A Survey on Swarm Intelligence Approaches for Feature Selection

Bach Hoai Nguyen*, Bing Xue, Mengjie Zhang

School of Engineering and Computer Science
Victoria University of Wellington, PO Box 600, Wellington 6140
New Zealand

Abstract

One of the problems in Big Data is a large number of features or dimensions, which causes difficulties to apply machine learning, especially classification algorithms. Feature selection is an important technique which selects small and informative feature subsets to improve the learning performance. However, feature selection is not an easy task due to its large and complex search space. Recently, swarm intelligence techniques have gained much attention from the feature selection community because of their simplicity and potential global search ability. There have been a few survey papers about applying swarm intelligence to feature selection. However, none of them provided a detailed and systematic discussion about how the search mechanisms (including both representation and updating mechanism) of swarm intelligence have been modified to solve feature selection problems effectively. This paper presents a comprehensive survey of the state-of-the-art work on swarm intelligence for feature selection. The main focus of this paper is on analyzing the search mechanism of the proposed algorithms. The expectation is to provide an overview to researchers and encourage them to investigate more effective search mechanisms for applying swarm intelligence to feature selection. At the end of the paper, several issues are presented for future research.

Keywords: Feature Selection, Swarm Intelligence, Particle Swarm Optimization, Ant Colony Optimization, Artificial Bee Colony, Classification

1. Introduction

In the last two decades, data has increased enormously in many fields including business, scientific research. The term "Big Data" has been widely used to describe a large amount of data that cannot be handled by the typical database software. Recently, Big Data is described by 5V characteristics: *volume, velocity, variety, value, veracity* [1]. *Volume* is probably the first and most common property when people talk about Big Data. The large *volume* of data can be caused by a large number of features (dimensionality) which is a challenging problem when applying machine learning, especially classification algorithms, to Big Data analysis.

Traditional machine learning usually does not work well on high-dimensional dataset due to the "curse of dimensionality" [2]. Notably, the increment in dimensionality enlarges the number of possible instances in the instance space, which makes the available data become sparse [3]. In order to achieve reliable results in such high-dimensional problems, classification algorithms require a large amount of data which usually grows exponentially with respect to the number of features. More importantly, not all features are useful. In contrast to relevant features which provide useful information about the learning task, irrelevant features provide misleading information leading to deterioration in the classification performance [4]. For example, in k-nearest neighbor (KNN), the irrelevant or noisy features may increase the distances between instances from the same class, which makes KNN more difficult to classify instances correctly. In some other classification algorithms such as decision trees (DT) or support vector machines (SVMs), the learned model may have to overfit the irrelevant features to cope with the data; in which case it will not work well on unseen/future instances. Redundant features provide the same or similar information about the learning task as other features. In respect of classification algorithms which directly use training instances in the classification process such as Naive Bayes or KNN, redundant features add unnecessary weights which can reduce the classification performance. For classification algorithms which explicitly build a classification model such as DT or SVM, the redundant features can be removed during the training process. However, the redundant features cause extra complexity which increases the training time.

To deal with high-dimensional datasets, feature selection [5] is proposed to reduce the number of features by removing irrelevant and redundant features. Feature selection has been used to improve many machine learning

*Corresponding author.
Email addresses: Hoai.Bach.Nguyen@ecs.vuw.ac.nz (Bach Hoai Nguyen), Bing.Xue@ecs.vuw.ac.nz (Bing Xue), Mengjie.Zhang@ecs.vuw.ac.nz (Mengjie Zhang)

tasks including classification [5], clustering [6, 7], and regression [8]. However, most studies apply feature selection to classification problems, so the focus of this paper is to review feature selection for classification. The benefits of feature selection include improving the learning performance, saving the cost of measuring unused features, and making the learned classifier simpler and easier to understand. However, feature selection is a challenging task due to its large search space. Suppose the number of original features is n, the total number of possible feature subsets is 2^n which increases exponentially with respect to the number of features. Hence, feature selection is an NP-hard problem [9] which makes an exhaustive search impractical. In order to achieve feature selection, it is necessary to have an efficient global search technique. Evolutionary computation (EC) is a family of population-based optimization techniques which have a potential global search ability. Swarm intelligence (SI) is a branch of EC which consists of algorithms inspired by behaviors of social animal/insect. Some well-known representatives of SI are particle swarm optimization (PSO), ant colony optimization (ACO), and artificial bee colony optimization (ABC). SI has been widely applied to feature selection because of its simplicity, effective search mechanism, and natural representation [10]. This paper presents a comprehensive survey with an expectation to provide a state-of-the-art overview of SI based feature selection algorithms.

There are a small number of papers reviewing SI based feature selection algorithms. Kothari et al. [11] discussed 17 papers applying PSO — one of the most popular SI algorithms — to feature selection. However, all the reviewed papers were published before 2010. Bin Basir and Binti Ahmad [12] presented a short review where there were about 30 papers published before 2013. Xue et al. [13] provided a comprehensive survey of EC based feature selection. However, due to the scope of the paper [13], only PSO and ACO based algorithms were reviewed in details. Recently, Brezočnik et al. [14] presented a broad survey about SI based feature selection algorithms. The paper [14] reviewed papers according to different components of the algorithms such as representation, initialization, updating mechanism. However, representation and updating mechanism are two interconnected components, so separating them may not give a good overview of how an SI algorithm works. This paper provides an overview of SI based feature selection algorithms from a different perspective. We category reviewed papers based on how feature selected is presented in the proposed algorithms. With respect to the representation, we discuss how the proposed algorithms perform feature selection.

The remainder of this paper is organized as follows. Section II describes the background of feature selection. Section III reviews SI based feature selection algorithms which are mainly based on PSO, ABC, and ACO. Section IV discussed current issues and future directions. The paper is concluded by Conclusions in Section VII.

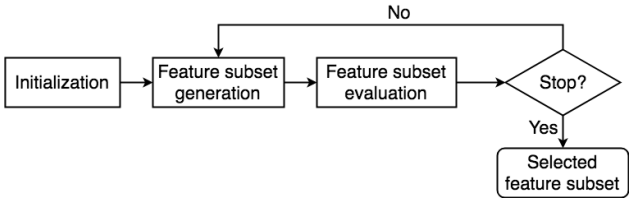

Figure 1: An overall feature selection process [15].

2. Background

Feature selection is a process of selecting a small and more informative feature subset from the original features. For a classification task, feature selection aims to find the smallest feature subset which is necessary and sufficient to describe the class label [16]. In general, there are four main steps in a feature selection algorithm, which can be seen in Figure 1. Among the four steps, "Subset Generation" and "Subset Evaluation" are the two most important steps. "Subset Generation" generates candidate feature subsets using a search mechanism. The goodness of candidate subsets is measured by an evaluation function (fitness function) in "Subset Evaluation". Based on feedback from "Subset Evaluation", "Subset Generation" is expected to generate more promising feature subsets.

According to the evaluation criteria, feature selection approaches can be divided into three categories: wrapper approaches, embedded approaches, and filter approaches [17]. Wrapper approaches evaluate candidate subsets by a classification algorithm. Embedded approaches also involve a classification algorithm, but features are selected during the training process of the algorithm. An example is DT which selects features as its internal nodes during its training process. The features that appeared in the final tree are the selected features. In contrast, filter approaches do not use any classification algorithm to evaluate candidate subsets. The evaluation is mainly based on intrinsic characteristics of a dataset. Among the three approaches, the filter one is usually the most efficient since it does not involve any learning process. However, wrapper and embedded approaches usually achieve better classification performance since they consider the interaction between the selected features and the wrapped classification algorithm. Embedded approaches are usually less computationally intensive than wrapper approaches, but they are only applicable to some specific classification algorithms.

Feature selection is a challenging problem due to its large search space and complex interactions between features [17]. On the one hand, the number of possible feature subsets increases exponentially with respect to the number of original features. On the other hand, the feature interaction significantly affects the classification performance. For example, two weakly relevant features may significantly improve classification performance when they are selected together. In contrast, selecting highly relevant features may result in many redundant features. There-

fore, it is essential to generate and evaluate feature subsets as a whole, which are responsibilities of the search mechanism and the evaluation criteria, respectively. In addition, feature selection can be considered a multi-objective problem since its two main objectives, maximizing the classification performance and minimizing the number of selected features, are usually in conflict. As a family of population-based optimization techniques, EC has been widely applied to feature selection because of its potential global search ability and natural mechanism to evolve a set of trade-off solutions for multi-objective problems. This paper focuses on reviewing feature selection algorithms based on SI which is a branch of EC. SI is gaining more attention by the feature selection community recently [13].

