

Review

Swarm Intelligence Algorithms for Feature Selection: A Review

Lucija Brezočnik ^{*}, Iztok Fister, Jr.  and Vili Podgorelec 

Institute of Informatics, Faculty of Electrical Engineering and Computer Science, University of Maribor, Koroška cesta 46, 2000 Maribor, Slovenia; iztok.fister1@um.si (I.F.J.); vili.podgorelec@um.si (V.P.)

* Correspondence: lucija.brezocnik@um.si; Tel.: +386-2-220-7385

Received: 25 July 2018; Accepted: 23 August 2018; Published: 1 September 2018



Featured Application: The paper analyzes the usage and mechanisms of feature selection methods that are based on swarm intelligence in different application areas.

Abstract: The increasingly rapid creation, sharing and exchange of information nowadays put researchers and data scientists ahead of a challenging task of data analysis and extracting relevant information out of data. To be able to learn from data, the dimensionality of the data should be reduced first. Feature selection (FS) can help to reduce the amount of data, but it is a very complex and computationally demanding task, especially in the case of high-dimensional datasets. Swarm intelligence (SI) has been proved as a technique which can solve NP-hard (Non-deterministic Polynomial time) computational problems. It is gaining popularity in solving different optimization problems and has been used successfully for FS in some applications. With the lack of comprehensive surveys in this field, it was our objective to fill the gap in coverage of SI algorithms for FS. We performed a comprehensive literature review of SI algorithms and provide a detailed overview of 64 different SI algorithms for FS, organized into eight major taxonomic categories. We propose a unified SI framework and use it to explain different approaches to FS. Different methods, techniques, and their settings are explained, which have been used for various FS aspects. The datasets used most frequently for the evaluation of SI algorithms for FS are presented, as well as the most common application areas. The guidelines on how to develop SI approaches for FS are provided to support researchers and analysts in their data mining tasks and endeavors while existing issues and open questions are being discussed. In this manner, using the proposed framework and the provided explanations, one should be able to design an SI approach to be used for a specific FS problem.

Keywords: swarm intelligence; feature selection; swarm intelligence framework; machine learning; computational intelligence; survey

1. Introduction

Nowadays, researchers in machine learning and similar domains are increasingly recognizing the importance of dimensionality reduction of analyzed data. Not only that such high-dimensional data affect learning models, increasing the search space and computational time, but they can also be considered information poor [1,2]. Furthermore, because of high-dimensional data and, consequently, a vast number of features, the construction of a suitable machine learning model can be extremely demanding and often almost fruitless. We are faced with the so-called *curse of dimensionality* that refers to a known phenomenon that arises when analyzing data in high-dimensional spaces, stating that data in high-dimensional space become sparse [2,3].

To overcome problems arising from the high-dimensionality of data, researchers use mainly two approaches. The first one is feature extraction, which comprises the creation of a new feature

space with low dimensionality. The second one is feature selection that focuses primarily on removal of irrelevant and redundant features of the original feature set and, therefore, the selection of a small subset of relevant features. The feature selection problem with n features has 2^n possible solutions (or feature subsets). With each additional feature, the complexity doubles. Thus, a dataset with 1000 features per instance has 2^{1000} possible feature subsets.

Feature Selection (FS) is tackled more and more with Swarm Intelligence (SI) algorithms [4,5], because SI has been proved as a technique which can solve NP-hard computational problems [6] and finding an optimal feature subset is certainly this kind of a problem. SI algorithms themselves have gained popularity in recent years [4], and, nowadays, we have by far surpassed its division into only the Particle Swarm Optimization [7] and Ant Colony Optimization [8], being the two best known SI approaches. In recent years, a considerable number of new swarm-inspired algorithms have emerged, as can be seen in the scope of surveys (e.g., [9,10]).

Despite the proven usage of SI algorithms for FS [11], a narrowed search for surveys in this field gives few results. Basir and Ahmad [12] conducted a comparison of swarm algorithms for feature selections/reductions. They focused only on six bio-inspired swarm algorithms, i.e., Particle Swarm Optimization (PSO), Ant Colony Optimization Algorithms (ACO), Artificial Fish Swarm Algorithms (AFSA), Artificial Bees Colony Algorithms (ABC), Firefly Algorithms (FA), and Bat Algorithms (BA).

In [13], Xue et al. provided a comprehensive survey of Evolutionary Computation (EC) techniques for FS. They divided EC paradigms into Evolutionary Algorithms (EAs), SI, and others. Among SI techniques, only PSO and ACO were analyzed in detail. Other briefly mentioned SI algorithms for FS include only the Bees algorithm and ABC.

However, even a quick literature review reveals that FS can also be addressed successfully by a number of other SI algorithms and approaches. Obviously, there is a gap in coverage of SI algorithms for FS. With this paper we try to fill this gap, providing a comprehensive overview of how SI can be used for FS, and what are the most used problems and domains where SI algorithms have proven to be especially successful.

In this paper, we try to find answers to three major research questions:

RQ1: What is the current state of this research area?

- 1.1: Is this research area interesting for wide audience?
- 1.2: How many papers that utilize SI approaches for FS tasks appeared in recent years?
- 1.3: Are there any special real-world applications combining swarm intelligence and feature selection?

RQ2: What are the main differences among SI approaches for FS?

- 2.1: Are there any special differences in architecture design, additional parameters, computational loads and operators?

RQ3: How could one researcher develop an SI approach for FS tasks?

- 3.1: What are the main steps and how can someone easily tailor a problem to existing frameworks?
- 3.2: How can also researchers in this domain describe experimental settings and conduct fair comparisons among various algorithms?

The main contributions of this paper are, thus, as follows:

- We performed a comprehensive literature review of swarm intelligence algorithms and propose an SI taxonomy based on the algorithms' biological inspiration. For all the reviewed algorithms we provide the year of its first appearance in scientific literature; the chronological perspective of SI evolution is presented in this manner.
- We provide unified SI framework, which consists of five fundamental phases. Although the algorithms use different mechanisms, these five phases are common for all of them. The SI framework is used to explain different approaches to FS.

- We provide a comprehensive overview of SI algorithms for FS. Using the proposed SI framework, different approaches, techniques, methods, their settings and implications for use are explained for various FS aspects. Furthermore, the most frequent datasets used for the evaluation of SI algorithms for FS are presented, as well as the most common application areas. Differences among EA approaches against SI approaches are also outlined.
- We systematically outline guidelines for researchers who would like to use SI approaches for FS tasks in real-world.

This paper firstly provides a SI definition, SI framework, and proposed SI taxonomy in Section 2. FS and its variants are presented in Section 3. Section 4 comprises methodology for searching the relevant literature for this survey, and in Section 5, SI is applied to the problem of FS. Based on the proposed SI framework, each stage of the latter provides examples of proposed SI algorithms for FS. Applications that combine SI and FS are summarized in Section 6, and further divided based on used datasets or application areas. Section 7 brings the discussion of SI for FS algorithms, where issues, open questions, and guidelines for SI researchers are debated in detail. Finally, Section 8 contains the conclusions and plans for further work in the field.

2. Swarm Intelligence

The term Swarm Intelligence was introduced by Beni and Wang [14] in the context of cellular robotic systems. Nowadays, SI is commonly accepted as one branch of Computational Intelligence (CI) [15].

When speaking of SI, the five principles of SI paradigm must be fulfilled. The mentioned principles were defined by Milonas [16] already in 1994, and comprise the *proximity principle*, the *quality principle*, the *principle of diverse response*, the *principle of stability*, and the *principle of adaptability*.

Steady growth of the scientific papers on the topic of SI shows that SI is one of the most promising forms of CI. Contributing to this are undoubtedly the SI advantages [17]. One of the clearest benefits is *autonomy*. Swarm does not have any outside management, but each agent in the swarm controls their behavior autonomously. Agent in this case represents a possible solution to a given problem. Based on that, we can deduce the second advantage, which is *self-organization*. The intelligence does not focus on the individual agent, but emerges in the swarm itself. Therefore, the solutions (agents) of the problem are not known in advance, but change themselves at the time of the running program. The self-organization plays an important role in *adaptability*. The latter is noticeable in changing environments where agents respond well to the mentioned changes, modify their behavior, and adapt to them autonomously. In addition, because there is no central coordination, the swarm is *robust*, as there is no single point of failure. Moreover, the swarm enables redundancy in which two other advantages hide. The first one is *scalability*, meaning that the swarm can consist of a few to up to thousands of agents and, either way, the control architecture remains the same. Moreover, because there is no single agent essential for the swarm, the SI advantage *flexibility* is fully satisfied.

Each SI algorithm must obey some fundamental phases. Thus, the SI framework (Figure 1) can be defined as follows:

1. Initialize population
2. Define stop condition
3. Evaluate fitness function
4. Update and move agents
5. Return the global best solution

Before the initialization phase, the values of the algorithm parameters should be defined. For example, in PSO, the initial population of feasible solutions is generated and the values of the following required parameters are set: cognitive and social coefficients c_1 and c_2 , inertia weight ω , and lower and upper bounds of velocity v_{min} and v_{max} , respectively. The evolutionary process then begins with initialization; some initialization strategies are presented in Section 5.2.

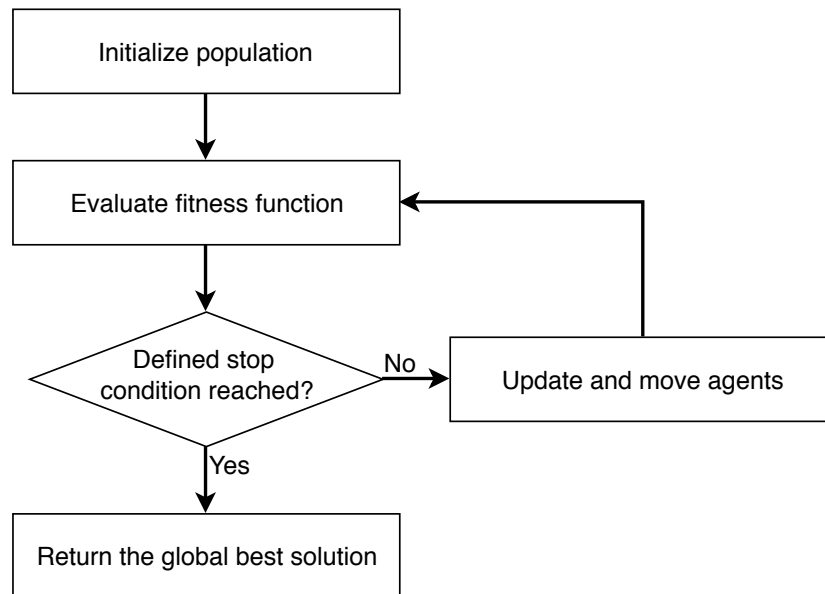


Figure 1. Swarm intelligence framework.

The primary function of the stop condition is to stop the execution of the algorithm. As shown in Section 5.3, the stopping criterion is either single or composed.

The third phase of the SI framework that evaluates the fitness function is responsible for the evaluation of the search agents. Similar to the stop condition, the fitness function is either single, i.e., some basic metric such as classification accuracy, or composed. The fitness functions used in FS are presented in Section 5.4.

Agents in an SI algorithm update and move, based on some mathematical background of the algorithm. The moves and discretized positions of agents for FS are presented in Section 5.5.

The result of an SI algorithm is the best search agent.

Based on the biological inspiration of SI algorithms, the taxonomy of SI is proposed (see Figure 2). The latter does not include other bio-inspired optimization algorithms [18], which do not have the swarm concept, e.g., monkey search [19], dolphin echolocation [20], and jaguar algorithm [21].

Lately, researchers tackle some optimization problems also with Agent Swarm Optimization (ASO). The benefits of such an approach are a multi-agent point of view, a coexistence of different agent breeds and their interaction, and agents endowment with problem-specific rules [22,23].

