenambal, "Multiswarm heterogeneous binary PSO using win-win approach for improved feature selection in liver and kidney disease diagnosis," *Comput. Med. Imag. Graph.*, vol. 70, pp. 135–154, Dec. 2018.
- [289] I. Jain, V. K. Jain, and R. Jain, "Correlation feature selection based improved-binary particle swarm optimization for gene selection and cancer classification," *Appl. Soft Comput.*, vol. 62, pp. 203–215, Jan. 2018.
- [290] J. Too, A. R. Abdullah, and N. Mohd Saad, "A new co-evolution binary particle swarm optimization with multiple inertia weight strategy for feature selection," *Informatics*, vol. 6, no. 2, p. 21, May 2019.
- [291] B. Ji, X. Lu, G. Sun, W. Zhang, J. Li, and Y. Xiao, "Bio-inspired feature selection: An improved binary particle swarm optimization approach," *IEEE Access*, vol. 8, pp. 85989–86002, 2020.
- [292] M. Mafarja, R. Jarrar, S. Ahmad, and A. A. Abusnaina, "Feature selection using binary particle swarm optimization with time varying inertia weight strategies," in *Proc. 2nd Int. Conf. Future Netw. Distrib. Syst.*, Jun. 2018, pp. 1–9.
- [293] M. Mafarja and N. R. Sabar, "Rank based binary particle swarm optimisation for feature selection in classification," in *Proc. 2nd Int. Conf. Future Netw. Distrib. Syst.*, Jun. 2018, pp. 1–6.
- [294] H. Dong, J. Sun, T. Li, and L. Li, "An improved niching binary particle swarm optimization for feature selection," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Jun. 2019, pp. 3571–3577.
- [295] B. H. Nguyen, B. Xue, P. Andraea, and M. Zhang, "A new binary particle swarm optimization approach: Momentum and dynamic balance between exploration and exploitation," *IEEE Trans. Cybern.*, vol. 51, no. 2, pp. 589–603, Feb. 2019.
- [296] S. Das and S. M. Idicula, "Biclustering gene expression data using KMeans-binary PSO hybrid," in *Proc. Int. Symp. Biocomput.*, 2010, p. 43.
- [297] Q. Gu and Y. Ding, "Binary particle swarm optimization based prediction of G-protein-coupled receptor families with feature selection," in *Proc. 1st ACM/SIGEVO Summit Genetic Evol. Comput.*, 2009, pp. 171–176.
- [298] F. Jolai, M. Taghipour, and B. Javadi, "A variable neighborhood binary particle swarm algorithm for cell layout problem," *Int. J. Adv. Manuf. Technol.*, vol. 55, nos. 1–4, pp. 327–339, Jul. 2011.
- [299] M. N. Elbedwehy, H. M. Zawbaa, N. Ghali, and A. E. Hassanien, "Detection of heart disease using binary particle swarm optimization," in *Proc. Federated Conf. Comput. Sci. Inf. Syst. (FedCSIS)*, Mar. 2012, pp. 177–182.
- [300] F. Han, C. Yang, Y.-Q. Wu, J.-S. Zhu, Q.-H. Ling, Y.-Q. Song, and D.-S. Huang, "A gene selection method for microarray data based on binary PSO encoding gene-to-class sensitivity information," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 14, no. 1, pp. 85–96, Jan. 2017.
- [301] S. Dehuri, R. Roy, and S.-B. Cho, "An adaptive binary PSO to learn Bayesian classifier for prognostic modeling of metabolic syndrome," in *Proc. 13th Annu. Conf. companion Genetic Evol. Comput.*, 2011, pp. 495–502.
- [302] L.-Y. Chuang, C.-H. Yang, J.-H. Tsai, and C.-H. Yang, "Operon prediction using chaos embedded particle swarm optimization," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 10, no. 5, pp. 1299–1309, Sep. 2013.
- [303] A. Ghosh, A. Ghosh, A. Chowdhury, and J. Hazra, "An evolutionary approach to drug-design using quantum binary particle swarm optimization algorithm," in *Proc. IEEE Students' Conf. Electr., Electron. Comput. Sci.*, Mar. 2012, pp. 1–4.
- [304] A. Merouane, S. Benziane, P. Boulet, A. El Hassan Benyamina, and L. Loukil, "Hybridization of discrete binary particle swarm optimization and invariant moments for dorsal hand vein feature selection," in *Proc. Int. Conf. Electron., Comput. Artif. Intell.*, Jun. 2013, pp. 1–6.
- [305] M. S. Mohamad, S. Omatu, S. Deris, M. Yoshioka, A. Abdullah, and Z. Ibrahim, "An enhancement of binary particle swarm optimization for gene selection in classifying cancer classes," *Algorithms Mol. Biol.*, vol. 8, no. 1, p. 15, Dec. 2013.
- [306] H. Banka and S. Dara, "A Hamming distance based binary particle swarm optimization (HDBPSO) algorithm for high dimensional feature selection, classification and validation," *Pattern Recognit. Lett.*, vol. 52, pp. 94–100, Jan. 2015.
- [307] M. S. Muhammad, K. V. Selvan, S. M. W. Masra, Z. Ibrahim, and A. F. Z. Abidin, "An improved binary particle swarm optimization algorithm for DNA encoding enhancement," in *Proc. IEEE Symp. Swarm Intell.*, Apr. 2011, pp. 1–8.
- [308] L. Djerou, N. Khelil, H. E. Dehimi, and M. Batouche, "Automatic multilevel thresholding using binary particle swarm optimization for image segmentation," in *Proc. Int. Conf. Soft Comput. Pattern Recognit.*, 2009, pp. 66–71.
- [309] L.-Y. Chuang, J.-H. Tsai, and C.-H. Yang, "Binary particle swarm optimization for operon prediction," *Nucleic Acids Res.*, vol. 38, no. 12, p. e128, Jul. 2010.