2.1. Existing Work on Feature Selection

In this subsection, we briefly discuss the existing feature selection algorithms based on two aspects: the evaluation criteria and the search mechanism.

2.1.1. Evaluation criteria

In a wrapper feature selection algorithm, feature subsets are evaluated by the classification performance. Most classification algorithms can be applied to feature selection, for example KNN [18, 19], Naive Bayes (NB) [20, 21], SVMs [22, 23, 24, 25], and artificial neural networks (ANNs) [26, 27, 28].

Filter approaches apply measures from different disciplines to evaluate feature subsets. The four most common filter measures in feature selection are distance measures, correlation measures, consistency measures, and information measures. The distance measures aim to select a feature subset that can separate instances from different classes as much as possible. A well-known representative of distance measures is Relief [29]. Consistency measures [30] show how consistent the selected features can separate different classes in comparison with using all features. Two instances are inconsistent if they have the same feature values, but they belong to different classes. The goal is to find a minimal feature subset that reaches an acceptable inconsistency rate. Correlation measures [31] evaluate how dependent two random variables are, so they can be used to select a feature subset which is highly related to the class label (maximizing relevance) and contains uncorrelated features (minimizing redundancy). Similar to correlation measures, information measures [32] can be used to measure the redundancy and relevance of a feature subset. Among the four measures, information measures usually gain more attention because they can detect non-linear relationships between random variables.

In an embedded feature selection algorithm, the selection process happens during the training process of a classification algorithm. DT is an example of embedded feature selection algorithms, where the features appeared in the final tree are the selected features. SVMs can also be considered an embedded approach where its model weights can reflect how important the corresponding features are. Recently, an embedded feature selection approach based on a sparse-learning mechanism is gaining more attention because of its good performance [33, 34]. The idea is to include a sparse regularization term in the objective function so that when the classification error is minimized, many feature coefficients are forced to be very small, or exactly zero. Feature selection is achieved by selecting features with large enough coefficients [35, 36].

2.1.2. Search mechanisms

The most straightforward mechanism is to consider all the possible feature subsets, which is known as an exhaustive search [37, 38, 39]. The exhaustive search guarantees to find an optimal feature subset but it is impractical due to its extremely high computational cost, especially when there are a large number of selected features. In order to reduce the computation cost, two greedy search mechanisms, sequential forward selection (SFS) [40] and sequential backward selection (SBS) [41], were proposed. While SFS started from an empty subset and sequentially added features until the classification could not be improved or a predefined number of selected features was reached, SBS started from a full set of features and sequentially removed features. Although SFS and SBS significantly reduced the computational time, they suffered from the "nesting effect" where the added/removed features could not be removed/added later. The problem was addressed by Stearns et al. [42] who applied l times forward steps followed by r times backward steps. However, l and r were problem-specific, so Pudil et al.[43] proposed two floating sequential searches, sequential forward floating search (SFFS) and sequential backward floating search (SBFS) which could dynamically determine the values (l, r).

Although sequential searches have been widely applied to feature selection [44, 45, 46, 47], they are usually stuck at local optima due to their sequential behavior. In recent years, EC techniques gain more attention from feature selection community since they do not make any assumptions about the search space. More importantly, they are population-based techniques, so they can produce multiple solutions in a single run which is suitable for multi-objective feature selection. In general, EC can be divided into two main categories: evolutionary algorithms (EAs) and swarm intelligence (SI). EAs refers to the evolutionary algorithms which follow Darwinian principles. In particular, these algorithms apply genetic operators such as mutation, crossover, reproduction, and selection to evolve a population of individuals. The individuals compete to survive based on their fitness values. Among EAs, genetic algorithms (GAs) [48] is probably the most common technique applying to feature selection [13]. SI algorithms are inspired by the behaviors of social insects/animals. In these algorithms, a population consisting of a set of individuals explore and share their knowledge about the search space to other members. The sharing mechanism

helps the whole swarm move toward better positions in the search space, which eventually converges to an optimum [49]. Some well-known representatives of SI algorithms are PSO [50], ACO [51], and ABC [52]. In comparison with GAs, SI algorithms usually converge faster and perform relatively better when the computational budget is low [53, 54], which might be a reason for a significant increase in the number of papers using SI algorithms for feature selection, especially PSO [13]. Therefore, the focus of this paper is SI based feature selection algorithms.

2.2. Detailed Coverage of This Paper

Recently, many SI algorithms have been applied to feature selection, and the three most popular ones are discussed in this paper, which are PSO, ABC, and ACO. For each algorithm, we realize that there are two main ways to represent feature selection which are a standard representation — the initial representation when the algorithm was proposed, and a binary representation — a representation tailored for feature selection. Besides, the updating mechanism depends heavily on the representation. Therefore, we divide SI based feature selection algorithms into two main categories corresponding to the above two representations. For each representation type, the algorithms are further classified as single objective algorithms and multi-objective algorithms since the search mechanisms are different when the algorithms have to deal with different numbers of objectives.

The reviewed literature is organized as follows. The feature selection algorithms based on PSO, ABC, and ACO are reviewed in Section 3. Each subsection in Section 3 discusses a particular SI technique for feature selection. In each subsection, the algorithms are further divided according to their representations for feature selection.

3. Swarm Intelligence for Feature Selection

3.1. PSO for Feature Selection

In PSO, each position of a candidate solution is represented by a vector which is a natural representation for feature selection. Specifically, each element of the vector corresponds to an original feature, and its value indicates whether the corresponding feature is selected or not. PSO has been applied to achieve feature selection in many real-world applications such as text mining [55, 56, 57, 58], data stream [59], image analysis [60, 61], medical problems [62, 63, 64]. A standard (continuous) PSO representation consists of real-value elements. If the element value is greater than a threshold θ, the corresponding feature is selected. Otherwise, the feature is discarded. A binary PSO representation consists of binary-value elements. If the element value is "1", the corresponding feature is selected. Otherwise, the feature is discarded. The following subsections discuss PSO based feature selection algorithms based on their representations.

3.1.1. Standard (continuous) representation

There have been many studies proposed to improve the performance of PSO based feature selection algorithms. The modifications have been made in the representation, initialization and search mechanism. In terms of representation, Lin et al. [65] proposed a representation so that PSO could perform both feature selection and optimizing SVMs kernel parameters simultaneously. The proposed representation consisted of the standard representation and additional elements for SVM's parameters. The results showed that the proposed algorithm achieved better classification performance than using all features and a similar work based on GAs [66]. The same idea was applied to perform feature selection for power distribution systems [67]. Tran et al. [68] proposed a representation that could achieve both feature selection and feature discretization. In the proposed representation, each element in the position vector was used as a cutting point to discretize an original feature. If the element value was out of a predefined range, the corresponding feature was discarded. The proposed representation assisted PSO to select a smaller number of features and achieve better classification performance than using the standard representation.

The length of a standard representation is equal to the number of original features, which results in a large search space and an expensive computation cost. Many works have been proposed to shorten the representation length. Nguyen et al. [69] firstly grouped similar (potentially redundant) features into the same group or cluster. A maximum number of features selected from each cluster was predefined. Each position element was a real-value number indicating a feature index from a cluster. Since the maximum number for each cluster was smaller than the number of features in the cluster, the proposed representation was shorter than the standard representation. However, in the proposed representation a small change of the position might not lead to any different feature subset. Therefore, Nguyen et al. [70] applied Gaussian distribution to propose a new transformation rule, which could form a smoother fitness landscape than the representation in [69]. The most important question was how to determine the maximum number of features selected from each feature cluster. Tran et al. [71] proposed a variable length representation for PSO. The main idea was to divide the swarm into multiple divisions, and each division had its maximum length. The original features were firstly ranked by symmetric uncertainty measure [72]. If the maximum length of a division was 100, only the top 100 features were considered to be selected. Therefore, although the proposed representation could significantly reduce the computation time, it still allowed the swarm to explore different feature subset sizes ranging from 1 to the total number of original features.