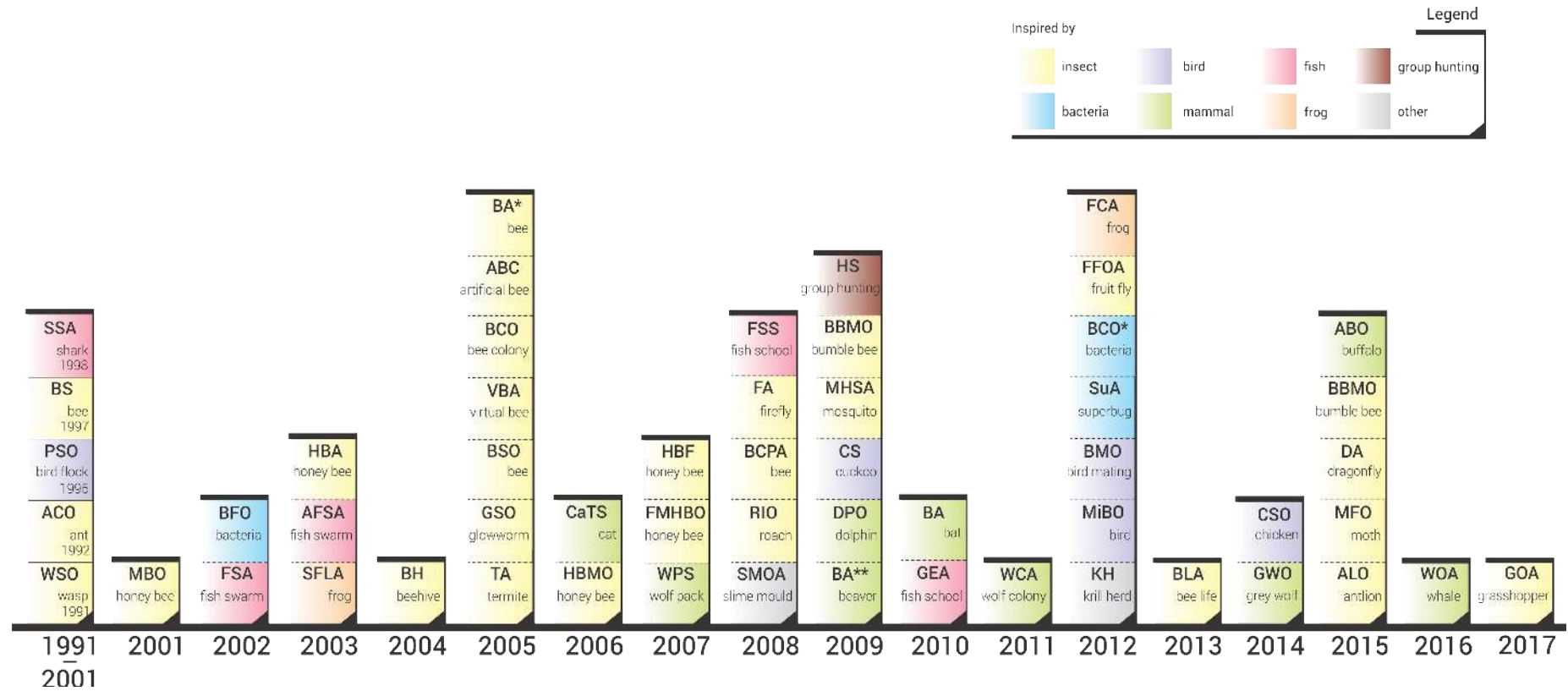


Figure 2. Swarm intelligence algorithms. * are added because the same abbreviations are used for different algorithms (see Table A1 in the Appendix A).

3. Feature Selection

In Machine Learning (ML), the belief that “the more the variables, the better the performance” has not been acceptable for some time now; thus, the application of variable (feature) selection techniques has been gaining popularity fast in the field of Data Mining [24]. Although there are different techniques and various approaches in ML (supervised classification or regression, unsupervised clustering, etc.), a dataset of instances, represented by values of a number of attributes (or features), is common to them all. The main goal of ML algorithms is then to find the best possible mapping between features’ values and the desired result (a classification or a regression model, a set of clusters, etc.) by searching for patterns in data.

When the dimensionality of data increases, the computational cost also increases, usually exponentially. To overcome this problem, it is necessary to find a way to reduce the number of features in consideration. Generally, two techniques are often used: feature (subset) selection, and feature extraction. Feature extraction creates new variables as combinations of others to reduce the dimensionality. FS, on the other hand, works by removing features that are not relevant or are redundant (Figure 3). In this manner, it searches for a projection of the data onto fewer variables (features) which preserves the information as much as possible and generally allows ML algorithms to find better mappings more efficiently.

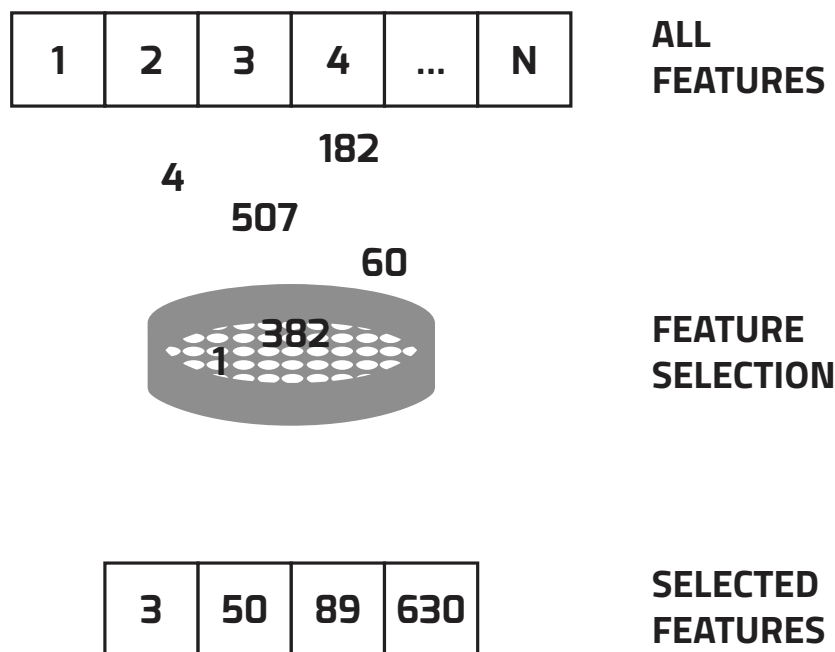


Figure 3. General Feature Selection.

FS algorithms can be separated into three categories [25]:

1. filter models;
2. wrapper models; and
3. embedded models.

A filter model evaluates features without utilizing any ML algorithms by relying on the characteristics of data [26]. First, it ranks features based on certain criteria—either independently of each other or by considering the feature space. Second, the features with highest rankings are selected to compose a feature subset. Some of the most common filter methods are Information Gain, Mutual Information, Chi2, Fisher Score, and ReliefF.

The major disadvantage of the filter approach is that it ignores the effects of the selected features on the performance of the used ML algorithm [27]. A wrapper model builds on an assumption that the

optimal feature set depends on why and how they will be used in a certain situation and by a certain ML algorithm. Thus, by taking the model hypothesis into account by training and testing in the feature space, wrapper models tend to perform better in selecting features. This, however, leads to a big disadvantage of wrapper models—the computational inefficiency.

Finally, an embedded model embeds FS within a specific ML algorithm. In this manner, embedded models have the advantages of both wrapper models, i.e., they include the interaction with the ML algorithm, and filter models, i.e., they are far less computationally intensive than wrapper methods [24]. The major disadvantage of embedded methods, however, is that an existing ML algorithm needs to be specifically redesigned, or even a new one developed, to embed the FS functionality alongside its usual ML elements. For this reason, the existing and commonly accepted ML algorithms cannot be used directly, which significantly increases the effort needed to perform the task.

4. Methodology

In this section, strategies for searching the relevant literature for this survey are described in details. Firstly, we have selected the most appropriate keywords and built a search string. Selected keywords were “swarm intelligence” and “feature selection”, while the search string was: “swarm intelligence” + “feature selection”. Search was limited on the major databases that mostly cover computer science related content: ACM Digital Library, Google Scholar, IEEE Xplore, ScienceDirect, SpringerLink, and Web of Science. Results obtained from those databases are presented in Table 1.

Table 1. Hits for “swarm intelligence” AND “feature selection” in online academic databases from 2007 to 2017.

Academic Online Database	Hits
ACM Digital Library	61
IEEE Xplore	52
ScienceDirect	409
SpringerLink	603
Web of Science	98

The next stage involved pre-screening the abstracts of returned results. The main aim of pre-screening was to remove redundant data (some papers were returned in more than one database) along with inappropriate results. Inappropriate results covered some papers, where authors claimed they used SI concept, but our study revealed that there were only bio-inspired algorithms without SI concept.

Altogether, we selected 64 papers that were studied in deep details. Findings of this study are presented in the next sections.

Papers comprising SI and FS have had almost exponential growth over the past 17 years. The growth is visible in Figure 4, where 1170 papers were written in 2017 on this topic alone. Nonetheless, the biggest increase is noted in 2017, when 206 more papers were written than the year before. The statistics were obtained from Google Scholar in June 2018.

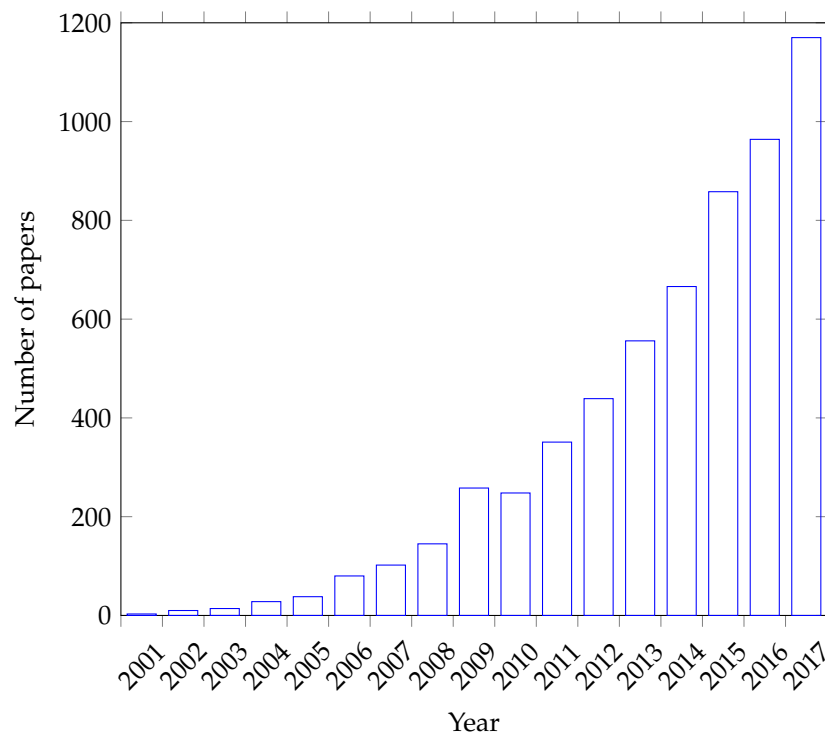


Figure 4. Hits for query (“swarm intelligence” AND “feature selection”) in Google Scholar.

5. Swarm Intelligence for Feature Selection

SI algorithms are a natural choice to be used for optimizing the feature subset selection process within a wrapper model approach. Wrapper models utilize a predefined ML algorithm (i.e., a classifier) to evaluate the quality of features and representational biases of the algorithm are avoided by the FS process. However, they have to run the ML algorithm many times to assess the quality of selected subsets of features, which is very computationally expensive. Thus, the SI algorithms can be used to optimize the process of FS within a wrapper model, their aim being to determine which subset of all possible features provides the best predictive performance when used with a predefined ML algorithm. In this manner, for an FS algorithm, the actual types and values of features are not important; they work well for either homogeneous or heterogeneous data that are continuous, discrete and/or mixed, or even represented as images or videos [28–33]. As classification is the most common ML task, we use it to demonstrate the FS approach; any other ML technique can be used similarly.

Given a predefined classification algorithm, a typical wrapper method iterates over the following two steps until the desired goal is achieved (i.e., certain accuracy is reached):

- Select a subset of features.
- Evaluate the selected subset by performing the selected classification algorithm.

A general framework for a wrapper model of FS for classification [27] is shown in Figure 5.

In the next subsections, the SI algorithms are analyzed based on the SI framework from Section 2. Only those derivations of the analyzed algorithms that comprise an FS have been considered.

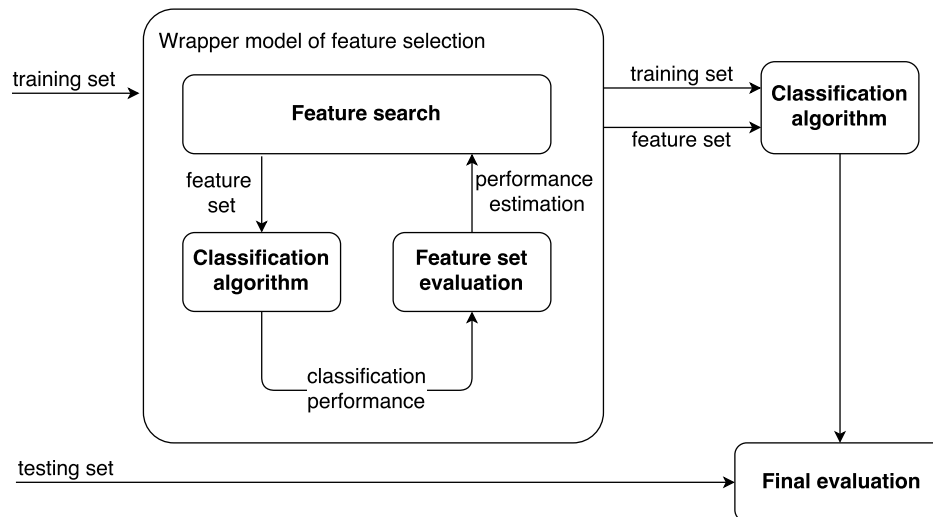


Figure 5. A general framework for wrapper models of feature selection.

5.1. Agent Representation

The continuous and binary versions of the SI algorithms have been used for FS. The main difference in SI approaches is in their representation of the search agent. Generally, in the binary versions of the algorithms, an agent is formed as a binary string, where the selected feature is represented with number 1 and the not selected feature with 0. In the continuous approaches, the search agent is represented with real values, and the selection of features is based on a threshold θ , $0 < \theta < 1$. The feature is selected only if its value is greater than the value assigned as a threshold.

Based on that, there are mainly three different encoding mechanisms. One of the most used is the binary encoding [30,34], followed by continuous encoding [29], and the hybrid combination of the two [31].