- [310] S. Agarwal and R. Reghunadhan, "Some modifications in binary particle swarm optimization for dimensionality reduction," in *Proc. World Conf. Soft Comput. Ind. Appl.* Cham, Switzerland: Springer, 2014, pp. 205–212.
- [311] H. Shaheen, S. Agarwal, and P. Ranjan, "Minmaxscaler binary pso for feature selection," *Adv. Intell. Syst. Comput.*, vol. 1045, pp. 705–716, Dec. 2020.
- [312] G. Sen and M. Krishnamoorthy, "Discrete particle swarm optimization algorithms for two variants of the static data segment location problem," *Appl. Intell.*, vol. 48, no. 3, pp. 771–790, Mar. 2018.
- [313] J. Too, A. Abdullah, N. Mohd Saad, and W. Tee, "EMG feature selection and classification using a pbest-guide binary particle swarm optimization," *Computation*, vol. 7, no. 1, p. 12, Feb. 2019.
- [314] M. Arican, K. Polat, and T. Bolu, "Binary particle swarm optimization (BPSO) based channel selection in the EEG signals and its application to speller systems," *J. Artif. Intell. Syst.*, vol. 2, no. 1, pp. 27–37, 2020.
- [315] I. Jain, V. K. Jain, and R. Jain, "An improved binary particle swarm optimization (iBPSO) for gene selection and cancer classification using DNA microarrays," in *Proc. Conf. Inf. Commun. Technol. (CICT)*, Oct. 2018, pp. 1–6.
- [316] Y. Chen, Y. Wang, L. Cao, and Q. Jin, "An effective feature selection scheme for healthcare data classification using binary particle swarm optimization," in *Proc. 9th Int. Conf. Inf. Technol. Med. Educ. (ITME)*, Oct. 2018, pp. 703–707.
- [317] M. A. Majid, A. F. Z. Abidin, N. D. K. Anuar, K. A. Kadiran, M. S. Karis, Z. M. Yusoff, N. H. K. Anuar, Z. I. Rizman, P. Gudang, U. Teknikal, and U. Teknikal, "A comparative study on the application of binary particle swarm optimization and binary gravitational search algorithm in feature selection for automatic classification of brain tumor MRI," *Fundam. Appl. Sci.*, vol. 10, no. 2S, pp. 486–498, 2018.
- [318] J.-R. Chang and H.-C. Juang, "An effective machine learning approach for refining the labels of web facial images," in *Frontier Computing*. Singapore: Springer, 2016, pp. 1073–1083.
- [319] R. Divya, A. Rath, S. Ramachandran, and K. Manikantan, "Astroid shaped DCT feature extraction for enhanced face recognition," in *Proc. Int. Inf. Technol. Conf.*, 2012, pp. 95–101.
- [320] R. Gagan, H. Sanghavi, S. Ramachandran, and K. Manikantan, "Circular sector DCT based feature extraction for enhanced face recognition using histogram based dynamic gamma intensity correction," in *Proc. Int. Inf. Technol. Conf.*, 2012, pp. 74–81.
- [321] C.-F. Lin and S.-L. Hu, "Multi-task overlapping coalition parallel formation algorithm," in *Proc. 6th Int. Joint Conf. Autom. Agents Multiagent Syst.*, 2007, p. 211.
- [322] N. N. Murthy, R. Raghunandana, K. Manikantan, and S. Ramachandran, "Face recognition using DWT thresholding based feature extraction with laplacian-gradient masking as a pre-processing technique," in *Proc. Int. Inf. Technol. Conf.*, 2012, pp. 82–89.
- [323] W. Shahzad, A. B. Siddiqui, and F. A. Khan, "Cryptanalysis of four-rounded DES using binary particleswarm optimization," in *Proc. 11th Annu. Conf. Companion Genetic Evol. Comput. Conf.*, 2009, pp. 2161–2166.
- [324] Z.-J. Zhang, J. Huang, and Y. Wei, "Frequent item sets mining from high-dimensional dataset based on a novel binary particle swarm optimization," *J. Central South Univ.*, vol. 23, no. 7, pp. 1700–1708, Jul. 2016.
- [325] A. Chatterjee, B. Tudu, and K. C. Paul, "Towards optimized binary pattern generation for grayscale digital halftoning: A binary particle swarm optimization (BPSO) approach," *J. Vis. Commun. Image Represent.*, vol. 23, no. 8, pp. 1245–1259, Nov. 2012.
- [326] R. C. Green, L. Wang, M. Alam, and C. Singh, "State space pruning for reliability evaluation using binary particle swarm optimization," in *Proc. IEEE/PES Power Syst. Conf. Expo.*, Mar. 2011, pp. 1–7.
- [327] V. F. Insanittaqwa and S. Rochimah, "Software reliability prediction based on support vector regression with binary particle swarm optimization for model mining," in *Proc. Int. Seminar Appl. Technol. Inf. Commun. (ISEmantic)*, Feb. 2016, pp. 300–305.
- [328] S. Li, J. Xu, W. Wei, and H. Qi, "Curve lane detection based on the binary particle swarm optimization," in *Proc. 29th Chin. Control Decis. Conf. (CCDC)*, May 2017, pp. 75–80.
- [329] K. Tayal and V. Ravi, "Fuzzy association rule mining using binary particle swarm optimization: Application to cyber fraud analytics," in *Proc. IEEE Int. Conf. Comput. Intell. Comput. Res. (ICCIC)*, Dec. 2015, pp. 1–5.
- [330] A. Zabidi, L. Y. Khuan, W. Mansor, I. M. Yassin, and R. Sahak, "Binary particle swarm optimization for feature selection in detection of infants with hypothyroidism," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2011, pp. 2772–2775.
- [331] A. Zabidi, W. Mansor, Y. K. Lee, I. M. Yassin, and R. Sahak, "Binary particle swarm optimization for selection of features in the recognition of infants cries with asphyxia," in *Proc. IEEE 7th Int. Colloq. Signal Process. Appl. (CSPA)*, 2011, pp. 272–276.