Shortening the particle dimension is an option to improve the efficiency of PSO based feature selection algorithms. However, the most time-consuming part is usu-

ally the evaluation process, especially in wrapper PSO based feature selection. In order to significantly reduce the computational cost, it is necessary to investigate an efficient evaluation. Wang and Liang [73] split a training set into many subsets. Feature selection was performed on each training subset, which resulted in several feature subsets. The final subset was formed by combining the obtained feature subsets. Since each training subset was much smaller than the original training set, the evaluation process was more efficient. However, it is possible that features selected from different subsets might be redundant. Nguyen et al. [74] also modified the training set to speed up the evaluation process. Mainly, for the first 70% iterations, candidate feature subsets were evaluated by a surrogate training set containing several instances selected from the original training set. The original training set was used only in the last 30% iterations. The surrogate training set significantly reduced the computation time while achieving similar or better classification accuracy than using the original training set. The work was further extended in [75] where the surrogate training set was dynamically determined during the evolutionary process. Butler-Yeoman et al. [76] proposed a hybrid approach combining filter and wrapper to reduce the evaluation cost. The idea was to use mutual information to estimate promising candidates which were then compared with *pbest* using a classification algorithm. Since not all candidates were evaluated by the classification performance, the proposed algorithm was less computationally expensive than the standard wrapper PSO algorithm.

Initialization is an essential step in PSO. A good starting point usually leads the swarm to a better solution. In [77], Xue et al. proposed three new initialization mechanism inspired by the sequential feature selection approach. The three mechanisms were different in terms of the number of features using for initialization. The small initialization allowed each particle to start with 10% of the total number of features. In contrast, the large initialization allowed each particle to start with 50% of the total number of features. The mixed initialization combined both small and large strategies where 2/3 of the swarm was initialized by the small strategy and the remaining 1/3 of the swarm was initialized by the large strategy. The results showed that the small initialization selected the smallest number of features while the large initialization selected the largest number of features. The mixed initialization selected a smaller number of features with similar or better accuracy than the random initialization.

The key idea of PSO is to improve the candidate solutions by experience learning from *pbest* and *gbest* which usually makes PSO converge quickly. However, when applying to a problem with a large and complex search space such as feature selection, PSO usually suffers the premature convergence problem in which the swarm is stuck at a poor solution. Many studies have integrated local searches in PSO based feature selection to improve its performance. Tran et al. [78] proposed a local search method that flipped a small number of selected elements in *pbest*. If the obtained feature subset was better than the current *pbest*, it replaced the *pbest*. The proposed algorithm could select a smaller number of features with a better classification performance than standard PSO. Later, Tran et al. [79] proposed another local search mechanism to improve *pbest*. The local search firstly removed features that were more relevant to other features than the class label, i.e. removing potentially redundant features. After that, the local search added features that were more relevant to the class label than the other features, i.e. adding potentially relevant features. The classification performance was improved over the one proposed in [78]. Nguyen et al. [80] focused on improving *gbest* by a local search mimicked the backward feature selection method. The idea was to remove features from *gbest* according to the relevant and redundant measure calculated by mutual information. Although the proposed algorithm spent an additional computation cost for the local search, it was still faster than other PSO based feature selection algorithms since it selected much smaller numbers of features while maintaining or even improving the classification performance. Mistry et al. [81] improved the swarm diversity by using the average of all discovered *pbest* instead of the best discovered *pbest* as in standard PSO. Additional random values were added to *gbest* before updating a particle so that the particle moved farther from the current position. The proposed algorithm also embedded the idea of a micro-genetic algorithm (mGA) to improve the swarm diversity by using a small secondary swarm. The secondary swarm was iteratively formed by selecting particles that had either the lowest or the highest correlation with *gbest*, so the proposed algorithm could balance between local exploitation and global exploration. The experimental results showed that the proposed algorithm outperformed conventional GAs and PSO based feature selection algorithms on facial emotion recognition. Also inspired by GAs, Nguyen et al. [82] proposed a local search mechanism based on crossover and mutation operators from GAs. The crossover was performed between a pair of particles, called parents. The obtained children replaced their parents if they had better fitness values. The mutation was applied to improve *gbest* once *gbest* had not been improved for a finite number of iterations. The position element not only indicated whether a feature was selected but also presented how much confident the selection decision was. The less confident element was more likely to be mutated than the more confident element. The results showed that the two operators assisted PSO to achieve better fitness values than standard PSO during an entire evolutionary process. Recently, Gu et al. [83] applied Competitive Swarm Optimizer (CSO) —a modified version of PSO— to achieve feature selection. CSO was proposed by Cheng et al. [84]. In CSO, *gbest* and *pbest* were removed. The particles had to enter a competition, and the winners went directly to the new population. The losers had to learn from the winners, i.e. their positions were updated with respect to the positions

of the winners, and then they could enter the new population. The results showed that the CSO based feature selection algorithm selected a smaller number of features with a lower classification error rate than standard PSO.

Dynamic parameter control, which can balance between exploration and exploitation, is also a good way to avoid premature convergence. In standard PSO, decreasing the inertia weight usually assists PSO to achieve better performance since the search process smoothly changes its focus from exploration to exploitation [85, 86]. The strategy was also applied to feature selection [55, 60]. Adeli and Broumandnia [87] proposed to control the inertia weight based on swarm diversity. When the diversity was low, the inertia weight was increased to encourage exploration, and vice versa when the diversity was high. The proposed algorithm evolved better feature subsets than Chaotic PSO [88] and Random PSO [89] which also controlled the inertia weight dynamically.

In multi-objective PSO (MOPSO), since there is not a best solution, MOPSO usually maintains an archive set containing all non-dominated solutions discovered so far. When updating a particle, an archive member can be selected as *gbest*. Xue et al. [90] proposed the first multi-objective PSO (MOPSO) algorithm for feature selection. The target was to minimize both the classification error and the number of selected features using either continuous or binary PSO. The experimental results showed that the proposed algorithm, called CMDPSOFS, was superior to NSGA-II [91] and SPEA2 [92] on feature selection problems. A filter MOPSO based feature selection was also proposed by Xue et al. [93], in which the two objectives were to minimize the number of features and to maximize the relevance between the selected features and the class labels. The relevance was calculated by using either mutual information or information gain. The experimental results suggested that MOPSO algorithms evolve feature subsets with higher classification performance than single-objective feature selection algorithms. Later, Nguyen et al. [94] improved the archive members in MOPSO by applying three local search operators, which are Inserting, Removing and Swapping. The inserting operator aimed to add at most one feature that could improve the archive member. The obtained solution was passed to the removing operator that removed at most one feature so that the solution was improved. Finally, the swapping operator improved the generated solution by replacing a selected feature by an unselected feature. As a result, the three operators improved the archive members quality by changing a few features in their feature subsets. The proposed algorithm could select a smaller number of features and achieved similar or better classification performance than NSGA-II [91], SPEA2 [92], and CMDPSOFS [90] on 12 UCI datasets. Maintaining diversity is essential in any evolutionary multi-objective algorithms, including MOPSO. Zhang et al. [95] increased the swarm diversity in MOPSO by re-initializing velocities and partially mutating positions of some particles. Amoozegar and Minaei-Bidgoli [96] improved the swarm diversity by applying uniform and non-uniform operators to two sub-swarms selected from the whole swarm. The proposed algorithm also generated new candidate feature subsets based on how frequent features were selected by the archive members (non-dominated solutions). The generated solution could be added to the archive set if it was not dominated by any archive members. The proposed algorithm evolved more diverse approximated Pareto fronts than CMDPSOFS [90].

3.1.2. Binary representation

PSO was originally proposed for continuous optimization. A straight forward way to extend PSO for solving binary optimization is to keep using the continuous updating equation and convert the continuous position to a binary position. A sigmoid function is widely used for this task since it can convert any continuous value to a continuous value in the range [0,1]. A random value between [0,1] is used to convert the obtained continuous value to a binary value. The above approach has been applied to achieve feature selection [97, 98, 99, 100, 101, 102]. However, due to the standard updating equations, the above approach also suffers the premature convergence as in continuous PSO. Chuang et al. [103] proposed a *gbest* resetting mechanism, which set all *gbest* position's elements to zero when the best fitness did not change for a finite number iterations. The experimental results showed that the resetting mechanism helped PSO to evolve a smaller set of features with higher classification accuracy than standard binary PSO [104] in most cases. The *gbest* resetting approach was also applied in [105] where the new *gbest* was determined based on all the *pbest* discovered so far. Moradi and Gholampour [106] proposed a local search to avoid the premature convergence. The original features were divided into dissimilar and similar sets based on their correlations with other features. The numbers of features selected from the two sets were predefined. For each candidate subset, the local search was performed to add or remove features so that the number of features selected from each set matches the predefined number of features. Chen et al. [107] added a new term called feature confidence to the sigmoid function. The confidence of a feature had two main components: the feature relevance measured by Relief and the feature frequency measured by the number of times the feature was selected by *gbest* (during the evolutionary process). Dong et al. [108] avoided the premature convergence by preventing all particles from communicating with each other. The swarm was divided into k niches. The particles in the same niche could share their knowledge. The niches communicated through their centers. The proposed topology was expected to slow down the information exchanging.