A particular case is the ACO algorithm, in which the features are represented as graph nodes. When an ant moves and visits the node in the graph, the node becomes part of the feature subset. Therefore, each feature subset is represented by an ant.

5.2. Initialize Population

Initialization of a population is the primary step in any SI technique, as shown in Section 2. Based on analyzed articles, six different strategies in population initialization are used commonly.

The first one is the initialization of the population with the full set of features (all features selected), also known as the backward selection. The opposite of this approach is the initialization of the population with an empty set of features (no feature is selected), also known as forward selection. Randomly generated swarm population is the third strategy, in which the choice of selecting a feature is random. Based on analyzed articles, this is also the one chosen most commonly.

The following strategies are some derivatives of the above approaches. One of them is a strategy in which only a small set of features is selected, and, with this, the forward selection is mimicked [32,34]. Swarm initialization with a large number of selected features mimicking the backward selection [32,34] is the next one.

Some approaches use continuous swarm intelligence techniques with the floating-point values of the initial population of agents [29].

In the ACO algorithm, the initialization is done mainly in such a way that the first node in the graph is selected randomly, and the following ones are based on the probabilistic transition rule [35]. There are also some variations of the graph representation in the way its nodes are connected. Although most algorithms use fully connected graph [35,36], which means the nodes are fully connected to each other, some researchers (e.g., [37]) connected each node with only two other nodes.

In [38], researchers initialized food sources in ABC with only one selected feature. The authors in [34,39] used four initialization approaches, i.e., normal, small, big, and mixed.

Another different approach was used in [40], where researchers used the Minimum Redundancy Maximum Relevance (MRMR) initialization which combines relevance with the target class and redundancy to other features.

5.3. Defined Stopping Condition

Stopping, or termination condition/criteria, is a parameter which is defined to stop the execution of an algorithm. In most SI algorithms, the stopping condition is either one, or is composed of two, separate conditions.

The most common stopping condition is the achieved number of iterations (n_ITER), which is sometimes referred to as generations. In the analyzed articles, researchers used the following number of iterations, ordered from the most used ones to the least: 100 [33,41–43], 50 [35,44,45], 500 [46], 200 [47], 70 [34], and 25 [36]. If a composed stopping condition is used, it is always combined with the number of iterations. Some other stopping conditions are: exceeded threshold value [48], optimal solution reached [49,50], and the objective function is equal to 0 [46].

5.4. Fitness Function

Each solution $\vec{X}_i = [x_{i1}, x_{i2}, \dots, x_{id}]$ (d and $i = 1, 2, \dots, n$ represents the dimension of solutions and number of solutions n , respectively) in the used SI technique must be evaluated by some fitness function $f(\vec{X}_i)$. Based on the algorithms presented in Section 2, the solution comprises mainly the search agent. For example, in the PSO, the search agent is a particle, in GSO a glow-worm, and in RIO a cockroach. The difference is, for example, in the ABC algorithm, where the solutions represent the food sources.

Fitness function is many times expressed as a classification accuracy. Many researchers use SVMs as the classifier method to evaluate solutions (e.g., [33,38,50–52]). Researchers in [46] proposed fitness function that is the average of the classification error rate obtained by three classifiers, SVM variations, ν -SVM, C-SVM and LS-SVM in the DA algorithm. The average results were used to prevent generation of a feature subset based on a single classifier. In [53], the C-SVM classifier using the Radial Basis Function kernel (RBF) was applied in the FA algorithm.

Classification performance is also evaluated frequently with KNN. In [36], the authors first evaluated each ant in ACO through KNN, and then used Mean Square Error (MSE) for performance evaluation. For the same algorithm, i.e., ACO, the researchers in [35] used KNN ($K = 1$). The classification performance was also evaluated with KNN ($K = 5$) in the binary ABC algorithm [54].

In [44], researchers proposed fitness function where each solution is measured with the Overall Classification Accuracy (OCA) in which the 1-Nearest Neighbor (1NN), K-Nearest Neighbor (KNN), or Weighted K-Nearest Neighbor (WKNN) is called. The value of K is changed dynamically based on iterations of the algorithm. With this approach, researchers enabled the diversity of the solutions in each iteration.

Composed weighted fitness function is also used in the GWO algorithm [40], which comprises an error rate and a number of selected features.

Some other approaches comprise classification accuracy of Optimum-Path Forest (OPF) in the BA algorithm [55], Root Mean Squared Error (RMSE) in FA [56], Greedy Randomized Adaptive Search Procedure (GRASP) algorithm for clustering in ABC [41], and Back Propagation Neural Network (BPNN) in RIO [49] and ACO [57].

Composed fitness function based on rough sets and classification quality is proposed in [58]. Packianather and Kapoor [48] proposed fitness function (Equation (1)),

$$\frac{1}{(k_1) \times ED + (k_2) \frac{n_s}{n_t}} \quad (1)$$

which is composed of Euclidean Distance (ED) used by the Minimum Distance Classifier (MDC) in the classifying process and some other parameters, i.e., weighted factors k_1 and k_2 , and the number of selected features and total number of features n_s and n_t , respectively.

Rough set reduct was also used in one of the first ACO algorithms for FS [59].

Authors in [47] proposed composite fitness function, which uses the classification accuracy and number of selected features $nbFeat$ (Equation (2)). SVM and NB classifiers obtained the classification accuracy, respectively.

$$90\% \times accuracy + 10\% \left(\frac{1}{nbFeat} \right) \tag{2}$$

In [43], the objective function is composed of the classification accuracy rate of KNN classifier ($K = 7$) and the measure class separability. Based on that, five variants were proposed, of which two are weighted sums. The first three use classification accuracy (f_1), Hausdorff distance (f_2), and Jeffries–Matusita (JM) distance (f_3), respectively; the fourth is the weighted sum of (f_1) and (f_2); and the last one is the weighted sum of (f_1) and (f_3).

The weighted fitness function is also used in [45]. The first part comprises the classification accuracy obtained by a KNN classifier ($K=1$) and the second part the number of selected features (Equation (3)).

$$w_1 \times acc(KNN) + w_2 \times \left(1 - \frac{n}{N} \right); w_1 = 1, w_2 = 0.001 \tag{3}$$

where

- n is the number of selected features,
- N is the number of all features,
- w_1 is the weight for classification accuracy,
- w_2 is the weight for selected features.

A similar approach is used in [34], where researchers proposed two variations of GWO, but in both, the goal of the fitness function is to maximize classification accuracy and lower the number of selected features. Here also has been applied the KNN classifier ($K = 5$).

5.5. Update and Move Agents

In the ABC algorithm, the neighborhood exploration is done by a small perturbation using Equation (4)

$$v_{ij} = x_{ij} + \Phi_{ij}(x_{ij} - x_{kj}) \tag{4}$$

where v_{ij} is the candidate food source, parameter x_{ij} represents the food source, indices j and k are random variables and i is the index of the current food source. Parameter Φ_{ij} is a real number in the interval $(-1, 1)$. In the proposed algorithm for FS based on ABC [38], the Modification Rate (MR), also called perturbation frequency, is used to select features with Equation (5):

$$x_i = \begin{cases} 1, & \text{if } R_i < MR, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

If the randomly generated value between 0 and 1 is lower than the MR , the feature is added to the feature subset. Otherwise, it is not. In [41], the update is done using the same equation as in the initially proposed ABC algorithm. Therefore, the sigmoid function is used (Equation (9)) for the transformation of values.

In [46], researchers proposed a two-sided DA algorithm in which the update of dragonflies' position is done using Equation (6):

$$\Delta X_{t+1} = (sS_i + aA_i + cC_i + dF_i + eE_i) + w\Delta X_t \tag{6}$$

The meaning of the components in Equation (6) are in the following order: Separation, alignment, cohesion, attraction, and distraction of the i th dragonfly, all summed with the inertia weight and the step. Discretization of variables' values is done using Equation (7):

$$X_{i,t+1} = \begin{cases} 1, & r < T(\Delta X_{i,t+1}), \\ 0, & r \geq T(\Delta X_{i,t+1}). \end{cases} \tag{7}$$

In Equation (7), the $T(\Delta X_{i,t+1})$ set the step for variables is based on Equation (8):

$$T(\Delta X_{i,t+1}) = \left| \frac{\Delta X_i}{\sqrt{\Delta X_i^2 + 1}} \right| \tag{8}$$

The researchers in [42] used sigmoid function sig_{ij} for the creation of the probability vector in the FA algorithm. The final position update is done after that with the Sigmoid function (Equation (9)) where the $rand$ is defined in $[0, 1]$.

$$x_{ij}^{t+1} = \begin{cases} 1, & \text{if } sig_{ij} \geq rand, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Instead of a Sigmoid function, researchers in [58] proposed the tanh function to improve the performance of an FA algorithm. The tanh function is responsible for scaling values to binary ones based on Equation (10) in which are the parameters $i = 1, \dots, n$ (n = number of fireflies) and $k = 1, \dots, m$ (m = number of dimensions).

$$f(X_i^k) = \frac{\exp(2X_i^k) - 1}{\exp(2X_i^k) + 1}. \tag{10}$$

The difference between Sigmoid and tanh functions is presented in Figure 6.

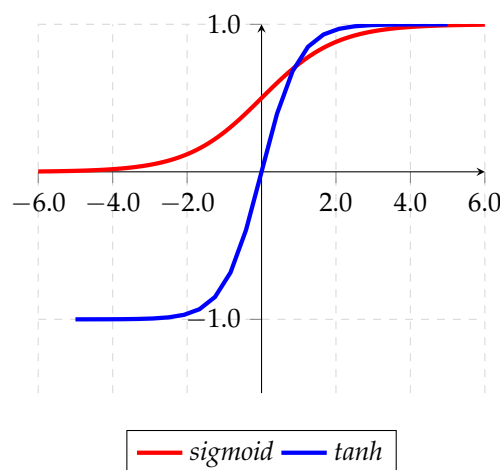


Figure 6. Sigmoid and tanh functions.

The movement of fireflies in the FA algorithm [53] is done using Equation (11):

$$X_i(t + 1) = X_i(t) + \beta(X_i(t) - X_j(t)) + \alpha(rand - \frac{1}{2}) \tag{11}$$

in which $X_i(t)$ and $X_j(t)$ determine the position of the less bright and brighter fireflies, respectively. Parameter β comprises the attractiveness component. Discretization is done using the Sigmoid function.

In [50], the change of glow-worms' positions is done using a step size calculated for each feature. The primary goal of glow-worms is to minimize the distance between features. The changes of chosen features are done in the GSO algorithm with Equation (12):

$$new_dim = \begin{cases} \text{ceil}(orig_dim - (rand * 5)), & \text{if probability} < 0.75, \\ \text{ceil}(orig_dim + (rand * 5)), & \text{if probability} > 0.75. \end{cases} \tag{12}$$

where $orig_dim$ and new_dim are original and new dimensions, respectively, and $rand = (0, 1)$.

Updating the roach location in the RIO algorithm is a composite of three parts [49]. The first part of Equation (13)

$$a = C_o \oplus CR(C_o \oplus CR(w \oplus MT(x_i), p_i), l_i)$$

$$x_i = \begin{cases} a, & \text{if } hunger_i < t_{hunger}, \\ random, & \text{otherwise.} \end{cases} \tag{13}$$

is velocity component $w \oplus MT(x_i)$ updated with mutation operator. The cognition part $C_o \oplus CR(w \oplus MT(x_i), p_i)$ comprises individual thinking of a cockroach, and is updated with a crossover or two point crossover operator. The last, social component, $C_o \oplus CR(C_o \oplus CR(w \oplus MT(x_i), p_i), l_i)$ comprises a collaboration among the swarm, and is updated with a crossover operator.

The Sigmoid function is used for transformation in continuous space, and is then compared to the randomly generated value in $(0, 1)$ in BBMO [44].

The movement of bats is affected by velocity and position [55]. To change the velocity in BA, the frequency must be updated first (Equation (14)). The velocity in the previous iteration, combined with the global best solution found so far for decision variable j (\hat{x}^j), the current value of decision value (x_i^j), and modified frequency, updates the new velocity shown in Equation (14)

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

$$v_i^j(t) = v_i^j(t - 1) + [\hat{x}^j - x_i^j(t - 1)]f_i \tag{14}$$

The position change is finally done based on the Sigmoid function.

The researchers in [34] proposed two variations of the GWO. In the first one, the update is done based on the calculated crossover between solutions (Equation (15)).

$$X_i^{t+1} = Crossover(x_1, x_2, x_3) \tag{15}$$

The parameters of the crossover are obtained based on the wolf movement towards the three best-known solutions, i.e., alpha, beta, and delta wolves. The second approach is a binary position vector changed by the Sigmoid function.

Researchers in [43] also used the GWO algorithm and updated the position according to Equation (16),

$$\vec{X}(t + 1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{16}$$

in which \vec{X}_1 to \vec{X}_3 are position vectors of best-known solutions/wolves. The new position is then discretized with Equation (17)

$$b(i, j) = \begin{cases} 1, & \text{if } X(i, j) \geq 0.5, \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

To sum up, based on the analyzed work, it has been shown that progress in the FS area has been made. There is a noticeable increasing extent of the proposed SI algorithms to solve FS problems, as well as mixed algorithms. In addition, there are increasing proposed novel fitness functions. An increase of determining the optimum parameter settings of algorithms has been seen in the already established algorithms.