- [332] S. H. Babu, S. A. Birajdhara, and S. Tambad, "Face recognition using entropy based face segregation as a pre-processing technique and conservative BPSO based feature selection," in *Proc. Indian Conf. Comput. Vis. Graph. Image Process.*, Dec. 2014, p. 46.
- [333] L. Zhao and F. Qian, "Tuning the structure and parameters of a neural network using cooperative binary-real particle swarm optimization," *Expert Syst. Appl.*, vol. 38, no. 5, pp. 4972–4977, May 2011.
- [334] N. Rao P R, M. Hebbar, and M. K, "Feature selection using dynamic binary particle swarm optimization for enhanced iris recognition," in *Proc. 3rd Int. Conf. Signal Process. Integr. Netw. (SPIN)*, Feb. 2016, pp. 139–146.
- [335] H. M. Abdelsalam and A. Al-shaar, "An enhanced binary particle swarm optimization algorithm for channel assignment in cognitive radio networks," in *Proc. Int. Conf. Modeling, Identificat. Control (ICMIC)*, 2013, pp. 221–226.
- [336] H. Wang and X. Yan, "Optimizing the echo state network with a binary particle swarm optimization algorithm," *Knowl.-Based Syst.*, vol. 86, pp. 182–193, Sep. 2015.
- [337] A. Ertürk, M. K. Güllü, D. Çesmecı, D. Gerçek, and S. Ertürk, "Spatial resolution enhancement of hyperspectral images using unmixing and binary particle swarm optimization," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 12, pp. 2100–2104, Dec. 2014.
- [338] M. U. Aneesh, A. A. K. Masand, and K. Manikantan, "Optimal feature selection based on image pre-processing using accelerated binary particle swarm optimization for enhanced face recognition," *Proc. Eng.*, vol. 30, pp. 750–758, 2012.
- [339] L. Ping and S. Yu, "An adaptive binary particle swarm optimization for evolvable hardware," in *Proc. 2nd Int. Conf. Ind. Inf. Syst.*, vol. 1, Jul. 2010, pp. 98–101.
- [340] G. Zhang, R. Yang, Z. Su, F. Yue, Y. Fan, M. Qi, and J. Jiang, "Using binary particle swarm optimization to search for maximal successful coalition," *Int. J. Speech Technol.*, vol. 42, no. 2, pp. 195–209, Mar. 2015.
- [341] N. J. Sanket, K. Manikantan, and S. Ramachandran, "Recursive binary particle swarm optimization based face localization," in *Proc. 4th Nat. Conf. Comput. Vis., Pattern Recognit., Image Process. Graph. (NCVPRIPG)*, Dec. 2013, pp. 1–4.
- [342] L. Özbakár and Y. Delice, "Exploring comprehensible classification rules from trained neural networks integrated with a time-varying binary particle swarm optimizer," *Eng. Appl. Artif. Intell.*, vol. 24, no. 3, pp. 491–500, Apr. 2011.
- [343] Y. Jianliang and P. Manman, "Hardware/software partitioning algorithm based on wavelet mutation binary particle swarm optimization," in *Proc. IEEE 3rd Int. Conf. Commun. Softw. Netw.*, May 2011, pp. 347–350.
- [344] Y. Morsly, N. Aouf, and M. Djouadi, "On the optimal placement of multiple visual sensor based binary particle swarm optimization," *IFAC Proc. Vol.*, vol. 42, no. 19, pp. 279–285, 2009.
- [345] K. G. Firouzjahi, A. Sheikholeslami, and T. Barforoushi, "Multi-objective allocation of measuring system based on binary particle swarm optimization," *Frontiers Electr. Electron. Eng.*, pp. 399–415, Nov. 2012.
- [346] R. Ziani, A. Felkaoui, and R. Zegadi, "Bearing fault diagnosis using multiclass support vector machines with binary particle swarm optimization and regularized Fisher's criterion," *J. Intell. Manuf.*, vol. 28, no. 2, pp. 405–417, Feb. 2017.
- [347] X. Cui and M. Jiang, "Chaotic time series prediction based on binary particle swarm optimization," *AASRI Proc.*, vol. 1, pp. 377–383, Jan. 2012.
- [348] R. Keshavarzi, M. Akhlaghi, and F. Emami, "Binary PSO algorithm assisted to investigate the optical sensor based plasmonic nano-bi-domes," *Optik*, vol. 127, no. 19, pp. 7670–7675, Oct. 2016.
- [349] D. Alameddine, C. Marquet, D. Alameddine, and M. Khalil, "Binary particle swarm optimization for feature selection on uterine electrohystero-gram signal," in *Proc. 2nd Int. Conf. Adv. Biomed. Eng.*, Sep. 2013, pp. 125–128.
- [350] S.-S. Kim, J.-H. Byeon, H. Liu, A. Abraham, and H. Yu, "Binary particle swarm optimization for TDMA broadcast scheduling problem," in *Proc. 3rd Int. Conf. Innov. Bio-Inspired Comput. Appl.*, Sep. 2012, pp. 1–6.

- [351] M. Naeem, U. Pareek, and D. C. Lee, "Swarm intelligence for sensor selection problems," *IEEE Sensors J.*, vol. 12, no. 8, pp. 2577–2585, Aug. 2012.
- [352] J. A. Navarro-Acosta and E. O. Reséndiz-Flores, "Gompertz binary particle swarm optimization and support vector data description system for fault detection and feature selection applied in automotive pedals components," *Int. J. Adv. Manuf. Technol.*, vol. 92, no. 5, pp. 2311–2324, Sep. 2017.
- [353] G. Bagheri and A. Rasouli, "Optimization of space domes using modified binary particle swarm method (Modified BPSO)," in *Proc. Int. Conf. Intell. Adv. Syst.*, Jun. 2010, pp. 1–5.
- [354] J. J. Lake, A. E. Duwel, and R. N. Candler, "Particle swarm optimization for design of slotted MEMS resonators with low thermoelastic dissipation," *J. Microelectromech. Syst.*, vol. 23, no. 2, pp. 364–371, 2014.