Similar to the continuous representation, binary representation was also modified to achieve both feature selection and parameter optimization for SVMs [109, 110]. Statistical feature clustering was also utilized in the binary representation which allowed to select one or multiple fea-

tures from each cluster [111, 112]. The proposed representation significantly reduced the number of selected features while maintaining or improving classification performance.

However, in a binary search space, there is no direction, and particles move by flipping their bits. Therefore, using velocity and momentum concepts from continuous PSO is not appropriate. For example, Liu et al. [113] showed that in binary PSO, increasing the inertia weight made the swarm change its focus from exploration to exploitation, which was opposite to continuous PSO. Besides, picking an appropriate variant of the sigmoid function is an essential but difficult task [114]. To avoid the above limitations, Nguyen et al. [115] proposed a novel binary PSO algorithm where the two concepts were redefined. In the proposed algorithm, the velocity was defined as the probability of flipping bits in the position. The momentum was defined as the tendency to stick with the current position. The proposed binary PSO algorithm could select better feature subsets with a higher classification performance than the standard binary PSO algorithm since it describes binary movements more accurately.

3.1.3. Discussion about PSO based feature selection

In comparison with continuous PSO, there are much fewer studies applying binary PSO to feature selection. In fact, a binary representation is more suitable to feature selection than a continuous representation. One feature subset can be represented by exactly one binary vector. Meanwhile, one feature subset can be represented by many (possibly infinite) numbers of continuous vectors. Therefore, using the continuous representation significantly enlarges the search space. However, Xue et al. [93] showed that continuous PSO selected better feature subsets than standard binary PSO. The limited performance of binary PSO is due to its inappropriate application of the velocity and momentum concepts from continuous PSO. The work in [115] showed that if the properties of a binary search space are considered, the performance of binary PSO is significantly improved. However, [115] is just a very initial work. More investigation and analysis such as parameter control, search behavior are needed to improve the performance of binary PSO for feature selection further.

PSO has a natural representation for feature selection where each position bit corresponds to an original feature. However, this representation does not scale well when it is applied to select from thousands or even millions of features. In addition, the standard representation can show which features are selected, but it can not show the interactions between features, i.e. which features are working well with the others. More efficient and meaning full representation is still an open issue in PSO based feature selection.

3.2. ABC for Feature Selection

In 2005, Kraboga [116] proposed a bee swarm algorithm called artificial bee colony (ABC) initially for numerical optimization. ABC represents each candidate solution as a food source. In a bee swarm, there are three kinds of bees including employed bees, onlooker bees, and scout bees. Each food source has its own employed bee which tries to improve the food source's quality by searching around the neighboring food sources. Onlooker bees work similar to employed bees, except they are more likely to search around high-quality food sources. If a food source is not improved for a number of iterations, the corresponding employed bee becomes a scout bee which selects a new food source randomly.

ABC utilizes a vector-based representation to solve the optimization task, which is a natural representation for feature selection. Similar to PSO, ABC can represent feature selection by either standard (continuous) or binary representations. In the next subsections, ABC based feature selection algorithms will be reviewed according to their representations.

3.2.1. Standard (continuous) representation

Since ABC was originally proposed for continuous optimization problems, its standard representation is a vector of real numbers which can be called a continuous representation. The continuous vector can represent a feature selection problem where each element corresponds to an original feature. A threshold θ is used to determine whether a feature is selected or not. Particularly, if the element is greater than θ, the corresponding feature is selected. Otherwise, the feature is not selected. This representation scheme was used in one of the early ABC based feature selection algorithms to select features for a keystroke problem [117]. The aim was to select features that could help to identify a user based on his key stroke's properties. The proposed algorithm was a wrapper approach which utilized ANNs to evaluate the candidate feature subsets. By performing feature selection, the classification accuracy could be up to 95%. Continuous ABC was also used to select relevant features for medical tasks in [118, 119, 19] where SVMs was the wrapped classification algorithm. However, the above methods do not consider the number of selected features which is one of the main objectives of feature selection. In 2016, Wang et al. [24] included the feature ratio (the number of selected features divided by the total number of features) in the fitness function. The authors also proposed an initialization mechanism based on integer tent maps that ensured each bit of a position vector has a unique value in the range [0,1]. The experimental results showed that the number of features was significantly reduced and the proposed initialization mechanism achieved better classification performance than the random initialization mechanism.

There are several works attempted to modify the standard representation in ABC based feature selection. Kuo et al. [120] slightly modified the representation by including SVMs' parameters in each candidate solution. The representation allowed ABC not only to select the relevant features but also to optimize the SVMs' parameters.

Same idea was proposed by Alshamlan et al. [121], except the number of selected was predefined. Rakshit et al. [122] proposed a modified representation for a filter feature selection approach. The fitness function was based on clustering distances where the aim was to reduce the distance between instances from the same cluster (cohesion) and increase the distance between instances from different clusters (separation). The modified representation had two parts. The first part had a predefined number of integer values which were the indices of the selected features. The second part was the positions of C cluster centers where C was the number of classes. The proposed algorithm achieved better performance than using all features, but it required to define the number of selected features which was not an easy task. In addition, the above modifications increased the representation's length, so the search space was also significantly enlarged.

ABC was also combined with other algorithms to improve selection performance. Alshamlan et al. [123] proposed one of the early works applying ABC to gene expression problems. Due to a huge number of genes, mRMR [124] was utilized to reduce the number of features by removing some irrelevant and redundant features. Among the remaining features, ABC selected the informative feature subsets which are evaluated by SVMs. The experimental results showed that using mRMR as a preprocessing step could significantly improve the classification performance. Shunmugapriya and Kanmani [125] combined ACO and ABC to avoid the stagnation problem in ACO and the delayed convergence in ABC. Firstly, ACO generated candidate solutions based on its pheromone values. The generated solutions were then treated as food sources which were further improved by three kinds of bees in ABC. Finally, the ACO's pheromone was updated by the best food source generated by ABC. The process was repeated until a predefined number of iterations is reached. The proposed hybrid algorithm achieved better performance than both standard ABC and ACO.

The standard continuous representation has also been widely used in multi-objective ABC based feature selection algorithms. Ghanem and Jantan [126] proposed a wrapper multi-objective ABC based algorithm which considered the number of selected features and the classification performance measured by ANN. The proposed algorithm maintained an archive set containing all the non-dominated solutions discovered so far. The employed bees generated neighboring solutions by selecting a non-dominated archive member randomly. Hancer et al. [127] proposed a filter multi-objective ABC-based algorithm (MOABC) where the two objectives were to reduce the redundancy and increase the relevance. Both objectives were measured by mutual information. The proposed algorithm was inspired by NSGA-II where the candidate solutions were ranked according to their non-dominated ranks and crowding distances. Instead of using the standard operators, the proposed ABC algorithm utilized GA's crossover and mutation operators to generate new candidate solutions. Experiment results showed that MOABC achieved better classification performance than single objective ABC.

3.2.2. Binary representation

In the binary representation, each feature is represented by a binary value where "1" means the corresponding feature is selected and "0" means the corresponding feature is not selected. Since binary representation is not a standard representation of ABC, it is necessary to develop a corresponding updating mechanism. The most straight forward way is to keep using the standard continuous updating mechanism. The binary representation can be obtained by applying a sigmoid function to convert from a continuous vector to a binary vector, which is similar to standard binary PSO. The idea was showed to be effective in feature selection [128, 129, 130]. Yavuz and Aydin [131] used an angle modulated to convert from binary optimization to continuous optimization. Particularly, there are four parameters used in a *sine* function to generate continuous values. Each generated value corresponded to an original feature. If the value is greater than 0, the corresponding feature is selected. Otherwise, the corresponding feature is discarded. Although the proposed algorithm significantly reduced the dimensionality of the search space — from the number of original features to four continuous variables —, it also introduced an unnecessary relationship between the binary values since they were generated from the same distribution. In the above-proposed algorithms, although the candidate solutions were binary vectors, the search procedure was essentially performed on a continuous search space.