6. Applications of Feature Selection with Swarm Intelligence

All proposed algorithms must be evaluated with either benchmark datasets or applied to real-life problems. Thus, the following section is organized in two subsections. Section 6.1 comprises research articles that only use available datasets for testing the algorithms, while Section 6.2 comprises research articles that apply proposed algorithms to the particular research field.

Artificially generated benchmarks, e.g., Griewank, Rosenbrock, and Ackley, are not addressed in a separate subsection because there are not many works that use this approach. Research articles in which the benchmark functions are subjected to testing are, for example, [60–62].

6.1. Datasets

The used datasets can be divided into three groups (Figure 7), regarding the number of features: Small datasets (up to 150 features), large datasets (more than 2000 features), and mixed datasets between the other two. The first group is composed of many datasets, such as Iris (4 features), Diabetes (8 features), Heart-Statlog (13 features), Wine (13 features), Australian Credit (14 features), Labor (16 features), Hepatitis (19 features), Auto (25 features), Ionosphere (34 features), Spambase (57 features), Sonar (60 features), Splice (62 features), 9_Tumors (90 features), and Mushrooms (112 features).

Researchers who tested the proposed algorithms only on small datasets are, for example, [38,44,63].

Large datasets consist mainly of medical data (DNA microarray datasets) such as ColonTumor (2000 features), Leukemia (5,147 features), Brain_Tumor1 (5920 features), Nervous-System and DLBCL-Harvard (both 7129 features), Brain_Tumor2 (10,367), 11_Tumors (12,533 features), LungCancer-Harvard1 (12,600 features), LungCancer-Harvard2 (12,534 features), 14_Tumors (15,009 features), Ovarian cancer (15,154 features), and Breast cancer (24,481 features).

Some researchers who used only such data are [45,46,50,64–66].

The remaining group is composed of mixed datasets. In [42], researchers used eleven large datasets and one small. Datasets from 73 up to 5000 features are used in [34]. Some other researchers in this group are, for example, [28,54,55,67,68].

The majority of datasets is available from four sources, i.e., UCI Machine Learning Repository [69], Gems System [70], LIBSVM Data [71], and Kent Ridge Bio-medical Dataset [72].

In Table 2, the research articles are collected and ordered based on used datasets' dimensions and taxonomy. We left out *Group Hunting*, because we were unable to find any work which included the HS algorithm for FS.

Table 2. Used dataset sizes in evaluating swarm intelligence (SI) for feature selection (FS).

	Small	Large	Mixed	Σ
Insect	ABC [38,41] BBMO [44,63] ACO [35] WSO [73]	DA [46] GSO [50]	ABC [54] FA [42] ACO [62]	11
Bacteria		BFO and BCO [64]		1
Bird	PSO [29,31]	PSO [66,74,75] [77]	PSO [28,30,76] [78]	10
Mammal	GWO [39] CaTS [65]		GWO [34,40] BA [55] WOA [67]	6
Fish	AFSA [51,52] FSA [79]			3
Frog		SFLA [45]		1
Other			KH [68,80]	2

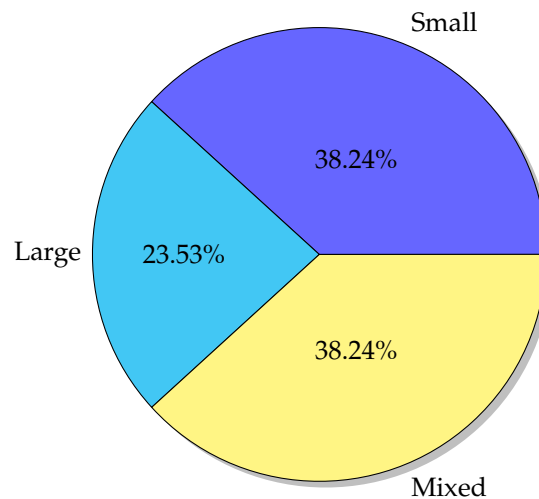


Figure 7. Analysis of used datasets in swarm intelligence (SI).

6.2. Application Areas

There are a variety of application areas in which the presented algorithms are used for FS. Based on that, we defined the following three main groups of application domains in which the algorithms fall (Figure 8):

1. Bio-Medical Engineering
2. Electrical and Electronic Engineering
3. Computer Engineering

The application field regulates all presented research articles and then classifies them based on the used algorithms in Table 3. For the same reason as in Table 2, we excluded the *Group Hunting* section.

Table 3. Application areas of SI for FS.

	Bio-Medical Engineering	Electrical and Electronic Engineering	Computer Engineering	Σ
Insects	DA[46] RIO [49]	FA [56]	ABC [33,41]	13
	GSO [50]		FA [53]	
	FA [58,81]	ACO [82]	ACO [83]	
	ACO [37,57]			
Bacteria	BFO [84]		BFO and BCO [64]	3
	BMA [85]			
Bird	PSO [66,74,75,86] [76,90,91]	PSO [87]	PSO [88,89]	10
Mammal	GWO [92,93]	GWO [43]	BA [47] GWO [94]	5
Fish	FSS [95]		AFSA [96]	2
Frog	SFLA [45,97,98]			3
Other	KH [99]		KH [100]	2

Based on the survey, we can state that Bio-Medical Engineering is the most used field for algorithms’ testing or applying. Researchers in [58] proposed the Chaos FA algorithm for heart disease prediction; the Bacterial Memetic Algorithm based feature selection for surface EMG-based hand motion recognition in long-term use was suggested by [85]. In [84], the authors tackled FS in Brain–Computer Interfacing (BCI) research with the BFO combined algorithm with learning automata. The proposed algorithm selects a feature subset using discrete wavelength transform, which is applied to break down the EEG data.

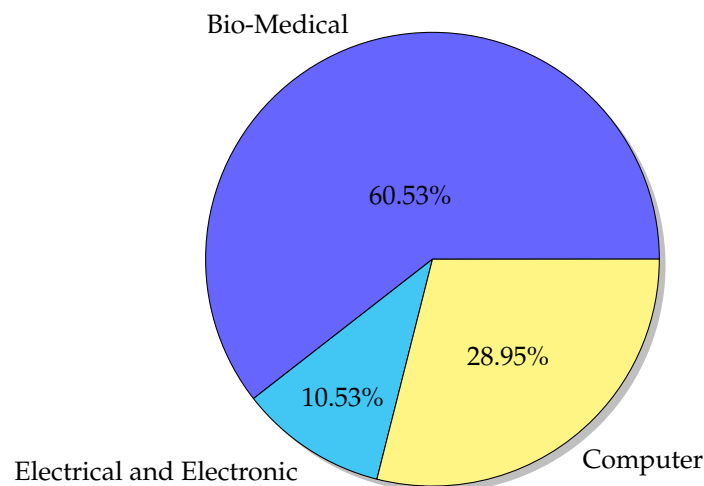


Figure 8. Analysis of applications in SI.

Microarray gene selection, or expression, is one field in Medical Engineering which is becoming more popular because it plays a crucial part in disease classifications. The researchers in [93] proposed binary GWO for classifying acute leukemia in which the multilayer perceptron neural network was used for fitness function.

Some studies [45,97] use the improved SFLA algorithm on high-dimensional biomedical data for disease diagnosis.

Image based FS is addressed in [99]. The researchers proposed a method for classifying lung Computed Tomography (CT) images, in which the KH algorithm was used alongside multilayer perceptron neural networks.

With binary FSS, the researchers in [95] proposed the application which predicts the readmission of Intensive Care Unit (ICU) patients within the time window of 24–72 h after being discharged.

Methodology IGWO-KELM for Medical Diagnosis was proposed in [92]. An Improved GWO algorithm (IGWO) is used to select the most informative subset of features, which are used in the second stage (KELM). In the latter, the prediction based on selected features in the previous stage is made.

For predicting metabolic syndrome disease, the adaptive binary PSO learnable Bayesian classifier is proposed in [90].

Research [49] addresses the problem of selecting textural features for determining the water content of cultured Sunagoke moss. The proposed approach comprises the RIO algorithm, in which the feature subset is evaluated with BPNN. A machine vision-based system is also used in addition to the latter in [86]. To improve the leap reading recognition, the BPSO is used with Euclidean Distances of subsets as the fitness function. Researchers in [48] proposed the BA algorithm for improvement in a wood defect classification.

The following studies have been done in the field of Electrical and Electronic Engineering. Authors in [56] used the FA algorithm in the near infrared (NIR) spectroscopy to find the most informative wavelengths. In the subarea of sensors, the researchers in [100] proposed a novel KH algorithm to improve gas recognition by electronic noses. Hyperspectral images are addressed in FS in [43,87]. The first paper uses the GWO algorithm for band selection. For hyperspectral image classification, researchers in the second article proposed the PSO-based automatic relevance determination and feature selection system.

In the Computer Engineering field, the intrusion detection is one of the most intriguing problems for researchers. It is addressed in [96] with the AFSA algorithm, in [47] with the BA* algorithm, and in [94] with the GWO algorithm. The dynamic subfield is also image steganalysis. The researchers in [33] presented a novel method named IFAB based on an ABC algorithm for feature selection, and the authors in [53] used the FA algorithm for blind image steganalysis. Some other problems tackled in this field comprise fault diagnosis of complex structures with the BFO algorithm [64], skin detection based background removal for enhanced face recognition with adaptive BPSO [89], identifying malicious web domains with BPSO in [88], and web page classification with ACO [83].

7. Discussion

Alone within this survey, we analyzed 22 different original SI algorithms for FS and, altogether, 64 of their variants. However, for the vast majority of all applications, only a few of the algorithms are used. In this manner, the most frequently used algorithm (PSO) is used in almost 47% of all cases, while the four most frequently used algorithms cover more than three quarters of cases (79%), and the top seven algorithms even cover 90% of all cases (see Table 4).

Table 4. Impact (share of citations) of SI algorithms for FS.

Algorithm	Publications	Citations	Share
PSO	16	766	46.62%
ACO	6	226	13.76%
BA	2	154	9.37%
ABC	4	152	9.25%
GWO	7	123	7.49%
FA	5	49	2.98%
WSO	1	26	1.58%
Others	23	147	8.95%

On the basis of the above, a reasonable doubt arises of whether one should even pursue the quest of analyzing the broad spectre of existing SI algorithms for FS. However, the fact is that some of the latest algorithms, e.g., BCO, CS, FA, and GWO, are also used in combination with some other techniques and present very promising results in FS. Some of the novel algorithms still provide enough improvements to be considered useful and produce an impact. Some of the selected algorithms and their components are sorted by proposed taxonomy in Table 5.

Thus, the goal of this paper is to support a researcher in choosing the most adequate algorithm and its setting for his/her data mining tasks and endeavors. For this purpose, alongside the characteristics of abundance of SI algorithms for FS, it may well help to be aware of various practical implications of these algorithms and their use. As some researchers report different modifications, adaptations and hybridizations of the basic algorithms, which have shown promising practical results, we collected some of them.

The introduction of different local search mechanisms is used to improve specific aspects of SI algorithms. For example, in [101], the Lévy flight is included to avoid premature convergence of the Shuffled Frog Leaping Algorithm (SFLA) for cancer classification. In [102], on the other hand, the authors proposed to combine tabu search (TS) and binary particle swarm optimization (BPSO) for feature selection, where BPSO acts as a local optimizer each time the TS has been run for a single generation.

In some cases, researchers combine specific characteristic of different SI algorithms to improve the overall efficiency. For example, in [103], the authors proposed a combination of Monkey Algorithm (MA) with Krill Herd Algorithm (KH) in order to balance the exploration and exploitation adaptively to find the optimal solution quickly. A similar case is presented in [81], where the diagnosis of childhood Atypical Teratoid/Rhabdoid Tumor (AT/RT) in Magnetic Resonance brain images and Hemochromatosis in Computed Tomography (CT) liver images has been improved through the hybridization of the Particle Swarm Optimization and Firefly (PSO-FF) algorithms.

Normally, in typical machine learning application, FS is used to optimize at least two major objectives: predictive performance and the number of selected features. To solve this multi-objective optimization problem, in some cases, researchers used several co-evolving swarms—so-called multi-swarms. For example, in [104], the authors developed a formulation utilizing two particles' swarms in order to optimize a desired performance criterion and the number of selected features simultaneously to reduce the dimensionality of spectral image data.

In [105], the authors proposed an SI FS algorithm, based on the initialization and update of only a subset of particles in the swarm, as they observed how feature selection failed to select small features subsets when discovering biomarkers in microarray data. Their approach increased the classification accuracy successfully, and decreased the number of selected features compared to other SI methods.

There are cases where the nature of the application requires very fast FS algorithms, such as data mining of data streams on the fly. An example of how to improve the speed of an SI based FS algorithm is presented in [77]. The authors proposed an accelerated PSO swarm search FS by simplifying some formulas for updating particles and applying the swarm search approach in incremental manner, obtaining a higher gain in accuracy per second incurred in the pre-processing.