- [355] G.-C. Luh and C.-Y. Lin, "A binary particle swarm optimization for structural topology optimization," in *Proc. 3rd Int. Joint Conf. Comput. Sci. Optim.*, 2010, pp. 395–399.
- [356] Y. Morsly, N. Aouf, M. S. Djouadi, and M. Richardson, "Particle swarm optimization inspired probability algorithm for optimal camera network placement," *IEEE Sensors J.*, vol. 12, no. 5, pp. 1402–1412, May 2012.
- [357] A. Martínez-Cruz, R. Barrón-Fernández, H. Molina-Lozano, M.-A. Ramírez-Salinas, L.-A. Villa-Vargas, P. Cortés-Antonio, and K.-T. T. Cheng, "An automatic functional coverage for digital systems through a binary particle swarm optimization algorithm with a reinitialization mechanism," *J. Electron. Test.*, vol. 33, no. 4, pp. 431–447, Aug. 2017.
- [358] H. P. Zhang, J. H. Ye, X. P. Yang, N. W. Muruve, and J. T. Wang, "Modified binary particle swarm optimization algorithm in lot-splitting scheduling involving multiple techniques," *Int. J. Simul. Model.*, vol. 17, no. 3, pp. 534–542, Sep. 2018.
- [359] H. Afaghzadeh, K. Valipour, M. Ebtehaj, and H. Ghadimi, "Optimal capacitor placement with consideration of voltage source harmonic distortion effect on radial distribution networks using binary particle swarm optimization (BPSO)," in *Proc. 16th Conf. Electr. Power Distrib. Netw. (EPDC)*, 2011, pp. 1–6.
- [360] M. Mohamadzeae, M. Shirali, M. Akhlaghi, and M. Kaboli, "Combination of binary particle swarm optimization algorithm and discrete dipole approximation method to investigate the plasmonic circuit-based coherent perfect absorption filter," *Opt. Quantum Electron.*, vol. 49, no. 3, p. 100, Mar. 2017.
- [361] A. Ahmadi, Y. Alinejad-Beromi, and M. Moradi, "Optimal PMU placement for power system observability using binary particle swarm optimization and considering measurement redundancy," *Expert Syst. Appl.*, vol. 38, no. 6, pp. 7263–7269, Jun. 2011.
- [362] N. Di Cesare, D. Chamoret, and M. Domaszewski, "Optimum topological design of negative permeability dielectric metamaterial using a new binary particle swarm algorithm," *Adv. Eng. Softw.*, vol. 101, pp. 149–159, Nov. 2016.
- [363] A. Ebrahimi, M. R. Ramezani, and M. Moieni, "Optimal operation of distribution network considering renewable energy sources by binary particle swarm optimization and fuzzy theory," in *Proc. Workshop, Integr. Renew. Distrib. Grid*, 2012, pp. 1–4.
- [364] A. Gutierrez, J. Perez, and J. Basterrechea, "Optimization of multilayer frequency selective surfaces using the binary particle swarm algorithm," in *Proc. 4th Eur. Conf. Antennas Propag. (EuCAP)*, 2010, pp. 1–5.
- [365] M. Rafiei and M. S. Zadeh, "Optimization of a typical biomass fueled power plant using genetic algorithm and binary particle swarm optimization," in *Proc. 17th Conf. Electr. Power Distrib. Netw. (EPDC)*, Dec. 2012, pp. 1–10.
- [366] J. N. Ramos, M. Bugnosen, E. Reyes, and M. A. Pedrasa, "Implementation and comparison of modified binary particle swarm optimization approaches to an interruptible load dispatch problem," in *Proc. Region Conf.*, Nov. 2012, pp. 1–6.
- [367] T. Remani and E. Jasmin, "Load scheduling with maximum demand using binary particle swarm optimization," in *Proc. Int. Conf. Technol. Adv. Power Energy*, Jun. 2015, pp. 294–298.
- [368] R. S. Wibowo, F. F. Utama, D. F. U. Putra, and N. K. Aryani, "Unit commitment with non-smooth generation cost function using binary particle swarm optimization," in *Proc. Int. Seminar Intell. Technol. Appl. (ISITIA)*, 2016, pp. 571–576.
- [369] H. Wu, Y. Sun, and L. Peng, "Binary particle swarm optimization algorithm for control of single-phase full-bridge inverter," in *Proc. Asia-Pacific Power Energy Eng. Conf.*, 2010, pp. 1–4.
- [370] W. H. Yan, T. S. Bi, S. F. Huang, A. C. Xue, and Q. X. Yang, "Identification of backbone-grid in power grid based on binary particle swarm optimization," in *Proc. Int. Conf. Power Syst. Technol.*, Oct. 2010, pp. 1–6.
- [371] M. Benidris, S. Elsaiah, and J. Mitra, "Power system reliability evaluation using a state space classification technique and particle swarm optimization search method," *IET Gener., Transmiss. Distrib.*, vol. 9, no. 14, pp. 1865–1873, Nov. 2015.
- [372] C. Mishra, K. D. Jones, A. Pal, and V. A. Centeno, "Binary particle swarm optimisation-based optimal substation coverage algorithm for phasor measurement unit installations in practical systems," *IET Gener., Transmiss. Distrib.*, vol. 10, no. 2, pp. 555–562, Feb. 2016.
- [373] M. A. A. Pedrasa, T. D. Spooner, and I. F. MaxGill, "Scheduling of demand side resources using binary particle swarm optimization," *IEEE Trans. Power Syst.*, vol. 24, no. 3, pp. 1173–1181, Aug. 2009.
- [374] D. Saxena, S. Bhaumik, and S. N. Singh, "Identification of multiple harmonic sources in power system using optimally placed voltage measurement devices," *IEEE Trans. Ind. Electron.*, vol. 61, no. 5, pp. 2483–2492, May 2014.