Later, Schiezaro and Pedrini [132] replaced the standard continuous updating mechanism in ABC by the GA mutation operator which was more natural for a binary representation. The proposed algorithm achieved significantly better performance than PSO, GAs, and ACO on 10 UCI datasets. However, the proposed algorithm was not compared with standard binary ABC that used a sigmoid function. GA crossover and mutation were also applied to multi-objective ABC [133]. The experimental results showed that binary multi-objective ABC generated a better Pareto front than the continuous one. Hancer et al. [134] substantially modified the updating mechanism using the Jaccard similarity coefficient. Firstly, for each candidate solution, instead of selecting only one other solution, two other solutions were selected. A binary mutant vector was built by an integer model programming so that the Jaccard dissimilarity between the current solution and the mutant vector was similar to the Jaccard dissimilarity between the two selected solutions. The new candidate solution was obtained by performing a crossover between the current solution and the mutant vector. The proposed algorithm selected better feature subsets than other binary optimization algorithms including GA, binary PSO, binary ABC, and ACO. Ozger et al. [135] performed a comparative study on different binary

ABC algorithms on feature selection. The compared algorithms included BitABC [136] using bitwise operators such as AND, OR, XOR to generate new candidate solutions, and binary ABC algorithms using different functions to convert continuous vector to binary vectors, for example, rounding function [137], sigmoid function [138], tangent function [139]. The experimental results showed that BitABC generated better feature subsets in a shorter computational time. Thus, designing an updating mechanism suitable to the binary representation usually results in better performance. Bitwise operators were also used in both single-objective ABC based feature selection [140] and multi-objective ABC based feature selection [141].

3.2.3. Discussion about ABC based feature selection algorithms

In comparison with PSO, there are fewer studies applying ABC to achieve feature selection. It is probably because ABC was proposed later than PSO. An advantage of ABC over other swarm intelligence algorithms is the clear separation between exploration and exploitation. Particularly, the employed bees and the onlooker bees look at neighboring solutions — perform exploitation, while the scout bees look at a new random solution — perform exploration. However, many studies combine ABC with other optimization algorithms such as DE [140], ACO [125], PSO [126] to balance between exploration and exploitation. Although the proposed hybrid algorithms achieve promising results, they also have more parameters than a single algorithm, i.e. parameters from both algorithms and parameters to control the algorithm combination. Given the clear separation between exploration and exploitation of ABC, it would be better to investigate a better mechanism to control exploration and exploitation for ABC rather than mixing it with other algorithms.

There are also a small number of multi-objective ABC feature selection algorithms. A possible reason is the onlooker bees select a food source based on the food source quality which is a single value. However, in multi-objective feature selection, each food source has at least two objective values, so it is difficult for the onlooker bees to select which food source to be improved. A further investigation of multi-objective ABC feature selection is still an open issue.

3.3. ACO for Feature Selection

ACO is one of the most well-known and widely used swarm intelligence algorithm proposed by Dorigo and Di Cario in 1999 [142]. Originally, ACO was designed to solve discrete optimization problems which have many states denoted by nodes. Each ant could move along edges connecting adjacent nodes to build a full solution to the optimization problem gradually. A fitness function then evaluated the solution. The goodness of the solution was utilized to update information (called "pheromone") on each edge that the solution used. Specifically, a good solution increases the amount of pheromone on its paths.

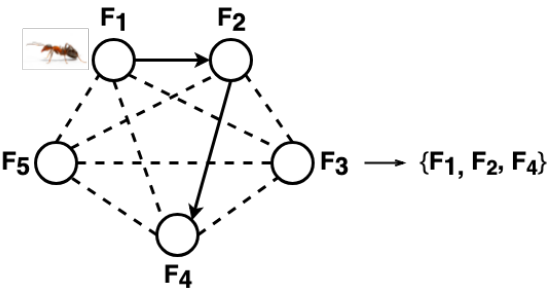

Figure 2: Standard fully-connected graph representation: each node is a feature, an ant traverses a node to select the corresponding feature. In this example, $\{F_1, F_2, F_4\}$ is the selected feature subset.

The following ants were attracted more by paths with higher amounts of pheromone, and they were expected to build more promising solutions. ACO has been applied to many real-world problems [143], for example routing [144], scheduling [145], DNA sequencing [146].

Since feature selection is a discrete (combinatorial) optimization problem, it has been widely achieved by ACO. In the following subsections, we firstly describe the overall view of an ACO based feature selection algorithm. The first and most important question is how to represent feature selection in an ACO algorithm. Based on our literature reviews, there are two main approaches which we called standard graph representation and binary graph representation. In the standard graph representation, each original feature is represented by a node in a graph. There are also edges connecting between nodes that an ant can follow to build its path. The nodes or features appearing in the path are features selected by the ant. This representation is also a standard way for ACO to represent other discrete problems. The second representation, binary graph representation, is more specific to feature selection. Feature selection can be formulated as a set of binary decisions where each decision determines whether a feature is selected or not. Therefore, the binary graph representation uses two sub-nodes: sub-node 0 and sub-node 1 to represent each feature. An ant visits all the nodes (features), and at each node, the ant visits one of the two sub-nodes. If the ant visits the sub-node 1, the corresponding feature is selected. Otherwise, the corresponding feature is discarded. The two following subsections reviews and discuss ACO based feature selection algorithms that use the standard and binary graph representations, respectively.

3.3.1. Standard graph representation

Firstly, we would like to describe an overall view of a standard ACO based feature selection algorithm. As discussed above, in standard representation, each feature is a node in a graph. An edge connecting between two features shows that one of the two features may be the next selected feature if the other feature is selected, which can be seen in Figure 2. In each iteration, a number of ants (defined by population size) simultaneously build their feature subsets. The process that each ant builds its feature

subset is described as bellows:

- Step 1: an ant starts with a random node — a random feature.
- Step 2: suppose that currently the ant select is at node (feature) i, the ant will select the next node (feature) to visit according to the following probability:

$$p_{ij} = \begin{cases} \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{k \in J_i}[\tau_{ik}]^{\alpha}[\eta_{ik}]^{\beta}} \text{ for } j \in J_i \\ 0 \end{cases} \quad (1)$$

  where J_i is the set of neighbor nodes that have not been visited by the ant, τ_{ij} and η_{ij} are the pheromone and heuristic information of the edge connecting node i and node j.
- Step 3: the process of selecting the next node is repeated until a traversal stopping criteria is satisfied.

The generated candidate feature subsets are then evaluated by a fitness function, and only the best ant is used to update the level of pheromone on each edge. The process of generating feature subsets and evaluating the generated features is repeated until a stopping criterion is reached. The best feature subset is then output as the final feature subset.

There are several essential questions raised when designing a standard ACO based algorithms:

- What is the traversal stopping criteria, i.e., when an ant should stop building its feature subset?
- For a node i, how to define the neighboring nodes (J_i)? The simplest way is to allow each node to connect to all other nodes, i.e., the graph is fully connected.
- How to define the heuristic information η? Tallln-Ballesteros and Riquelme [147] showed that using information gain to calculate a heuristic value could improve the performance over a pure ACO based algorithm that did not have any heuristic value. Therefore, it is essential to define the heuristic information η thoroughly.

Different algorithms have different answers to the above questions. In one of the early works, a wrapper ACO based feature selection was proposed for a network intrusion detection task [148]. The standard graph representation was used in the proposed algorithm, where each ant stopped building its feature subsets once it already selected a predefined number of features. The heuristic information between a pair of nodes (features) was measured by Fish Discrimination rate which showed how relevant (redundancy) the two features were. The candidate feature subsets were evaluated by the squared error obtained by training an SVMs classification algorithm on the subset. Only the best feature subset with the smallest error was allowed to update the pheromone level at each node. The main idea was to increase the pheromone level of all the edges appearing in the best feature subset. Meanwhile, the pheromone level of the unselected edges was evaporated over time. Kanan et al. [149] proposed a similar ACO based feature selection. The number of selected features and the classification accuracy calculated by KNN were used to define the heuristic value. However, it was not clear what was the traversal stopping criteria and how the heuristic values were updated.