Authors in [13] exposed that PSO has similar advantages as genetic algorithms in terms of a straightforward representation. However, neither of them can be used for feature construction. Moreover, scalability might be a bottleneck for both of them, especially when we deal with problems with thousands or more features [75]. Interestingly, we can support this claim by our study too, since nowadays, in the fields such as genetics and bioinformatics, the number of features may exceed a million. For such large dimensions, we need to use very specific algorithms as well as hardware. However, our study has not revealed any special solutions that apply feature selection methods using swarm intelligence on such dimensions. By the same token, no special datasets were found during our literature review.

Table 5. Summary of selected algorithms and their components sorted by proposed taxonomy.

Algorithm	Presentation; Initialization	Definition of Stop Condition	Evaluation	Move	Validation	Advantages
ABC [38]	binary; forward search strategy	100 iterations	acc (SVM)	with the perturbation frequency	10-fold CV	+ outperforms PSO, ACO, and GA + on average smaller number of selected features
ABC [54]	binary; random	50 iterations	classification error rate (5NN)	use of mutation, recombination and selection	70% training set, 30% test set, 10-fold CV on training set	+ introduced DE based neighborhood mechanism for FS + method removes redundant features effectively and is the most stable and consistent
ACO [35]	graph	50 iterations	acc (1NN)	graph search with the probabilistic transition rule	60% training set, 40% test set	+ ants have the authority to select or deselect visiting features; + method permits ants to explore all features; + ants can see all of the unvisited features simultaneously
PSO [75]	continuous; random	different tests	combined function with acc(adaptive KNN) and number of selected features	based on inertia weight, social and cognitive component	different CV techniques	+ high classification accuracy, small number of selected features, and lower computing time
GWO [92]	binary; random	100 iterations	combined function with acc(KELM) and number of selected features	based on alpha, beta, and delta wolves	10-fold CV	+ method adaptively converges more quickly, produces much better solution quality, but also gains less number of selected features, achieving high classification performance
WOA [67]	binary; random	100 iterations	combined function with acc (5NN) and number of selected features	based on neighbours	CV	+ method combined with the Simulated annealing following two hybrid models: LTH and HRH
AFSA [52]	binary; random	lack of changes in the optimal features subset for a period of 1 h	acc (SVM)	execution of the three search steps (follow, swarm, and prey)	10-fold CV	+ experiment with and without parameter optimization + dynamic vision proposed to assign an appropriate search space for each fish
SFLA [45]	binary; random with two equations	50 iterations	combined function with acc (1NN) and number of selected features	with equations containing adaptive transition factor	10-fold CV	+ provided three improvement strategies
KH [68]	binary; random	100 iterations	acc (OPF)	with Sigmoid function	5-fold CV	+ fast method

Additionally, characteristic of most SI algorithms is that they move individuals towards the best solution [13]. That helps them to achieve better and potentially faster convergence that is not always optimal. Interestingly, even though PSO is a good and fast search algorithm, it can converge prematurely, especially in complex search problems. Many variants were proposed to deal with this problem [31]. For example, an enhanced version of binary particle swarm optimization, designed to cope with premature convergence of the BPSO algorithm, is proposed in [30]. On the other hand, lack of explicit crossover can lead to local optima. Some methods, for example product graphs that are similar to arithmetic crossover in evolutionary algorithms, are proposed [106] to cope with this problem.

Particle swarm optimization algorithm is easy for implementation according to genetic algorithms or genetic programming, as stated in [13]. Contrarily, we cannot confirm that all SI algorithms are easy for implementation.

7.1. Issues and Open Questions

Our systematic review has revealed many issues, while some open questions that should be explored in the future years also appeared. Although research area is very vibrant, there is still room for improvement. Regarding the main issues, we can mention that some authors do not give special attention to the description of algorithms along with description of experiments. For that reason, many studies are not reproducible. It would be very important for the future that authors follow some standardized guidelines about presenting algorithms and experimental settings in scientific papers [107]. On the other hand, many authors used number of iterations (n_ITER) as termination criteria instead of number of function evaluations (n_FES). Because of that, many discrepancies appear during the comparison of various algorithms.

In line with this, papers should also rigorously report how parameter tuning was conducted. As we can see from different papers that were reviewed in our survey, many studies do not report how they find proper parameter settings, although parameter settings have a very big influence on the quality of results.

It would also be important to encourage more theoretical studies about various representation of individuals. There is still unknown which representation either binary or real is better for particular family of problems. On the other hand, scalability of SI algorithms seems to be also a very big bottleneck, especially for problems with thousands or more features. Nevertheless, studying the effect of parallelization might be a fruitful direction for future research. Finally, developing new visualization techniques for presenting results is also a promising direction for future.

7.2. Guidelines for SI Researchers

Based on presented topics in previous sections, we can provide some practical guidelines for researchers which would like to conduct research in this field.

If an author wants to propose a new SI algorithm, it must obey the defined steps of the SI framework (see Section 2). We would like to highlight that the future of this field is not in proposing new algorithms, but in improving existing ones. Sörensen in his paper [108] pointed out that instead of introducing novel algorithms, which do not always contribute to the field of meta-heuristics, the existing ones should be used in combination with some other techniques. In this manner, some of the latest algorithms, e.g., BCO, FA, and GWO, are used in combination with some other techniques and present promising results in FS [39,40,42,53,64].

The optimization ideas for algorithms are collected in Section 5. We provide a broad overview of different approaches that have been used in individual stages, i.e., initialization, defined stopping condition, fitness function, and update and move agents. Researchers can not only change a specific phase of their algorithm, but they can also combine algorithms. Thus, a hybrid version of algorithms can be proposed.

Each algorithm must also be tested. In this stage, there are three possible ways, as presented in Section 6. Researchers can test their algorithms on benchmarks, available datasets or in real life applications. Algorithms that have been tested on available datasets are collected in Section 6.1 and are best seen in Table 2. The latter is divided not only by the size of the datasets, i.e., the number of features that each instance has, but also by the proposed taxonomy (Figure 2). In addition, algorithms that have been tested on real-life applications are similarly displayed in Table 3. Therefore, authors can easily find relevant articles based on their wishes.

Researchers also have some problems regarding the implementation of the algorithms, as stated in Section 7.1. With this kind of problems, existing libraries can be very useful. One of those is NiaPy microframework for building nature-inspired algorithms [109]. NiaPy is intended for simple and quick use, without spending time for implementing algorithms from scratch.

A very important aspect touches also the design and conduction of experiments. A very big percentage of papers reported that comparison of algorithms were done according to the number iterations/generations. This should be avoided in the future, because number of iterations/generations do not ensure fair comparisons.

Since many SI algorithms have not been applied to the FS problem, the possibilities for further research in this field are vast.

8. Conclusions

This paper presents a comprehensive survey of SI algorithms for FS, which reduces the gap in this research field. We first analyzed 64 SI algorithms systematically, and sorted them as per the proposed taxonomy. After that, the SI for FS was analyzed based on the SI framework.

Different approaches are described in each framework phase of the SI algorithms. Moreover, to show the usage of FS in different application areas, we break down the presented papers in which researchers only used datasets for testing their methods and those who applied them to practical problems in the field.

Based on the analyzed data, and many papers comprising SI for FS as shown in Figure 2, it can be seen that the interest in this field is increasing, because we live in the era of Big Data, which is challenging not only for FS, but for all machine learning techniques. Therefore, the growing tendency of many researchers is to select only the most informative features out of the whole set of several thousands or even ten thousands of features. Such high dimensionality of data is most evident in Medicine in the DNA microarray datasets. Another reason for the increased research efforts in this field is the result of the SI algorithms' boom in recent years, which were later applied for the FS problem.

We think that the future in the SI field, not limited only to SI for FS, includes not only developing new algorithms based on the behavior of some animals, but also modifications of existing ones. The latter is expected to be done in the sense of combining different parts of algorithms together. Anyway, we strongly believe that the use of SI for FS will continue to grow and will provide effective and efficient solutions.

Let this review serve as a foundation for further research in this field. Researchers can focus on comparing different SI algorithms and their variants, and prepare in-depth empirical and analytic research. It would also be interesting to look at the computational performance and time complexity of such algorithms and compare their efficiency.

Because all of the novel algorithms must be tested, a systematization of datasets is recommended. As a basis, our division of datasets presented in Section 6.1 can be taken into account. We must also think in the direction of a unified measurement approach, which could improve comparison of the results. A more systematic approach to the evaluation of performance, advantages and drawbacks of specific algorithms would greatly improve the decision-making process of a data miner when facing the problem of feature selection.

It would also be fruitful to compare whether some algorithms perform better in some specific cases, e.g., on specific datasets, small/large number of features, etc., and if their hybridization, e.g.,

the introduction of some specific local search, affects all SI algorithms or just some of them. There is no obvious answer to what is the most effective SI algorithm in FS, for the following reasons. Firstly, the No Free Lunch Theorem [110] tells us a general-purpose, universal optimization strategy is impossible [111]. Furthermore, not all SI algorithms, as presented in Table 1, have been applied to the FS problems, and most of those that were lack good empirical evaluations. Experimental results obtained on different real-world problems are even harder to compare. For that reason, a systematical and exhaustive empirical investigation should be conducted in the future on how different algorithms, especially those which have shown the most promising results thus far, e.g., PSO, ACO, FA, GWO, DA, and GSO, and some promising newcomers, perform in varying conditions (on different datasets, for very big datasets, etc.). In this manner, the implementation issues and the impact of different configurations shall also be addressed.

Nowadays, the majority of papers that address the feature selection problem use instances that have up to several thousand features. From the papers, it is evident that researchers manage to adapt algorithms to work on such dimensions, but the question that arises is: What happens when we scale algorithms to problems that have millions of features (such as in genetics and bioinformatics)? Since there is a big scalability gap in this field, this problem is another one that needs to be addressed in the future.

Author Contributions: Conceptualization, L.B., I.F.J. and V.P.; Methodology, L.B.; Validation, L.B., I.F.J. and V.P.; Formal Analysis, L.B.; Writing—Review and Editing, L.B., I.F.J. and V.P.; and Visualization, L.B. and V.P.

Funding: The authors acknowledge the financial support from the Slovenian Research Agency (Research Core Funding No. P2-0057).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Swarm Intelligence Algorithms Sorted by Name

Table A1. Swarm Intelligence algorithms sorted by name.

Algorithm	Full Name	Reference
ABC	Artificial Bee Colony Algorithm	[112]
ABO	African Buffalo Optimization	[113]
ACO	Ant Colony Optimization	[8]
AFSA	Artificial Fish Swarm Algorithm	[114]
ALO	Antlion Optimizer	[115]
BA	Bat Algorithm	[116]
BA*	Bees Algorithm	[117]
BA**	Beaver Algorithm	[118]
BBMO	Bumble Bees Mating Optimization	[119]
BCO	Bee Colony Optimisation	[120]
BCO*	Bacterial Colony Optimization	[121]
BCPA	Bee Collecting Pollen Algorithm	[122]
BFO	Bacterial Foraging Optimization	[123]
BH	BeeHive	[124]
BLA	Bees Life Algorithm	[125]
BMO	Bird Mating Optimizer	[126]
BS	Bee System	[127]
BSO	Bees Swarm Optimization	[128]
CaTS	The Cat Swarm Optimization	[129]
CS	Cuckoo Search	[130]
CSO	Chicken Swarm Optimization	[131]
DA	Dragonfly Algorithm	[132]
DPO	Dolphin Partner Optimization	[133]
FA	Firefly Algorithm	[134]
FCA	Frog Calling Algorithm	[135]
FFOA	Fruit Fly Optimization Algorithm	[136]
FMHBO	Fast Marriage in Honey Bees Optimization	[137]

Table A1. Cont.

Algorithm	Full Name	Reference
FSA	Fish-Swarm Algorithm	[138]
FSS	Fish School Search	[139]
GEA	Group Escaping Algorithm	[140]
GOA	Grasshopper Optimisation Algorithm	[141]
GSO	Glow-worm Swarm Optimization	[142]
GWO	Grey Wolf Optimizer	[143]
HBA	Honey Bee Algorithm	[144]
HBF	Honey Bee Foraging	[145]
HBMO	Honey Bees Mating Optimization	[146]
HS	Hunting Search Algorithm	[147]
KH	Krill Herd Algorithm	[148]
MBO	Marriage in Honey Bees Optimization	[149]
MFO	Moth-Flame Optimization	[150]
MHSA	Mosquito Host-Seeking Algorithm	[151]
MiBO	Migrating Birds Optimization	[152]
PSO	Particle Swam Optimization	[7]
RIO	Roach Infestation Optimization	[153]
SFLA	Shuffled Frog Leaping Algorithm	[154]
SMOA	Slime Mould Optimization Algorithm	[155]
SSA	Shark-Search Algorithm	[156]
SuA	Superbug Algorithm	[157]
TA	Termite Algorithm	[158]
VBA	Virtual Bee Algorithm	[159]
WCA	Wolf Colony Algorithm	[160]
WOA	Whale Optimization Algorithm	[161]
WPS	Wolf Pack Search Algorithm	[162]
WSO	Wasp Swarm Optimization	[163]

References

- Cao, J.; Cui, H.; Shi, H.; Jiao, L. Big Data: A Parallel Particle Swarm Optimization-Back-Propagation Neural Network Algorithm Based on MapReduce. *PLoS ONE* **2016**, *11*, e157551. [[CrossRef](#)] [[PubMed](#)]
- Cheng, S.; Shi, Y.; Qin, Q.; Bai, R. Swarm Intelligence in Big Data Analytics. In *Intelligent Data Engineering and Automated Learning—IDEAL 2013*; Yin, H., Tang, K., Gao, Y., Klawonn, F., Lee, M., Weise, T., Li, B., Yao, X., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 417–426.
- Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed.; Springer: New York, NY, USA, 2009; p. 745.
- Blum, C.; Li, X. Swarm Intelligence in Optimization. In *Swarm Intelligence*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 43–85.
- Hassanien, A.E.; Emary, E. *Swarm Intelligence: Principles, Advances, and Applications*; CRC Press: Boca Raton, FL, USA, 2016.
- Garey, M.R.; Johnson, D.S. *Computers and Intractability, a Guide to the Theory of NP-Completeness*; W. H. Freeman & Co.: New York, NY, USA, 1979.
- Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks (ICNN '95), Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948. [[CrossRef](#)]
- Dorigo, M. Learning and Natural Algorithms. Ph.D. Thesis, Politecnico di Milano, Milano, Italy, 1992.
- Parpinelli, R.; Lopes, H. New inspirations in swarm intelligence: A survey. *Int. J. Bio-Inspir. Comput.* **2011**, *3*, 1–16. [[CrossRef](#)]
- Kar, A.K. Bio inspired computing—A review of algorithms and scope of applications. *Expert Syst. Appl.* **2016**, *59*, 20–32. [[CrossRef](#)]
- Fong, S.; Deb, S.; Yang, X.S.; Li, J. Feature Selection in Life Science Classification: Metaheuristic Swarm Search. *IT Prof.* **2014**, *16*, 24–29. [[CrossRef](#)]
- Basir, M.A.; Ahmad, F. Comparison on Swarm Algorithms for Feature Selections Reductions. *Int. J. Sci. Eng. Res.* **2014**, *5*, 479–486.