- [375] S. A. Yin and C. N. Lu, "Distribution feeder scheduling considering variable load profile and outage costs," *IEEE Trans. Power Syst.*, vol. 24, no. 2, pp. 652–660, May 2009.
- [376] S.-Y. Su, C.-N. Lu, R.-F. Chang, and G. Gutierrez-Alcaraz, "Distributed generation interconnection planning: A wind power case study," *IEEE Trans. Smart Grid*, vol. 2, no. 1, pp. 181–189, Mar. 2011.
- [377] S. Chakraborty, T. Ito, T. Senjyu, and A. Y. Saber, "Unit commitment strategy of thermal generators by using advanced fuzzy controlled binary particle swarm optimization algorithm," *Int. J. Electr. Power Energy Syst.*, vol. 43, no. 1, pp. 1072–1080, Dec. 2012.
- [378] Y. Li, H. Ye, and Z. Chen, "Binary particle swarm optimization algorithm with gene translocation for distribution network fault location," in *Proc. Spring Congr. Eng. Technol. (S-CET)*, 2012, pp. 1–4.
- [379] J. Zhao, M. Li, Z. Wang, and W. Xu, "A novel binary quantum-behaved particle swarm optimization algorithm," in *Proc. 12th Int. Symp. Distrib. Comput. Appl. Bus., Eng. Sci.*, Sep. 2013, pp. 119–123.
- [380] R. Falcon, M. Almeida, and A. Nayak, "A binary particle swarm optimization approach to fault diagnosis in parallel and distributed systems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, 2010, pp. 1–8.
- [381] A. H. El-Maleh, A. T. Sheikh, and S. M. Sait, "Binary particle swarm optimization (BPSO) based state assignment for area minimization of sequential circuits," *Appl. Soft Comput.*, vol. 13, no. 12, pp. 4832–4840, Dec. 2013.
- [382] P. Li, D. Xu, Z. Zhou, W.-J. Lee, and B. Zhao, "Stochastic optimal operation of microgrid based on chaotic binary particle swarm optimization," *IEEE Trans. Smart Grid*, vol. 7, no. 1, pp. 66–73, Jan. 2016.
- [383] M. Benidris, S. Elsaiah, and J. Mitra, "Composite system reliability assessment using dynamically directed particle swarm optimization," in *Proc. North Amer. Power Symp. (NAPS)*, Sep. 2013, pp. 1–6.
- [384] T. K. Maji and P. Acharjee, "Multiple solutions of optimal PMU placement using exponential binary PSO algorithm for smart grid applications," *IEEE Trans. Ind. Appl.*, vol. 53, no. 3, pp. 2550–2559, May/June 2017.
- [385] S. M. H. Hosseini, H. Shakhali, and Y. Ghalandaran, "Thermal unit commitment using hybrid binary particle swarm optimization and genetic algorithm," in *Proc. Asia-Pacific Power Energy Eng. Conf.*, Mar. 2012, pp. 1–5.
- [386] J. Agneessens, T. Vandoorn, B. Meersman, and L. Vandeveldel, "The use of binary particle swarm optimization to obtain a demand side management system," in *Proc. IET Conf. Renew. Power Gener. (RPG)*, 2011, pp. 1–6.
- [387] K. Suresh and N. Kumarappan, "Generation maintenance scheduling using improved binary particle swarm optimisation considering aging failures," *IET Gener., Transmiss. Distrib.*, vol. 7, no. 10, pp. 1072–1086, Oct. 2013.
- [388] J. Lang, L. Tang, and Z. Zhang, "An improved binary particle swarm optimization for unit commitment problem," in *Proc. Asia-Pacific Power Energy Eng. Conf.*, 2010, pp. 1–4.
- [389] X. Yuan, H. Nie, A. Su, L. Wang, and Y. Yuan, "An improved binary particle swarm optimization for unit commitment problem," *Expert Syst. Appl.*, vol. 36, no. 4, pp. 8049–8055, May 2009.

- [390] A. Li and B. Wang, "Feature subset selection based on binary particle swarm optimization and overlap information entropy," in *Proc. Int. Conf. Comput. Intell. Softw. Eng.*, Dec. 2009, pp. 1–4.
- [391] W. Prommee, N. Pongprapunt, and W. Ongsakul, "Improved reliability model and optimal protective device placement in micro grid by improved binary particle swarm optimization," in *Proc. 8th Int. Conf. Power Electron.*, May 2011, pp. 1514–1519.
- [392] V. H. Hinojosa and C. Leyton, "Short-term hydrothermal generation scheduling solved with a mixed-binary evolutionary particle swarm optimizer," *Electric Power Syst. Res.*, vol. 92, pp. 162–170, Nov. 2012.
- [393] K. Prescilla and A. Immanuel Selvakumar, "Modified binary particle swarm optimization algorithm application to real-time task assignment in heterogeneous multiprocessor," *Microprocessors Microsyst.*, vol. 37, nos. 6–7, pp. 583–589, Aug. 2013.
- [394] R. A. Hooshmand, M. Parastegari, and M. Yazdanpanah, "Simultaneous location of two partial discharge sources in power transformers based on acoustic emission using the modified binary partial swarm optimization algorithm," *IET Sci., Meas. Technol.*, vol. 7, no. 2, pp. 119–127, 2013.
- [395] K. Anusha and G. I. Shamini, "Power and resource allocation using modified binary particle swarm optimization in neural network," in *Proc. Int. Conf. Wireless Commun., Signal Process. Netw. (WiSPNET)*, Mar. 2016, pp. 387–390.
- [396] C. Ma, L. Qian, L. Wang, M. Ilyas Menhas, and M. Fei, "Determination of the PID controller parameters by modified binary particle swarm optimization algorithm," in *Proc. Chin. Control Decis. Conf.*, May 2010, pp. 2689–2694.
- [397] L. Li and C. Xuefeng, "Distribution network reconfiguration based on niche binary particle swarm optimization algorithm," *Energy Proc.*, vol. 17, pp. 178–182, Jun. 2012.