Similar to PSO, ACO also suffers the premature convergence problem where all ants had the same or similar paths. To avoid such problem, many works attempt to modify the updating rules of pheromone and heuristic values which play essential roles in ACO's search mechanism. In [28], a dynamic traversal stopping criteria was proposed where the predefined number of selected features was increased at a constant rate. Two updating rules — local updating rules and global updating rule — were also proposed. While the global one only allowed the best candidate subset to update the pheromone, the local one focused on getting rid of irrelevant features and gave more chance to features that never been selected to be considered. The local rule was designed to prevent premature convergence. The two rules were also used in [150] to perform feature selection for surface electromyography signals classification. In the proposed algorithm, the heuristic value is calculated by mRMR [124]. The same updating strategy was also used in [151] to select features for intrusion detection. The selected features assisted SVMs to achieve better classification performance than using all features. Joseph [152] also utilized the two rules to perform feature selection for text data. Peng et al. [153] proposed two stages of updating pheromone. At the first stage, all the paths traversed in the current iterations were used to update the pheromone level of their edges. The updating pheromone amount depended on the classification accuracy and the number of selected features of the corresponding path. The second stage added more pheromone to edges that appearing in the best path of the current iteration. The two-stages updating rule provided more randomness to the process of building feature subsets. Therefore, the algorithm could avoid local optimal and achieved higher performance than two recently proposed ACO based feature selection algorithms [154], [155].

A different direction to avoid premature convergence is to control the trade-off between exploration and exploitation better. Kabir et al. [156] proposed an ACO algorithm that kept track of two best solutions: the best solution discovered so far — called global best, and the best solution discovered in the current iteration — called local best. The two best solutions contributed to updating the pheromone level so that the exploitation and the exploration were balanced. Besides, a dynamic traversal stopping criteria was defined as a predefined number of features that were randomly selected by a roulette wheel selection. The aim was to let the smaller subset size had

a higher chance to be selected. The same idea was also applied in [157] to perform feature selection for a speech processing task. Forasti et al. [158] also proposed a new selection rule that aimed to balance between exploration and exploitation. The selection probability of an edge depends on its pheromone and its selected frequency. In other words, the selected frequency is the heuristic value of an edge. Two pheromone updating rules called local and final rules were also proposed. While the local one increased the pheromone value of the rarely visited edges — increased exploration ability, the final one increased the pheromone value of the popularly visited edge — increased exploitation ability. Two local searches based on the opposite mechanism and sequential adding/removing features were integrated into the proposed algorithm. The results showed that the proposed algorithm could achieve better performance than benchmark ACO and PSO based algorithms. However, the superiority was subjected to the number of selected features that needed to be predefined for each dataset. Besides, due to the complexity, a large number of parameters needed to be tuned.

While most standard ACO based feature selection algorithms used the number of selected features as the traversal stopping criteria, Rashno et al. [159] proposed a traversal stopping criteria that consists of both the number of selected features and the classification accuracy. Although the proposed algorithm achieved good performance, users needed to set the weights to combine the above two terms. Besides, including the classification performance could significantly enlarge the computation cost. This algorithm was applied to image analysis.

A hybrid algorithm combining of DE and ACO was proposed in [160]. In the proposed algorithm, at each iteration, instead of starting with a single node, the ants started from a small number of nodes that were selected by the top feature subset candidates from the previous iteration. Once all ants finished building their feature subset candidates, the top candidates evaluated by the Linear Discriminant Analysis were passed to DE to be further improved. The idea of initializing each ant with a core set of good features was also used in [161]. The heuristic values calculated by mutual information were updated with respect to the core feature set. Hamamoto [162] proposed another hybrid algorithm that combined ACO and GAs to achieve feature selection. Menghour and Souici-Meslati [163] investigated three mechanisms to combine ACO and PSO. The first mechanism ran ACO and PSO together, and the best solution evolved by both solution was used to evolve the next population. The second mechanism ran ACO first, then the best solutions evolved by ACO were used to initialize PSO. In the third mechanism, ACO used the idea of PSO to maintain *gbest* and *pbest* which were then utilized to update the pheromone levels. The results showed that the first mechanism achieved the best classification performance.

Most of ACO based feature selection algorithms used a fully connected graph which was too complicated. There have been some attempts to reduce complexity. In [164], the pheromone level and the heuristic value were assigned to each node instead of each edge. The proposed strategy significantly reduced the number of pheromone and heuristic values since the number of nodes was usually smaller than the number of edges, especially when there were a large number of nodes. However, one disadvantage of this scheme was the missing feature interactions. Particularly, a pheromone level of an edge represented how good when the two connected features were selected together, while a pheromone level of a node represented how good the corresponding individual feature. Chen et al. [165] reduced the graph complexity by forcing an order between features, i.e. one feature could connect to only one other feature. Hence, the number of edges was even smaller than the number of nodes. Although the proposed mechanism significantly reduced the number of pheromone and heuristic values, it also reduced the chance of detecting feature interactions. In addition, the order of features significantly affected the algorithm's performance. Zhao et al. [166], on the other hand, reduced the complexity by restricting the possible selection at each node. The main idea was to group similar features in the same group. The traversal process had to ensure that each group contributed at least one feature and no more than four features. The constraint could effectively reduce the scale of the search space and avoid selecting too many redundant features. Note that, a feature was added only if it could improve the classification performance of the current feature subset. Rashno [167] also divided the original feature set into 13 feature clusters, but only one feature was selected from each cluster. The proposed algorithm significantly reduced the data dimensionality while preserving the classification performance.

In ACO based feature selection algorithms, the pheromone values were usually updated by the classification performance, and the heuristic values were calculated by a filter measure such as mutual information [164], Fisher score [165]. Hence, ACO based feature selection algorithms could be considered hybrid algorithms that combine both wrapper and filter. However, several works did not use any classification algorithm to update the pheromone values. Naseer et al. [168] proposed a pure filter ACO based feature selection approach where the goodness of the feature subsets and the heuristic values are calculated based on a gain ratio. The algorithm also maintained the ten best feature subsets which were then evaluated based on an ensemble classifier. The feature subset with the highest accuracy was the final subset of the algorithm. Tabakhi et al. [169] used a similarity measure as the heuristic value. The pheromone was assigned to each node, and it was updated by the number of times the corresponding feature appeared in a path. The results show that the proposed algorithm always achieved better performance than some univariate filter feature selection algorithms since it considered the redundancy between features. However, the proposed approach was not compared

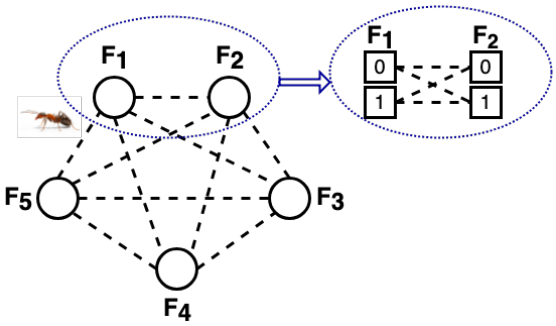

Figure 3: Binary fully-connected graph representation: each node is represented by two sub-nodes including sub-node 0 and sub-node 1.

with other EC based feature selection algorithms. Another filter ACO based approach was proposed by Dadaneh et al. [170] where the Pearson correlation was used to calculate the heuristic value. The overall goal was to minimize the redundancy between selected features. The selected features had comparative performance in comparison with other filter approaches on a wide range of classification algorithms. Mehmood et al. [171] utilized information gain to measure the relevance of the candidate subsets generated by ACO. The number of selected features was also used in the fitness function to avoid selecting redundant features. Experimental results on UCI datasets showed that the selected features were generalized to both KNN and C4.5 classification algorithms. Many other filter measures have also been widely used in ACO based feature selections, for example, Fisher score [172], fuzzy set [173].

3.4. Binary graph representation

The standard graph representation represents each feature as a node, so there is only one edge connecting between two features. In contrast, the binary graph representation represents each feature as two sub-nodes: 0 and 1, so four edges are connecting between two features, as shown in Figure 3. Therefore, in comparison with the standard graph representation, the binary graph representation needs four times (computational and memory) cost to maintain the edge information, i.e. the pheromone and heuristic values. However, the advantage of the binary representation is that it does not need to specify the traversal stopping criteria as in the standard one. An ant simply traverses all the nodes. At each node, the ant can visit either the sub-node 1 or the sub-node 0, which indicates the corresponding feature is either selected or discarded.