13. Xue, B.; Zhang, M.; Browne, W.N.; Yao, X. A Survey on Evolutionary Computation Approaches to Feature Selection. *IEEE Trans. Evol. Comput.* **2016**, *20*, 606–626. [[CrossRef](#)]
14. Beni, G.; Wang, J. Swarm Intelligence in Cellular Robotic Systems. In *Robots and Biological Systems: Towards a New Bionics?* Dario, P., Sandini, G., Aebischer, P., Eds.; Springer: Berlin/Heidelberg, Germany, 1993; pp. 703–712.
15. Engelbrecht, A.P. *Computational Intelligence: An introduction*; Wiley: Hoboken, NJ, USA, 2007.
16. Millonas, M.M. Swarms, Phase Transitions, and Collective Intelligence, arXiv 1993, arXiv:adap-org/9306002. [[CrossRef](#)]
17. Olariu, S.; Zomaya, A.Y. *Handbook of Bioinspired Algorithms and Applications*; Chapman and Hall/CRC: London, UK, 2005; Volume 5, p. 704.
18. Fister, I., Jr.; Yang, X.S.; Fister, I.; Brest, J.; Fister, D. A Brief Review of Nature-Inspired Algorithms for Optimization. *Elektroteh. Vestn.* **2013**, *80*, 116–122.
19. Mucherino, A.; Seref, O. Monkey search: A novel metaheuristic search for global optimization. *AIP Conf. Proc.* **2007**, *953*, 162–173. [[CrossRef](#)]
20. Kaveh, A.; Farhoudi, N. A new optimization method: Dolphin echolocation. *Adv. Eng. Softw.* **2013**, *59*, 53–70. [[CrossRef](#)]
21. Chen, C.C.; Tsai, Y.C.; Liu, I.I.; Lai, C.C.; Yeh, Y.T.; Kuo, S.Y.; Chou, Y.H. A Novel Metaheuristic: Jaguar Algorithm with Learning Behavior. In Proceedings of the 2015 IEEE International Conference on Systems, Man, and Cybernetics, Kowloon, China, 9–12 October 2015; pp. 1595–1600. [[CrossRef](#)]
22. Montalvo, I.; Izquierdo, J. Agent Swarm Optimization: Exploding the search space. In *Machine Learning for Cyber Physical Systems*; Beyerer, J., Niggemann, O., Kühnert, C., Eds.; Springer: Berlin/Heidelberg, Germany, 2017; pp. 55–64.
23. Montalvo, I.; Izquierdo, J.; Pérez-García, R.; Herrera, M. Water Distribution System Computer-Aided Design by Agent Swarm Optimization. *Comput.-Aided Civ. Infrastruct. Eng.* **2014**, *29*, 433–448. [[CrossRef](#)]
24. Liu, H.; Yu, L. Toward integrating feature selection algorithms for classification and clustering. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 491–502.
25. Tang, J.; Alelyani, S.; Liu, H. Feature selection for classification: A review. In *Data Classification: Algorithms and Applications*; CRC Press: Boca Raton, FL, USA, 2014; p. 37.
26. Liu, H.; Motoda, H. *Computational Methods of Feature Selection*; CRC Press: Boca Raton, FL, USA, 2007.
27. Kohavi, R.; John, G.H. Wrappers for feature subset selection. *Artif. Intell.* **1997**, *97*, 273–324. [[CrossRef](#)]
28. Brezočnik, L. Feature Selection for Classification Using Particle Swarm Optimization. In Proceedings of the 17th IEEE International Conference on Smart Technologies (IEEE EUROCON 2017), Ohrid, Macedonia, 6–8 July 2017; pp. 966–971.
29. Lin, S.W.; Ying, K.C.; Chen, S.C.; Lee, Z.J. Particle swarm optimization for parameter determination and feature selection of support vector machines. *Expert Syst. Appl.* **2008**, *35*, 1817–1824. [[CrossRef](#)]
30. Vieira, S.M.; Mendonça, L.F.; Farinha, G.J.; Sousa, J.M. Modified binary PSO for feature selection using SVM applied to mortality prediction of septic patients. *Appl. Soft Comput.* **2013**, *13*, 3494–3504. [[CrossRef](#)]
31. Boubezoul, A.; Paris, S. Application of global optimization methods to model and feature selection. *Pattern Recognit.* **2012**, *45*, 3676–3686. [[CrossRef](#)]
32. Xue, B.; Zhang, M.; Browne, W.N. Novel Initialisation and Updating Mechanisms in PSO for Feature Selection in Classification. In *Applications of Evolutionary Computation: 16th European Conference, EvoApplications 2013*; Springer: Vienna, Austria, 2013; pp. 428–438.
33. Mohammadi, F.G.; Abadeh, M.S. Image steganalysis using a bee colony based feature selection algorithm. *Eng. Appl. Artif. Intell.* **2014**, *31*, 35–43. [[CrossRef](#)]
34. Emary, E.; Zawbaa, H.M.; Hassanien, A.E. Binary grey wolf optimization approaches for feature selection. *Neurocomputing* **2016**, *172*, 371–381. [[CrossRef](#)]
35. Kashef, S.; Nezamabadi-pour, H. An advanced ACO algorithm for feature subset selection. *Neurocomputing* **2015**, *147*, 271–279. [[CrossRef](#)]
36. Kanan, H.R.; Faez, K.; Taheri, S.M. Feature Selection Using Ant Colony Optimization (ACO): A New Method and Comparative Study in the Application of Face Recognition System. In *Advances in Data Mining. Theoretical Aspects and Applications: 7th Industrial Conference, ICDM 2007*; Perner, P., Ed.; Springer: Leipzig, Germany, 2007; Volume 4597, pp. 63–76.

37. Yu, H.; Gu, G.; Liu, H.; Shen, J.; Zhao, J. A Modified Ant Colony Optimization Algorithm for Tumor Marker Gene Selection. *Genom. Proteom. Bioinform.* **2009**, *7*, 200–208. [[CrossRef](#)]
38. Schiezzaro, M.; Pedrini, H. Data feature selection based on Artificial Bee Colony algorithm. *EURASIP J. Image Video Process.* **2013**, *2013*, 47. [[CrossRef](#)]
39. Emary, E.; Zawbaa, H.M.; Grosan, C.; Hassenian, A.E. Feature Subset Selection Approach by Gray-Wolf Optimization. In *Afro-European Conference for Industrial Advancement: Proceedings of the First International Afro-European Conference for Industrial Advancement AECIA 2014*; Abraham, A., Krömer, P., Snasel, V., Eds.; Springer: Cham, Switzerland, 2015; pp. 1–13.
40. Emary, E.; Zawbaa, H.M.; Grosan, C. Experienced Gray Wolf Optimization Through Reinforcement Learning and Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, 1–14. [[CrossRef](#)] [[PubMed](#)]
41. Marinakis, Y.; Marinaki, M.; Matsatsinis, N. A hybrid discrete Artificial Bee Colony—GRASP algorithm for clustering. In *Proceedings of the 2009 International Conference on Computers & Industrial Engineering*, Troyes, France, 6–9 July 2009; pp. 548–553. [[CrossRef](#)]
42. Zhang, L.; Shan, L.; Wang, J. Optimal feature selection using distance-based discrete firefly algorithm with mutual information criterion. *Neural Comput. Appl.* **2016**, *28*, 2795–2808. [[CrossRef](#)]
43. Medjahed, S.A.; Ait Saadi, T.; Benyettou, A.; Ouali, M. Gray Wolf Optimizer for hyperspectral band selection. *Appl. Soft Comput. J.* **2016**, *40*, 178–186. [[CrossRef](#)]
44. Marinaki, M.; Marinakis, Y. A bumble bees mating optimization algorithm for the feature selection problem. *Handb. Swarm Intell.* **2016**, *7*, 519–538. [[CrossRef](#)]
45. Hu, B.; Dai, Y.; Su, Y.; Moore, P.; Zhang, X.; Mao, C.; Chen, J.; Xu, L. Feature Selection for Optimized High-dimensional Biomedical Data using the Improved Shuffled Frog Leaping Algorithm. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2016**. [[CrossRef](#)] [[PubMed](#)]
46. Medjahed, S.A.; Saadi, T.A.; Benyettou, A.; Ouali, M. Kernel-based learning and feature selection analysis for cancer diagnosis. *Appl. Soft Comput.* **2017**, *51*, 39–48. [[CrossRef](#)]
47. Enache, A.C.; Sgarciu, V.; Petrescu-Nita, A. Intelligent feature selection method rooted in Binary Bat Algorithm for intrusion detection. In *Proceedings of the 10th Jubilee IEEE International Symposium on Applied Computational Intelligence and Informatics, Proceedings (SACI 2015)*, Timisoara, Romania, 21–23 May 2015; pp. 517–521. [[CrossRef](#)]
48. Packianather, M.S.; Kapoor, B. A wrapper-based feature selection approach using Bees Algorithm for a wood defect classification system. In *Proceedings of the 10th System of Systems Engineering Conference, SoSE 2015*, San Antonio, TX, USA, 17–20 May 2015; pp. 498–503. [[CrossRef](#)]
49. Hendrawan, Y.; Murase, H. Neural-Discrete Hungry Roach Infestation Optimization to Select Informative Textural Features for Determining Water Content of Cultured Sunagoke Moss. *Environ. Control Biol.* **2011**, *49*, 1–21. [[CrossRef](#)]
50. Gurav, A.; Nair, V.; Gupta, U.; Valadi, J. Glowworm Swarm Based Informative Attribute Selection Using Support Vector Machines for Simultaneous Feature Selection and Classification. In *5th International Conference Swarm, Evolutionary, and Memetic Computing*; Springer: Cham, Switzerland, 2015; Volume 8947, pp. 27–37.
51. Lin, K.C.; Chen, S.Y.; Hung, J.C. Feature Selection for Support Vector Machines Base on Modified Artificial Fish Swarm Algorithm. In *Ubiquitous Computing Application and Wireless Sensor, Lecture Notes in Electrical Engineering 331*; Springer: Dordrecht, The Netherlands, 2015.
52. Lin, K.C.; Chen, S.Y.; Hung, J.C. Feature Selection and Parameter Optimization of Support Vector Machines Based on Modified Artificial Fish Swarm Algorithms. *Math. Probl. Eng.* **2015**, *2015*, 604108. [[CrossRef](#)]
53. Chhikara, R.R.; Sharma, P.; Singh, L. An improved dynamic discrete firefly algorithm for blind image steganalysis. *Int. J. Mach. Learn. Cybern.* **2016**, *9*, 821–835. [[CrossRef](#)]
54. Hancer, E.; Xue, B.; Karaboga, D.; Zhang, M. A binary ABC algorithm based on advanced similarity scheme for feature selection. *Appl. Soft Comput. J.* **2015**, *36*, 334–348. [[CrossRef](#)]
55. Nakamura, R.Y.M.; Pereira, L.A.M.; Costa, K.A.; Rodrigues, D.; Papa, J.P.; Yang, X.S. BBA: A binary bat algorithm for feature selection. In *Proceedings of the Brazilian Symposium of Computer Graphic and Image Processing, Ouro Preto, Brazil, 22–25 August 2012*; pp. 291–297. [[CrossRef](#)]
56. Goodarzi, M.; dos Santos Coelho, L. Firefly as a novel swarm intelligence variable selection method in spectroscopy. *Anal. Chim. Acta* **2014**, *852*, 20–27. [[CrossRef](#)] [[PubMed](#)]