- [398] Q. Xu, X. Li, H. Ji, and X. Du, "Energy-efficient resource allocation for heterogeneous services in OFDMA downlink networks: Systematic perspective," *IEEE Trans. Veh. Technol.*, vol. 63, no. 5, pp. 2071–2082, Jun. 2014.
- [399] Y.-W. Jeong, J.-B. Park, S.-H. Jang, and K. Y. Lee, "A new quantum-inspired binary PSO: Application to unit commitment problems for power systems," *IEEE Trans. Power Syst.*, vol. 25, no. 3, pp. 1486–1495, Aug. 2010.
- [400] A. Soeprijanto and M. Abdillah, "Type 2 fuzzy adaptive binary particle swarm optimization for optimal placement and sizing of distributed generation," in *Proc. 2nd Int. Conf. Instrum., Commun., Inf. Technol., Biomed. Eng.*, Nov. 2011, pp. 233–238.
- [401] W.-C. Wu and M.-S. Tsai, "Feeder reconfiguration using binary coding particle swarm optimization," *Int. J. Control, Autom., Syst.*, vol. 6, no. 4, pp. 488–494, 2008.
- [402] T. M. Khalil, H. K. Youssef, and M. A. Aziz, "Optimal capacitor placement on radial distribution feeders in presence of nonlinear loads using binary particle swarm optimization," in *Proc. 19th Int. Conf. Electr. Distrib.*, vol. 5, 2007, pp. 1–4.
- [403] A. R. Jordehi, "Binary particle swarm optimisation with quadratic transfer function: A new binary optimisation algorithm for optimal scheduling of appliances in smart homes," *Appl. Soft Comput.*, vol. 78, pp. 465–480, May 2019.
- [404] R. A. Pegado and Y. P. M. Rodriguez, "Distribution network reconfiguration with the OpenDSS using improved binary particle swarm optimization," *IEEE Latin Amer. Trans.*, vol. 16, no. 6, pp. 1677–1683, Jun. 2018.
- [405] J. Li, H. Huang, B. Lou, Y. Peng, Q. Huang, and K. Xia, "Wind farm reactive power and voltage control strategy based on adaptive discrete binary particle swarm optimization algorithm," in *Proc. IEEE Asia Power Energy Eng. Conf. (APEEC)*, Mar. 2019, pp. 99–102.
- [406] M. Alam, S. Kundu, S. S. Thakur, and S. Banerjee, "Improved binary particle swarm optimization applied to solve multiple line outage identification problem," in *Proc. 7th Int. Conf. Signal Process. Integr. Netw. (SPIN)*, Feb. 2020, pp. 290–294.
- [407] W. Silabut, V. Kesornpatumanun, and W. Wongtrairat, "Improvement of antenna return loss using binary particle swarm optimization method," in *Proc. 8th Electr. Eng./Electron., Comput.*, May 2011, pp. 224–227.
- [408] W.-C. Weng, W.-H. Ho, and M.-C. Chang, "Optimal design of a planar antenna using binary particle swarm optimization," in *Proc. IEEE Int. Workshop Electromagn. (iWEM)*, Aug. 2014, pp. 68–69.
- [409] Z.-H. Zhan, J. Zhang, K.-J. Du, and J. Xiao, "Extended binary particle swarm optimization approach for disjoint set covers problem in wireless sensor networks," in *Proc. Conf. Technol. Appl. Artif. Intell.*, Nov. 2012, pp. 327–331.
- [410] S. Mirhadi, N. Komjani, and M. Soleimani, "Ultra wideband antenna design using discrete Green's functions in conjunction with binary particle swarm optimisation," *IET Microw., Antennas Propag.*, vol. 10, no. 2, pp. 184–192, 2016.
- [411] J. Moreno, M. Domingo, L. Valle, J. R. Lopez, R. P. Torres, and J. Basterrechea, "Design of indoor WLANs: Combination of a ray-tracing tool with the BPSO method," *IEEE Antennas Propag. Mag.*, vol. 57, no. 6, pp. 22–33, Dec. 2015.
- [412] M. A. Taha and D. I. Abu al Nadi, "Spectrum sensing for cognitive radio using binary particle swarm optimization," *Wireless Pers. Commun.*, vol. 72, no. 4, pp. 2143–2153, Oct. 2013.
- [413] I. F. M. Zain and S. Y. Shin, "Distributed localization for wireless sensor networks using binary particle swarm optimization (BPSO)," in *Proc. IEEE 79th Veh. Technol. Conf. (VTC Spring)*, May 2014, pp. 1–5.
- [414] A. Modiri and K. Kiasaleh, "Efficient design of microstrip antennas for SDR applications using modified PSO algorithm," *IEEE Trans. Magn.*, vol. 47, no. 5, pp. 1278–1281, May 2011.
- [415] J. Yang, H. Zhang, Y. Ling, C. Pan, and W. Sun, "Task allocation for wireless sensor network using modified binary particle swarm optimization," *IEEE Sensors J.*, vol. 14, no. 3, pp. 882–892, Mar. 2014.
- [416] Z. Ying, J. Kang-ming, C. Yuan-yuan, and T. Liang-rui, "A novel load balancing algorithm based on binary particle swarm optimization for heterogeneous integrated networks," in *Proc. 9th Int. Conf. Natural Comput. (ICNC)*, Jul. 2013, pp. 476–480.
- [417] Z. M. Yusof, T. Z. Hong, A. F. Z. Abidin, M. N. A. Salam, A. Adam, K. Khalil, J. A. A. Mukred, N. Shaikh-Husin, and Z. Ibrahim, "A two-step binary particle swarm optimization approach for routing in VLSI with iterative RLC delay model," in *Proc. 3rd Int. Conf. Comput. Intell., Modeling Simulation*, Sep. 2011, pp. 63–67.
- [418] H. Xu and Z. Zhou, "Cognitive radio decision engine using hybrid binary particle swarm optimization," in *Proc. 13th Int. Symp. Commun. Inf. Technol. (ISCIT)*, Sep. 2013, pp. 143–147.