One of the early works applied the binary representation to feature selection was proposed by Yan and Yuan [174]. The aim was to improve the recognition rate in a face recognition task. Firstly, eigen-features were extracted from facial images using PCA. The eigen-features were then ranked according to their eigen-values so that there was a sequence of features that ants could follow for building their feature subsets. The heuristic value was the number of selected features so far, which was dynamically determined during the traversal process. The idea was to consider the trade-off between increasing the pheromone intensity of the path and the number of selected features. The pheromone updating process was similar to the standard graph representation, except the number of edges to updated was four times larger. The proposed algorithm could select features that can significantly improve classification performance than using all features. However, the proposed algorithm was not compared with a standard ACO based feature selection algorithm. Yu et al. [175] applied the same idea to select tumor-related marker genes. The feature subsets were also evaluated by SVM. However, there were no heuristic values used in the proposed algorithm. The combination of ACO and SVMs was also used by Kadri et al. [176], where ACO was designed to perform not only feature selection but also parameter selection for SVM. The proposed algorithm achieved a lower classification error rate than two feature selection algorithms based on standard ACO and GAs. The binary representation with ordered features was also used in [165, 177].

One limitation of the above works was the order of features was pre-defined, which essentially limited the search space since a feature could be connected to only one other feature. Kashef and Nezamabadi-pour [178], on the other hand, used a fully-connected topology that allowed a feature to connect to all other features. Experimental results on three UCI datasets showed that the fully-connected topology yielded better feature subsets with higher classification performance.

3.5. Discussion about ACO based feature selection algorithms

In summary, ACO usually represents feature selection as a graph problem which is flexible but does not scale well. Let take the standard graph as an example. Suppose there are n original features, the total number of edges in a fully connected graph is $E = n \times (n-1)/2$, which means the algorithm needs to maintain E pheromone values and E heuristic values. Besides, the standard graph representation requires a traversal stopping criteria to be predefined, which limits the search space. In contrast, the binary graph representation does not need the traversal stopping criteria, but it significantly enlarges the search space since there are four edges connecting every pair of features. Although many attempts have been made to improve ACO's scalability, they also restrict the interaction between features. Hence, ACO has been applied mainly to problems with small numbers of features (less than 100).

In ACO, the traversal process of building feature subsets is determined by two factors: the pheromone level and the heuristic values. In most ACO based feature selection algorithms, the classification performance is used to update the pheromone level, and a filter measure such as information gain, F-score is used to calculate the heuristic value. Hence, the ACO based algorithms can be considered hybrid approaches that have promising performance.

However, there have been very few works analyzing the interaction between the two essential factors. Furthermore, in all ACO based works, the pheromone and heuristic values are defined between two features, so they measure two-way interactions only. Meanwhile, three-way or higher interactions are also common in real-world feature selection problems. Therefore, the performance of ACO can be further improved by considering such interactions. There are also very few studies on ACO based multi-objective feature selection which is still an open issue.

4. Issues and Challenges

Although SI algorithms, especially PSO, ABC, and ACO, have been successfully applied to feature selection, we still believe there is more work can be done to further improve the performance of SI based feature selection algorithms. The following subsection presents several challenges that need to be discussed.

4.1. Representation

More papers are using standard representations than binary representations which are designed specifically for feature selection. The possible reason is that the standard representation, for example, continuous PSO, has been studied for a long time. However, such standard representation is not the most natural representation for feature selection. Meanwhile, the binary representation, the most natural representation, is not well studied yet. Recent works on binary PSO [115], ABC [140] show that if the characteristics of binary search spaces are considered in the representation and updating mechanism, the obtained performance is even better than the standard representation. However, these are still very initial work, and more investigation is needed.

The second limitation is that the current representation usually shows which features are selected. However, in many real-world applications, it is also essential to know the interaction between features, i.e. which feature combinations are good. Therefore, a good representation should be able to reveal such information to users.

The third limitation is the scalability of the representation. For PSO and ABC, the vector-based representation results in the search space size is 2^n where n is the number of features. The search space of ACO is even larger since the task is to select a subset of edges from $n \times (n-1)/2$ edges. However, the number of features in many real-world applications can reach thousands or even millions of features. SI usually do not work well on such high-dimensional problems [179]. A simple way is to rank features and select top-ranked features. The selected features were further refined by SI based feature selection algorithms. However, selecting top-ranked features may miss many feature interactions. A good representation with corresponding updating mechanisms, which can reduce the search space size of such large-scale feature selection, is still an open issue.

4.2. Multi-objective Feature Selection

Most of the existing SI based feature selection algorithms are dominance-based algorithms which assumes that the two objectives are equally important and in conflict. However, the assumptions is not true in feature selection. Firstly, the two objectives of feature selection are not always conflicting with each other. For example, removing irrelevant features can improve the classification performance. Secondly, the two objectives are not equally important since in most feature selection problems, achieving higher classification performance is usually more important than reducing the number of features. In addition, feature selection is a binary problem, so its Pareto front is also discrete. It has been shown that dominance-based algorithms do not generate an evenly distributed approximated front for a discrete Pareto front [180, 181]. This requires the development of multi-objective SI based feature selection algorithms that take the above feature selection characteristics into account.

4.3. Embedded SI based Feature Selection

Most, if not all, SI based feature selection algorithms belong to either wrapper or filter categories. Among the three feature selection categories, embedded approaches seem to have the best trade-off between effectiveness and efficiency. Recently, sparse-learning based feature selection methods —an embedded feature selection approach— have received much attention. The main task of sparse-learning based methods is to learn a set of feature coefficients that can form a classification model and indicate whether the corresponding features are selected or not. However, most sparse-learning based methods are gradient-based, which are easy to be trapped at local optima. SI algorithms, such as PSO and ABC, are well-known for their global search ability, and they have been widely applied to optimize coefficients [182]. Therefore, it is promising to apply SI algorithms to achieve sparse-learning feature selection, which can be considered embedded SI based feature selection.

5. Conclusions

This paper provided a comprehensive survey of SI based feature selection algorithms, which covers the three most common SI algorithms: PSO, ABC, and ACO. The main focus of this paper is the representation and the corresponding updating mechanism. Current issues and challenges were also discussed.

The survey shows that there have been many studies attempting to not only apply SI to feature selection but also improve the selection performance. A common approach is to optimize both parameters of the wrapped classifiers and the feature subset. The performance can also be improved by modifying the updating mechanism which is different for different algorithms due to their characteristics. In

PSO, the swarm is led by *gbest* and *pbest*, so most modifications are to enhance the two best positions with an expectation of improving the swarm's quality. ACO generates feature subsets based on the pheromone and heuristic values, so most ACO studies focus on updating rules of the two values to balance between exploration and exploitation. ABC is less mature than PSO and ACO, and most studies aim to improve the performance of ABC by combining it with other algorithms. The survey also shows that the standard representations of SI algorithms are suitable to feature selection, but they are not as natural as the binary representation. Although there have been several studies on the binary representation, it is still necessary to have more investigation, especially on the updating mechanism and parameter control, so the performance of SI feature selection algorithms can be further improved.

References

[1] Ishwarappa, J. Anuradha, A brief introduction on big data 5vs charateristics and hadoop technology, Procedia Computer Science 48 (2015) 319–324.

[2] R. Bellman, Dynamic programming, Courier Corporation, 2013.

[3] E. Keogh, A. Mueen, Curse of Dimensionality, Springer US, Boston, MA, 2017, pp. 314–315.

[4] H. Zhao, A. P. Sinha, W. Ge, Effects of feature construction on classification performance: An empirical study in bank failure prediction, Expert Systems with Applications 36 (2) (2009) 2633–2644.

[5] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, The Journal of Machine Learning Research 3 (2003) 1157–1182.

[6] A. Lensen, B. Xue, M. Zhang, Using particle swarm optimisation and the silhouette metric to estimate the number of clusters, select features, and perform clustering, in: G. Squillero, K. Sim (Eds.), Applications of Evolutionary Computation, Lecture Notes in Computer Science, Springer International Publishing, 2017, pp. 538–554.