57. Erguzel, T.T.; Ozekes, S.; Gultekin, S.; Tarhan, N. Ant Colony Optimization Based Feature Selection Method for QEEG Data Classification. *Psychiatry Investig.* **2014**, *11*, 243–250. [[CrossRef](#)] [[PubMed](#)]
58. Long, N.C.; Meesad, P.; Unger, H. A highly accurate firefly based algorithm for heart disease prediction. *Expert Syst. Appl.* **2015**, *42*, 8221–8231. [[CrossRef](#)]
59. Jensen, R.; Jensen, R.; Shen, Q. Finding Rough Set Reducts with Ant Colony Optimization. In Proceedings of the 2003 UK Workshop on Computational Intelligence, Guilford, UK, 9–11 September 2003; pp. 15–22.
60. Lee, S.; Soak, S.; Oh, S.; Pedrycz, W.; Jeon, M. Modified binary particle swarm optimization. *Prog. Nat. Sci.* **2008**, *18*, 1161–1166. [[CrossRef](#)]
61. Khanesar, M.A.; Teshnehlab, M.; Shoorehdeli, M.A. A novel binary particle swarm optimization. In Proceedings of the IEEE 2007 Mediterranean Conference on Control & Automation, Athens, Greece, 27–29 June 2007; pp. 1–6. [[CrossRef](#)]
62. Kabir, M.M.; Shahjahan, M.; Murase, K. A new hybrid ant colony optimization algorithm for feature selection. *Expert Syst. Appl.* **2012**, *39*, 3747–3763. [[CrossRef](#)]
63. Marinakis, Y.; Marinaki, M.; Matsatsinis, N. A Hybrid Bumble Bees Mating Optimization—GRASP Algorithm for Clustering. In *Hybrid Artificial Intelligence Systems, Proceedings of the 4th International Conference, HAIS 2009, Salamanca, Spain, 10–12 June 2009*; Corchado, E., Wu, X., Oja, E., Herrero, Á., Baroque, B., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 549–556. [[CrossRef](#)]
64. Wang, H.; Jing, X.; Niu, B. Bacterial-inspired feature selection algorithm and its application in fault diagnosis of complex structures. In Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 24–29 July 2016; pp. 3809–3816. [[CrossRef](#)]
65. Lin, K.C.; Zhang, K.Y.; Huang, Y.H.; Hung, J.C.; Yen, N. Feature selection based on an improved cat swarm optimization algorithm for big data classification. *J. Supercomput.* **2016**, *72*, 3210–3221. [[CrossRef](#)]
66. Dara, S.; Banka, H. A Binary PSO Feature Selection Algorithm for Gene Expression Data. In Proceedings of the 2014 International Conference on Advances in Communication and Computing Technologies, Mumbai, India, 10–11 August 2014; pp. 1–6.
67. Mafarja, M.M.; Mirjalili, S. Hybrid Whale Optimization Algorithm with simulated annealing for feature selection. *Neurocomputing* **2017**, *260*, 302–312. [[CrossRef](#)]
68. Rodrigues, D.; Pereira, L.A.M.; Papa, J.P.; Weber, S.A.T. A binary krill herd approach for feature selection. *Proc. Int. Conf. Pattern Recognit.* **2014**, *1407*, 1407–1412. [[CrossRef](#)]
69. Dheeru, D.; Karra Taniskidou, E. UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences, 2017. Available online: <http://archive.ics.uci.edu/ml>. (accessed on June 20 2018).
70. Statnikov, A.; Tsamardinos, I.; Dosbayev, Y.; Aliferis, C.F. GEMS: A system for automated cancer diagnosis and biomarker discovery from microarray gene expression data. *Int. J. Med. Inform.* **2005**, *74*, 491–503. [[CrossRef](#)] [[PubMed](#)]
71. Chang, C.-C.; Lin, C.-J.. LIBSVM : a library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 27:1–27:27. [[CrossRef](#)]
72. Li, J. Kent Ridge Bio-medical Dataset. Available online: <http://leo.ugr.es/elvira/DBCRepository/> (accessed on 20 June 2018).
73. Fan, H.; Zhong, Y. A Rough Set Approach to Feature Selection Based on Wasp Swarm Optimization. *J. Comput. Inf. Syst.* **2012**, *8*, 1037–1045.
74. Chen, K.H.; Wang, K.J.; Tsai, M.L.; Wang, K.M.; Adrian, A.M.; Cheng, W.C.; Yang, T.S.; Teng, N.C.; Tan, K.P.; Chang, K.S. Gene selection for cancer identification: A decision tree model empowered by particle swarm optimization algorithm. *BMC Bioinform.* **2014**, *15*, 49. [[CrossRef](#)] [[PubMed](#)]
75. Kar, S.; Das Sharma, K.; Maitra, M. Gene selection from microarray gene expression data for classification of cancer subgroups employing PSO and adaptive K-nearest neighborhood technique. *Expert Syst. Appl.* **2015**, *42*, 612–627. [[CrossRef](#)]
76. Pashaei, E.; Ozen, M.; Aydin, N. Improving medical diagnosis reliability using Boosted C5.0 decision tree empowered by Particle Swarm Optimization. In Proceedings of the 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) Milan, Italy, 25–29 August 2015; pp. 7230–7233. [[CrossRef](#)]
77. Fong, S.; Wong, R.; Vasilakos, A. Accelerated PSO Swarm Search Feature Selection for Data Stream Mining Big Data. *IEEE Trans. Serv. Comput.* **2015**, 33–45. [[CrossRef](#)]

78. Tran, B.; Xue, B.; Zhang, M. A New Representation in PSO for Discretization-Based Feature Selection. *IEEE Trans. Cybern.* **2017**. [[CrossRef](#)] [[PubMed](#)]
79. Chen, Y.; Zhu, Q.; Xu, H. Finding rough set reducts with fish swarm algorithm. *Knowl.-Based Syst.* **2015**, *81*, 22–29. [[CrossRef](#)]
80. Shahana, A.H.; Preeja, V. A binary krill herd approach based feature selection for high dimensional data. In Proceedings of the IEEE 2016 International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 26–27 August 2016; pp. 1–6. [[CrossRef](#)]
81. Thamarachelvi, B.; Yamuna., G. Hybrid Firefly Swarm Intelligence Based Feature Selection for Medical Data Classification and Segmentation in SVD—NSCT Domain. *Int. J. Adv. Res.* **2016**, *4*, 744–760. [[CrossRef](#)]
82. Wan, Y.; Wang, M.; Ye, Z.; Lai, X. A feature selection method based on modified binary coded ant colony optimization algorithm. *Appl. Soft Comput.* **2016**, *49*, 248–258. [[CrossRef](#)]
83. Saraç, E.; Özel, S.A. An ant colony optimization based feature selection for web page classification. *Sci. World J.* **2014**, *2014*, 649260. [[CrossRef](#)] [[PubMed](#)]
84. Pal, M.; Bhattacharyya, S.; Roy, S.; Konar, A.; Tibarewala, D.; Janarthan, R. A bacterial foraging optimization and learning automata based feature selection for motor imagery EEG classification. In Proceedings of the IEEE 2014 International Conference on Signal Processing and Communications (SPCOM), Bangalore, India, 22–25 July 2014; pp. 1–5. [[CrossRef](#)]
85. Zhou, D.; Fang, Y.; Botzheim, J.; Kubota, N.; Liu, H. Bacterial Memetic Algorithm based Feature Selection for Surface EMG based Hand Motion Recognition in Long-term Use. In Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence, Athens, Greece, 6–9 December 2016; pp. 1–7.
86. Wang, M.; Wang, X.; Li, G. A improved speech synthesis system utilizing BPSO-based lip feature selection. In Proceedings of the IEEE 2011 4th International Conference on Biomedical Engineering and Informatics (BMEI), Shanghai, China, 2011; pp. 1292–1295. [[CrossRef](#)]
87. Zhang, X.; Wang, W.; Li, Y.; Jiao, L. PSO-based automatic relevance determination and feature selection system for hyperspectral image classification. *Electron. Lett.* **2012**, *48*, 1263–1265. [[CrossRef](#)]
88. Hu, Z.; Chiong, R.; Pranata, I.; Susilo, W.; Bao, Y. Identifying malicious web domains using machine learning techniques with online credibility and performance data. In Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 24–29 July 2016; pp. 5186–5194. [[CrossRef](#)]
89. Sattiraju, M.; Manikandan, M.V.; Manikantan, K.; Ramachandran, S. Adaptive BPSO based feature selection and skin detection based background removal for enhanced face recognition. In Proceedings of the 2013 IEEE Fourth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), Jodhpur, India, 18–21 December 2013; pp. 1–4. [[CrossRef](#)]
90. Dehuri, S.; Roy, R.; Cho, S.B. An adaptive binary PSO to learn bayesian classifier for prognostic modeling of metabolic syndrome. In Proceedings of the 13th annual conference companion on Genetic and evolutionary computation (GECCO '11), Dublin, Ireland, 12–16 July 2011; ACM Press: New York, NY, USA, 2011; pp. 495–502. [[CrossRef](#)]
91. Chen, K.H.; Wang, K.J.; Wang, K.M.; Angelia, M.A. Applying particle swarm optimization-based decision tree classifier for cancer classification on gene expression data. *Appl. Soft Comput.* **2014**, *24*, 773–780. [[CrossRef](#)]
92. Li, Q.; Chen, H.; Huang, H.; Zhao, X.; Cai, Z.; Tong, C.; Liu, W.; Tian, X. An Enhanced Grey Wolf Optimization Based Feature Selection Wrapped Kernel Extreme Learning Machine for Medical Diagnosis. *Comput. Math. Methods Med.* **2017**, *2017*, 9512741. [[CrossRef](#)] [[PubMed](#)]
93. Manikandan, S.; Manimegalai, R.; Hariharan, M. Gene Selection from Microarray Data Using Binary Grey Wolf Algorithm for Classifying Acute Leukemia. *Curr. Signal Transduct. Ther.* **2016**, *11*, 76–78. [[CrossRef](#)]
94. Seth, J.K.; Chandra, S. Intrusion detection based on key feature selection using binary GWO. In Proceedings of the IEEE 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 16–18 March 2016; pp. 3735–3740.
95. Sargo, J.A.G.; Vieira, S.M.; Sousa, J.M.C.; Filho, C.J.B. Binary Fish School Search applied to feature selection: Application to ICU readmissions. In Proceedings of the 2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Beijing, China, 6–11 July 2014; pp. 1366–1373. [[CrossRef](#)]
96. Wang, G.; Dai, D. Network Intrusion Detection Based on the Improved Artificial Fish Swarm Algorithm. *J. Comput.* **2013**, *8*, 2990–2996. [[CrossRef](#)]