- [419] L. Wang, X. Fu, J. Fang, H. Wang, and M. Fei, "Optimal node placement in industrial wireless sensor networks using adaptive mutation probability binary particle swarm optimization algorithm," in *Proc. 7th Int. Conf. Natural Comput.*, Jul. 2011, pp. 2199–2203.
- [420] D. Liu, Q. Jiang, and J. X. Chen, "Binary inheritance learning particle swarm optimisation and its application in thinned antenna array synthesis with the minimum sidelobe level," *IET Microw., Antennas Propag.*, vol. 9, no. 13, pp. 1386–1391, Oct. 2015.
- [421] J. Dong, Q. Li, and L. Deng, "Design of fragment-type antenna structure using an improved BPSO," *IEEE Trans. Antennas Propag.*, vol. 66, no. 2, pp. 564–571, Feb. 2018.
- [422] T. S. C. Amma Palanisamy and M. Murugappan, "Joint optimisation of ground, feed shapes with material distributive topology of patch in UWB antennas using improved binary particle swarm optimisation," *IET Microw., Antennas Propag.*, vol. 12, no. 12, pp. 1967–1972, Oct. 2018.
- [423] A. ur Rehman and A. Bermak, "Recursive DBPSO for computationally efficient electronic nose system," *IEEE Sensors J.*, vol. 18, no. 1, pp. 320–327, Jan. 2018.
- [424] S. Chelbi, H. Dhahri, M. Abdouli, C. Duvallat, and R. Bouaziz, "A new hybrid routing protocol for wireless sensor networks," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 28, no. 4, pp. 247–257, 2018.
- [425] A. Kaushik, M. Goswami, M. Manuja, S. Indu, and D. Gupta, "A binary PSO approach for improving the performance of wireless sensor networks," *Wireless Pers. Commun.*, vol. 113, no. 1, pp. 263–297, 2020.
- [426] Z. Sun, Y. Liu, and L. Tao, "Attack localization task allocation in wireless sensor networks based on multi-objective binary particle swarm optimization," *J. Netw. Comput. Appl.*, vol. 112, pp. 29–40, Jun. 2018.
- [427] Z. Ma and G. A. E. Vandenbosch, "Impact of random number generators on the performance of particle swarm optimization in antenna design," in *Proc. 6th Eur. Conf. Antennas Propag. (EUCAP)*, Mar. 2012, pp. 925–929.
- [428] L. S. Yu, H. J. Ku, C. L. Ting, R. I. Chang, Y. C. Wang, and J. M. Ho, "Content-based feature matching for fragment reassembly of ceramic reconstruction," in *Proc. IEEE/WIC/ACM Int. Joint Conferences Web Intell. (WI) Intell. Agent Technol. (IAT)*, Aug. 2014, pp. 344–351.

- [429] Y. Guo, X. Kong, and X. You, "Secure steganography based on binary particle swarm optimization," *J. Electron. (China)*, vol. 26, no. 2, pp. 285–288, Feb. 2009.
- [430] M. Sheikhan, M. Pezhmanpour, and M. S. Moin, "Improved contourlet-based steganalysis using binary particle swarm optimization and radial basis neural networks," *Neural Comput. Appl.*, vol. 21, no. 7, pp. 1717–1728, Oct. 2012.
- [431] M. Kesharaju and R. Nagarajah, "Particle swarm optimization approach to defect detection in armour ceramics," *Ultrasonics*, vol. 75, pp. 124–131, Mar. 2017.
- [432] I. M. Yassin, M. N. Taib, A. Zabidi, H. A. Hassan, and H. Z. Abidin, "Comparison between narx parameter estimation methods with binary particle swarm optimization-based structure selection method," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Dec. 2010, pp. 1–7.
- [433] T. Gong and A. L. Tuson, "Binary particle swarm optimization: A forma analysis approach," in *Proc. 9th Annu. Conf. Genetic Evol. Comput.*, 2007, p. 172.
- [434] K. N. V. D. Sarath and V. Ravi, "Association rule mining using binary particle swarm optimization," *Eng. Appl. Artif. Intell.*, vol. 26, no. 8, pp. 1832–1840, Sep. 2013.
- [435] F. Sameer, M. A. Bakar, A. Zaidan, and B. Zaidan, "A new algorithm of modified binary particle swarm optimization based on the gustafson-kessel for credit risk assessment," *Neural Comput. Appl.*, vol. 31, no. 2, pp. 337–346, 2017.
- [436] L. Shang, Z. Zhou, and X. Liu, "Particle swarm optimization-based feature selection in sentiment classification," *Soft Comput.*, vol. 20, no. 10, pp. 3821–3834, Oct. 2016.
- [437] A. S. Sairam and S. K. Verma, "Using bounded binary particle swarm optimization to analyze network attack graphs," in *Proc. 19th Int. Conf. Distrib. Comput. Netw.*, Jan. 2018, p. 41.
- [438] I. M. Yassin, A. Zabidi, M. S. A. M. Ali, N. M. Tahir, H. Z. Abidin, and Z. I. Rizman, "Binary particle swarm optimization structure selection of nonlinear autoregressive moving average with exogenous inputs (narmax) model of a flexible robot arm," *Int. J. Adv. Sci., Eng. Inf. Technol.*, vol. 6, no. 5, pp. 630–637, 2016.
- [439] G. Lin and J. Guan, "A binary particle swarm optimization for the minimum weight dominating set problem," *J. Comput. Sci. Technol.*, vol. 33, no. 2, pp. 305–322, Mar. 2018.
- [440] Fanny, L. A. Wulandhari, and S. M. Isa, "Optimum nutrition intake from daily dietary recommendation for Indonesian children using binary particle swarm optimization algorithm," *Proc. Comput. Sci.*, vol. 157, pp. 16–24, Jun. 2019.
- [441] G. Lin and J. Guan, "A hybrid binary particle swarm optimization for the obnoxious p -median problem," *Inf. Sci.*, vol. 425, pp. 1–17, Jan. 2018.



HUGO SIQUEIRA received the B.Sc. degree in electrical engineering from the State University of São Paulo, in 2006, and the master's and Ph.D. degrees from the University of Campinas, in 2009 and 2013, respectively. He held the first postdoctoral position at the University of Campinas, in 2014, and the second postdoctoral position at Illinois State University, USA, and the University of Pernambuco, in 2017. He worked on electric engineering and computer science, acting on the following subjects: neural networks, evolutionary algorithms, immune algorithms, swarm intelligence, time series forecasting, pollutant impact on human health, clustering tasks, and among others. He is currently an Adjunct Professor at the Federal University of Technology–Parana. He is also an Advisor of Graduate Programs in Computer Science (PPGCC) and Production Engineering (PPGEP). He is the Co-ordinator of the Interdisciplinary Group of Computational Intelligence and the Laboratory of Computational Intelligence and Advanced Control (LICON), UTFPR-PG.



ELLIACKIN FIGUEIREDO received the B.Eng. degree in computer engineering from the University of Pernambuco, Brazil, in 2010, and the M.Sc. and Ph.D. degrees in computer science from the Federal University of Pernambuco (UFPE), Brazil, in 2013 and 2017, respectively. Since 2020, he has been a Brazilian Civil Servant and a Project Manager with the Innovation Laboratory of Information Technology (STI Labs), UFPE. In the STI Labs, he works developing and delivering software solutions that meet business needs of the university. His passion for agile methods is based on his belief that these methods enable to deliver high quality software products to customers and end-users. His research interests include artificial intelligence and data science as well as their applications in engineering and financial problems. Besides that, he has interests in software engineering and agile methods, such as XP and Scrum.



CLODOMIR SANTANA received the bachelor's degree in computer engineering from the Polytechnic School, University of Pernambuco, and the master's degree in systems engineering from the University of Pernambuco. He is currently pursuing the Ph.D. degree in computer science with the University of Exeter, U.K. During his undergraduate, he was awarded a Scholarship from the Brazilian Coordination of Superior Level Staff Improvement (CAPES) to be a Visiting Student at the Faculty of Engineering and Applied Science, Memorial University, Canada, for a period for 16 months. His master's dissertation received an honorable mention from the University of Pernambuco, in 2019. His research interests include bio-inspired metaheuristics, robots' swarm, clustering techniques, multi-objective optimization, complex networks, human dynamics, and neural networks.



MARIANA MACEDO received the B.Sc. and M.Sc. degrees in computer engineering from the University of Pernambuco, in 2016 and 2017, respectively. She is currently pursuing the Ph.D. degree with the University of Exeter, U.K. She studied one extra year at Pace University as a Non-Degree International Student sponsored by the Brazil Scientific Mobility Program Fully-Funded Scholarship Recipient, in 2015. Her research interests include swarm intelligence, evolutionary computation, multi and many-objective, binary optimization, complex networks, human mobility, data mining, and machine learning.



RODRIGO C. LIRA was born in Recife, Brazil. He received the bachelor's and master's degrees in computer engineering from the Polytechnic School, University of Pernambuco, in 2013 and 2014, respectively, where he is currently pursuing the Ph.D. degree in computer engineering. Since 2017, he has been a Professor with the Federal Institute of Pernambuco (IFPE) acting on projects using computational intelligence and the Internet of Things.



ANU GOKHALE (Senior Member, IEEE) received the master's degree in physics-electronics from the College of William & Mary, and the Ph.D. degree from Iowa State University. She has completed 30 years as a Faculty and has received several college and university research, teaching and service awards. She was named Fulbright Distinguished Chair in STEM+C at the University of Pernambuco, Brazil, during 2016–2017; was a Faculty Fellow in Israel and a Fulbright Specialist in Cybersecurity at Gujarat Technological University, India, in summer 2017; and a Visiting Professor with the College of Business, Shandong University, Jinan, China, in spring 2017, where she focused on data analytics and e-commerce. She is currently a Distinguished Professor and the Co-ordinator of the Computer Systems Technology Program, Illinois State University (ISU). Her achievements encompass extensively cited refereed publications; groundbreaking externally funded research supported by a continuous stream for 20 years of grants from state and federal agencies, including the National Science Foundation; and elevation of the ISU student experience through excellence in teaching, mentorship, and the creation of opportunities for students to get involved in research. The current NSF funded project is in Computing Education for the 21st Century.



CARMELO BASTOS-FILHO (Senior Member, IEEE) was born in Recife, Brazil, in 1978. He received the B.Sc. degree in electronics engineering and the M.Sc. and Ph.D. degrees in electrical engineering from the Federal University of Pernambuco (UFPE), in 2000, 2003, and 2005, respectively. From 2016 to 2020, he was the Scientist-in-Chief of the Technological Park for Electronics and Industry 4.0 of Pernambuco. He is currently an Associate Professor with the Polytechnic School, University of Pernambuco. He is the Director for Innovation Parks and Graduate Studies of Pernambuco. He is also the Co-ordinator of the National Initiative for Promoting ICT, Pernambuco jointly with the Ministry for Regional Development. He is a Research Fellow Level 1D of the National Research Council of Brazil (CNPq). He published roughly 300 full papers in journals and conferences and advised over 50 Ph.D. and M.Sc. candidates. His research interests include the development of protocols and algorithms to manage and to design optical communication networks, development of novel swarm intelligence algorithms and the deployment of swarm intelligence for complex optimization and clustering problems, development and application of multiobjective optimization and many-objective optimization for real-world problems, development and application of soft deep learning techniques, development of solutions using network sciences for big data, applications in robotics, and applications in biomedical problems. In 2006, he received the Best Brazilian Thesis Award in electrical engineering. More information can be found at: <http://scholar.google.com/citations?user=t3A96agAAAAJ&hl=en>.

...