97. Dai, Y.; Hu, B.; Su, Y.; Mao, C.; Chen, J.; Zhang, X.; Moore, P.; Xu, L.; Cai, H. Feature selection of high-dimensional biomedical data using improved SFLA for disease diagnosis. In Proceedings of the 2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Washington, DC, USA, 9–12 November 2015; pp. 458–463. [\[CrossRef\]](#)
98. Ladgham, A.; Torkhani, G.; Sakly, A.; Mtibaa, A. Modified support vector machines for MR brain images recognition. In Proceedings of the 2013 IEEE International Conference on Control, Decision and Information Technologies (CoDIT), Hammamet, Tunisia, 6–8 May 2013; pp. 32–35. [\[CrossRef\]](#)
99. Baranidharan, T.; Sumathi, T.; Shekar, V. Weight Optimized Neural Network Using Metaheuristics for the Classification of Large Cell Carcinoma and Adenocarcinoma from Lung Imaging. *Curr. Signal Transduct. Ther.* **2016**, *11*, 91–97. [\[CrossRef\]](#)
100. Wang, L.; Jia, P.; Huang, T.; Duan, S.; Yan, J.; Wang, L. A Novel Optimization Technique to Improve Gas Recognition by Electronic Noses Based on the Enhanced Krill Herd Algorithm. *Sensors* **2016**, *16*, 1275. [\[CrossRef\]](#) [\[PubMed\]](#)
101. Gunavathi, C.; Premalatha, K. A comparative analysis of swarm intelligence techniques for feature selection in cancer classification. *Sci. World J.* **2014**, *2014*, 693831. [\[CrossRef\]](#) [\[PubMed\]](#)
102. Chuang, L.Y.; Yang, C.H.; Yang, C.H. Tabu Search and Binary Particle Swarm Optimization for Feature Selection Using Microarray Data. *J. Comput. Biol.* **2009**, *16*, 1689–1703. [\[CrossRef\]](#) [\[PubMed\]](#)
103. Hafez, A.I.; Hassanien, A.E.; Zawbaa, H.M. Hybrid Swarm Intelligence Algorithms for Feature Selection: Monkey and Krill Herd Algorithms. In Proceedings of the IEEE International Computer Engineering Conference—ICENCO, Cairo, Egypt, 29–30 December 2015.
104. Monteiro, S.T.; Kosugi, Y. Applying Particle Swarm Intelligence for Feature Selection of Spectral Imagery. In Proceedings of the IEEE Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007), Rio de Janeiro, Brazil, 20–24 October 2007; pp. 933–938. [\[CrossRef\]](#)
105. Martinez, E.; Alvarez, M.M.; Trevino, V. Compact cancer biomarkers discovery using a swarm intelligence feature selection algorithm. *Comput. Biol. Chem.* **2010**, *34*, 244–250. [\[CrossRef\]](#) [\[PubMed\]](#)
106. Fister, I., Jr.; Tepeh, A.; Brest, J.; Fister, I. Population Size Reduction in Particle Swarm Optimization Using Product Graphs. In *Mendel 2015*; Springer: Cham, Switzerland, 2015; pp. 77–87.
107. Črepinšek, M.; Liu, S.H.; Mernik, M. Replication and comparison of computational experiments in applied evolutionary computing: common pitfalls and guidelines to avoid them. *Appl. Soft Comput.* **2014**, *19*, 161–170. [\[CrossRef\]](#)
108. Sörensen, K. Metaheuristics—The metaphor exposed. *Int. Trans. Oper. Res.* **2013**, *3*–18. [\[CrossRef\]](#)
109. Vrbančič, G.; Brezočnik, L.; Mlakar, U.; Fister, D.; Fister, I., Jr. NiaPy: Python microframework for building nature-inspired algorithms. *J. Open Source Softw.* **2018**, *3*. [\[CrossRef\]](#)
110. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [\[CrossRef\]](#)
111. Ho, Y.C.; Pepyne, D.L. Simple explanation of the no-free-lunch theorem and its implications. *J. Optim. Theory Appl.* **2002**, *115*, 549–570. [\[CrossRef\]](#)
112. Karaboga, D. *An Idea Based on Honey Bee Swarm for Numerical Optimization*; Technical Report; Erciyes University, Engineering Faculty, Computer Engineering Department: Kayseri, Turkey, 2005.
113. Odili, J.B.; Kahar, M.N.M.; Anwar, S. African Buffalo Optimization: A Swarm-Intelligence Technique. *Procedia Comput. Sci.* **2015**, *76*, 443–448. [\[CrossRef\]](#)
114. Li, X.L. A New Intelligent Optimization-Artificial Fish Swarm Algorithm. Ph.D. Thesis, Zhejiang University, Hangzhou, China, 2003.
115. Mirjalili, S. The Ant Lion Optimizer. *Adv. Eng. Softw.* **2015**, *83*, 80–98. [\[CrossRef\]](#)
116. Yang, X.S. A New Metaheuristic Bat-Inspired Algorithm. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*; Springer: Berlin/Heidelberg, Germany, 2010; Volume 284, pp. 65–74.
117. Pham, D.; Ghanbarzadeh, A.; Koc, E.; Otri, S.; Rahim, S.; Zaidi, M. *The Bees Algorithm*. Technical Note; Technical Report; Manufacturing Engineering Centre, Cardiff University: Cardiff, UK, 2005.
118. Ayesh, A. Beaver algorithm for network security and optimization: Preliminary report. In Proceedings of the 2009 IEEE International Conference on Systems, Man and Cybernetics, San Antonio, TX, USA, 11–14 October 2009; pp. 3657–3662. [\[CrossRef\]](#)

119. Marinakis, Y.; Marinaki, M.; Matsatsinis, N. A Bumble Bees Mating Optimization Algorithm for Global Unconstrained Optimization Problems. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*; González, J.R., Pelta, D.A., Cruz, C., Terrazas, G., Krasnogor, N., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 305–318.
120. Teodorović, D.; Dell’Orco, M. Bee colony optimization—A cooperative learning approach to complex transportation problems. *Adv. OR AI Methods Transp.* **2005**, *51*, 60.
121. Niu, B.; Wang, H. Bacterial Colony Optimization. *Discret. Dyn. Nat. Soc.* **2012**, *2012*, 698057. [[CrossRef](#)]
122. Lu, X.; Zhou, Y. A Novel Global Convergence Algorithm: Bee Collecting Pollen Algorithm. In *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 518–525.
123. Passino, K. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Syst. Mag.* **2002**, *22*, 52–67. [[CrossRef](#)]
124. Wedde, H.F.; Farooq, M.; Zhang, Y. BeeHive: An Efficient Fault-Tolerant Routing Algorithm Inspired by Honey Bee Behavior. In *Ant Colony Optimization and Swarm Intelligence*; Dorigo, M., Birattari, M., Blum, C., Gambardella, L.M., Mondada, F., Stützle, T., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; pp. 83–94.
125. Bitam, S.; Mellouk, A. Bee life-based multi constraints multicast routing optimization for vehicular ad hoc networks. *J. Netw. Comput. Appl.* **2013**, *36*, 981–991. [[CrossRef](#)]
126. Askarzadeh, A.; Rezaeadeh, A. A new heuristic optimization algorithm for modeling of proton exchange membrane fuel cell: bird mating optimizer. *Int. J. Energy Res.* **2012**, *37*, 1196–1204. [[CrossRef](#)]
127. Sato, T.; Hagiwara, M. Bee System: finding solution by a concentrated search. In Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics, Orlando, FL, USA, 12–15 October 1997; Volume 4, pp. 3954–3959. [[CrossRef](#)]
128. Drias, H.; Sadeg, S.; Yahi, S. Cooperative Bees Swarm for Solving the Maximum Weighted Satisfiability Problem. In *International Work-Conference on Artificial Neural Networks (IWANN)*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 318–325.
129. Chu, S.C.; Tsai, P.w.; Pan, J.S. Cat Swarm Optimization. In *Pacific Rim International Conference on Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 854–858.
130. Yang, X.S.; Suash, D. Cuckoo Search via Lévy flights. In Proceedings of the 2009 IEEE World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, India, 9–11 December 2009; pp. 210–214. [[CrossRef](#)]
131. Meng, X.; Liu, Y.; Gao, X.; Zhang, H. A New Bio-inspired Algorithm: Chicken Swarm Optimization. In *Advances in Swarm Intelligence*; Springer: Cham, Switzerland, 2014; pp. 86–94.
132. Mirjalili, S. Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput. Appl.* **2015**, *27*, 1053–1073. [[CrossRef](#)]
133. Shiqin, Y.; Jianjun, J.; Guangxing, Y. A Dolphin Partner Optimization. In Proceedings of the 2009 IEEE WRI Global Congress on Intelligent Systems, Xiamen, China, 19–21 May 2009; pp. 124–128. [[CrossRef](#)]
134. Yang, X.S. Firefly Algorithm. In *Nature-Inspired Metaheuristic Algorithms*; Luniver Press: Beckington, UK, 2008; p. 128.
135. Mutazono, A.; Sugano, M.; Murata, M. Energy efficient self-organizing control for wireless sensor networks inspired by calling behavior of frogs. *Comput. Commun.* **2012**, *35*, 661–669. [[CrossRef](#)]
136. Pan, W.T. A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example. *Knowl.-Based Syst.* **2012**, *26*, 69–74. [[CrossRef](#)]
137. Yang, C.; Chen, J.; Tu, X. Algorithm of Fast Marriage in Honey Bees Optimization and Convergence Analysis. In Proceedings of the 2007 IEEE International Conference on Automation and Logistics, Jinan, China, 18–21 August 2007; pp. 1794–1799. [[CrossRef](#)]
138. Li, X.L.; Shao, Z.J.; Qian, J.X. An Optimizing Method based on Autonomous Animate: Fish Swarm Algorithm. *Syst. Eng. Theory Pract.* **2002**, *22*, 32–38.
139. Bastos Filho, C.J.A.; de Lima Neto, F.B.; Lins, A.J.C.C.; Nascimento, A.I.S.; Lima, M.P. A novel search algorithm based on fish school behavior. In Proceedings of the 2008 IEEE International Conference on Systems, Man and Cybernetics, Singapore, 12–15 October 2008; pp. 2646–2651. [[CrossRef](#)]

140. Min, H.; Wang, Z. Group escape behavior of multiple mobile robot system by mimicking fish schools. In Proceedings of the IEEE International Conference on Robotics and Biometrics (ROBIO), Tianjin, China, 14–18 December 2010; pp. 320–326.
141. Saremi, S.; Mirjalili, S.; Lewis, A. Grasshopper Optimisation Algorithm: Theory and application. *Adv. Eng. Softw.* **2017**, *105*, 30–47. [[CrossRef](#)]
142. Krishnanand, K.; Ghose, D. Detection of multiple source locations using a glowworm metaphor with applications to collective robotics. In Proceedings of the IEEE Swarm Intelligence Symposium, Pasadena, CA, USA, 8–10 June 2005; pp. 84–91.
143. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
144. Nakrani, S.; Tovey, C. On Honey Bees and Dynamic Allocation in an Internet Server Colony. In Proceedings of the 2nd International Workshop on The Mathematics and Algorithms of Social Insects, Atlanta, GA, USA, 15–17 December 2003; pp. 1–8.
145. Baig, A.R.; Rashid, M. Honey bee foraging algorithm for multimodal & dynamic optimization problems. In Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation (GECCO '07), London, UK, 7–11 July 2007; ACM Press: New York, NY, USA, 2007; p. 169.
146. Haddad, O.B.; Afshar, A.; Mariño, M.A. Honey-Bees Mating Optimization (HBMO) Algorithm: A New Heuristic Approach for Water Resources Optimization. *Water Resour. Manag.* **2006**, *20*, 661–680. [[CrossRef](#)]
147. Oftadeh, R.; Mahjoob, M.J. A new meta-heuristic optimization algorithm: Hunting Search. In Proceedings of the 2009 IEEE Fifth International Conference on Soft Computing, Computing with Words and Perceptions in System Analysis, Decision and Control, Famagusta, Cyprus, 2–4 September 2009; pp. 1–5. [[CrossRef](#)]
148. Gandomi, A.H.; Alavi, A.H. Krill herd: A new bio-inspired optimization algorithm. *Commun. Nonlinear Sci. Numer. Simul.* **2012**, *17*, 4831–4845. [[CrossRef](#)]
149. Abbass, H. MBO: marriage in honey bees optimization—a Haplometrosis polygynous swarming approach. In Proceedings of the 2001 IEEE Congress on Evolutionary Computation (IEEE Cat. No.01TH8546), Seoul, Korea, 27–30 May 2001; Volume 1, pp. 207–214. [[CrossRef](#)]
150. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl.-Based Syst.* **2015**, *89*, 228–249. [[CrossRef](#)]
151. Feng, X.; Lau, F.C.M.; Gao, D. A New Bio-inspired Approach to the Traveling Salesman Problem. In *International Conference on Complex Sciences*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 1310–1321.
152. Duman, E.; Uysal, M.; Alkaya, A.F. Migrating Birds Optimization: A new metaheuristic approach and its performance on quadratic assignment problem. *Inf. Sci.* **2012**, *217*, 65–77. [[CrossRef](#)]
153. Havens, T.C.; Spain, C.J.; Salmon, N.G.; Keller, J.M. Roach Infestation Optimization. In Proceedings of the 2008 IEEE Swarm Intelligence Symposium, St. Louis, MO, USA, 21–23 September 2008; pp. 1–7. [[CrossRef](#)]
154. Eusuff, M.M.; Lansey, K.E. Optimization of Water Distribution Network Design Using the Shuffled Frog Leaping Algorithm. *J. Water Resour. Plan. Manag.* **2003**, *129*, 210–225. [[CrossRef](#)]
155. Monismith, D.R.; Mayfield, B.E. Slime Mold as a model for numerical optimization. In Proceedings of the 2008 IEEE Swarm Intelligence Symposium, St. Louis, MO, USA, 21–23 September 2008; pp. 1–8. [[CrossRef](#)]
156. Hersovici, M.; Jacovi, M.; Maarek, Y.S.; Pelleg, D.; Shtalhaim, M.; Ur, S. The shark-search algorithm. An application: Tailored Web site mapping. *Comput. Netw. ISDN Syst.* **1998**, *30*, 317–326. [[CrossRef](#)]
157. Anandaraman, C.; Madurai Sankar, A.V.; Natarajan, R. A New Evolutionary Algorithm Based on Bacterial Evolution and Its Application for Scheduling A Flexible Manufacturing System. *J. Tek. Ind.* **2012**, *14*, 1–12. [[CrossRef](#)]
158. Roth, M.H. Termite: A Swarm Intelligent Routing Algorithm for Mobile Wireless Ad-Hoc Networks. Ph.D. Thesis, Cornell University, Ithaca, NY, USA, 2005.
159. Yang, X.S. Engineering Optimizations via Nature-Inspired Virtual Bee Algorithms. In *International Work-Conference on the Interplay Between Natural and Artificial Computation*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 317–323.
160. Liu, C.; Yan, X.; Liu, C.; Wu, H. The Wolf Colony Algorithm and Its Application. *Chin. J. Electron.* **2011**, *20*, 664–667. [[CrossRef](#)]
161. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [[CrossRef](#)]

162. Yang, C.; Tu, X.; Chen, J. Algorithm of Marriage in Honey Bees Optimization Based on the Wolf Pack Search. In Proceedings of the 2007 IEEE International Conference on Intelligent Pervasive Computing (IPC 2007), Jeju City, Korea, 11–13 October 2007; pp. 462–467. [[CrossRef](#)]
163. Theraulaz, G.; Goss, S.; Gervet, J.; Deneubourg, J.L. Task differentiation in Polistes wasp colonies: A model for self-organizing groups of robots. In Proceedings of the First international Conference on Simulation of Adaptive Behavior on from Animals to Animats, Paris, France, 24–28 September 1991; pp. 346–355.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).