[7] Y. Zhang, H.-G. Li, Q. Wang, C. Peng, A filter-based bare-bone particle swarm optimization algorithm for unsupervised feature selection, Applied Intelligence.

[8] Q. Chen, M. Zhang, B. Xue, Feature selection to improve generalization of genetic programming for high-dimensional symbolic regression, IEEE Transactions on Evolutionary Computation 21 (5) (2017) 792–806.

[9] A. A. Albrecht, Stochastic local search for the feature set problem, with applications to microarray data, Applied Mathematics and Computation 183 (2) (2006) 1148–1164.

[10] S. Fong, S. Deb, X.-S. Yang, J. Li, Feature selection in life science classification: metaheuristic swarm search, IT Professional 16 (4) (2014) 24–29.

[11] V. Kothari, J. Anuradha, S. Shah, P. Mittal, A survey on particle swarm optimization in feature selection, in: International Conference on Computing and Communication Systems, Springer, 2011, pp. 192–201.

[12] M. A. bin Basir, F. binti Ahmad, Comparison on swarm algorithms for feature selections/reductions, International Journal of Scientific & Engineering.

[13] B. Xue, M. Zhang, W. N. Browne, X. Yao, A survey on evolutionary computation approaches to feature selection, IEEE Transactions on Evolutionary Computation 20 (4) (2016) 606–626.

[14] L. Brezočnik, I. Fister, V. Podgorelec, Swarm intelligence algorithms for feature selection: A review, Applied Sciences 8 (9) (2018) 1521.

[15] M. Dash, H. Liu, Feature selection for classification, Intelligent data analysis 1 (1-4) (1997) 131–156.

[16] K. Kira, L. A. Rendell, The feature selection problem: Traditional methods and a new algorithm, in: AAAI, Vol. 2, 1992, pp. 129–134.

[17] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, H. Liu, Feature selection: A data perspective, ACM Computing Surveys (CSUR) 50 (6) (2018) 94.

[18] A. Wang, N. An, G. Chen, L. Li, G. Alterovitz, Accelerating wrapper-based feature selection with K-nearest-neighbor, Knowledge-Based Systems 83 (2015) 81–91.

[19] V. Agrawal, S. Chandra, Feature selection using artificial bee colony algorithm for medical image classification, in: International Conference on Contemporary Computing, 2015, pp. 171–176.

[20] D. Mladenic, M. Grobelnik, Feature selection for unbalanced class distribution and naive bayes, in: ICML, Vol. 99, 1999, pp. 258–267.

[21] M.-L. Zhang, J. M. Peña, V. Robles, Feature selection for multi-label naive bayes classification, Information Sciences 179 (19) (2009) 3218–3229.

[22] H. Frohlich, O. Chapelle, B. Scholkopf, Feature selection for support vector machines by means of genetic algorithm, in: IEEE International Conference on Tools with Artificial Intelligence, 2003, pp. 142–148.

[23] H. Fröhlich, O. Chapelle, B. Schölkopf, Feature selection for support vector machines using genetic algorithms, International Journal on Artificial Intelligence Tools 13 (04) (2004) 791–800.

[24] Y. Wan, M. Wang, Z. Ye, X. Lai, A feature selection method based on modified binary coded ant colony optimization algorithm, Applied Soft Computing 49 (2016) 248–258.

[25] X. Wang, ACO and SVM selection feature weighting of network intrusion detection method, International Journal of Security and Its Applications 9 (4) (2015) 129–270.

[26] A. A. Altun, N. Allahverdi, Neural network based recognition by using genetic algorithm for feature selection of enhanced fingerprints, in: International Conference on Adaptive and Natural Computing Algorithms, Springer, 2007, pp. 467–476.

[27] A. A. Altun, H. E. Kocer, N. Allahverdi, Genetic algorithm based feature selection level fusion using fingerprint and iris biometrics, International Journal of Pattern Recognition and Artificial Intelligence 22 (03) (2008) 585–600.

[28] R. K. Sivagaminathan, S. Ramakrishnan, A hybrid approach for feature subset selection using neural networks and ant colony optimization, Expert Systems with Applications 33 (1) (2007) 49–60.

[29] M. Robnik-Šikonja, I. Kononenko, Theoretical and empirical analysis of ReliefF and RReliefF, Machine learning 53 (1-2) (2003) 23–69.

[30] M. Dash, H. Liu, H. Motoda, Consistency based feature selection, in: Knowledge Discovery and Data Mining. Current Issues and New Applications, Springer, 2000, pp. 98–109.

[31] M. A. Hall, Correlation-based feature selection of discrete and numeric class machine learning.

[32] H. Akaike, Information theory and an extension of the maximum likelihood principle, in: Selected Papers of Hirotugu Akaike, Springer, 1998, pp. 199–213.

[33] J. Tang, S. Alelyani, H. Liu, Feature selection for classification: A review, Data Classification: Algorithms and Applications (2014) 37.

[34] J. Gui, Z. Sun, S. Ji, D. Tao, T. Tan, Feature selection based on structured sparsity: A comprehensive study, IEEE Transactions on Neural Networks and Learning Systems 28 (7) (2017) 1490–1507.

[35] A. Y. Ng, Feature selection, l1 vs. l2 regularization, and rotational invariance, in: The International Conference on Machine learning, ACM, 2004, p. 78.

[36] F. Nie, H. Huang, X. Cai, C. H. Ding, Efficient and robust feature selection via joint l2, 1-norms minimization, in: Advances in Neural Information Processing Systems, 2010, pp.

1813–1821.
[37] H. Almuallim, T. G. Dietterich, Learning boolean concepts in the presence of many irrelevant features, Artificial Intelligence 69 (1) (1994) 279–305.
[38] H. Liu, Z. Zhao, Manipulating data and dimension reduction methods: Feature selection, Encyclopedia of Complexity and Systems Science (2009) 5348–5359.
[39] H. Liu, H. Motoda, R. Setiono, Z. Zhao, Feature selection: An ever evolving frontier in data mining, in: Feature Selection in Data Mining, 2010, pp. 4–13.
[40] A. W. Whitney, A direct method of nonparametric measurement selection, IEEE Transactions on Computers 100 (9) (1971) 1100–1103.
[41] T. Marill, D. M. Green, On the effectiveness of receptors in recognition systems, IEEE Transactions on Information Theory 9 (1) (1963) 11–17.
[42] S. D. Stearns, On selecting features for pattern classifiers., in: The International Conference on Pattern Recognition (ICPR), Coronado, CA, 1976, pp. 71–75.
[43] P. Pudil, J. Novovičová, J. Kittler, Floating search methods in feature selection, Pattern Recognition Letters 15 (11) (1994) 1119–1125.
[44] H. Peng, F. Long, C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (8) (2005) 1226–1238.
[45] S. Nakariyakul, D. P. Casasent, An improvement on floating search algorithms for feature subset selection , Pattern Recognition 42 (9) (2009) 1932–1940.
[46] J. Lee, D.-W. Kim, Feature selection for multi-label classification using multivariate mutual information, Pattern Recognition Letters 34 (3) (2013) 349–357.
[47] J. Lee, D.-W. Kim, Mutual information-based multi-label feature selection using interaction information, Expert Systems with Applications 42 (4) (2015) 2013–2025.
[48] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, 1st Edition, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
[49] J. Kennedy, J. F. Kennedy, R. C. Eberhart, Y. Shi, Swarm intelligence, Morgan Kaufmann, 2001.
[50] J. Kennedy, R. Eberhart, et al., Particle swarm optimization, in: IEEE International Conference on Neural Networks, Vol. 4, 1995, pp. 1942–1948.
[51] M. Dorigo, G. Di Caro, Ant colony optimization: a new metaheuristic, in: IEEE Congress on Evolutionary Computation, Vol. 2, 1999, pp. 1470–1477.
[52] D. Karaboga, An idea based on honey bee swarm for numerical optimization, Tech. rep., Technical report-tr06, Erciyes university, engineering faculty, computer (2005).
[53] R. C. Eberhart, Y. Shi, Comparison between genetic algorithms and particle swarm optimization, in: International conference on evolutionary programming, Springer, 1998, pp. 611–616.
[54] A. P. Piotrowski, M. J. Napiorkowski, J. J. Napiorkowski, P. M. Rowinski, Swarm intelligence and evolutionary algorithms: Performance versus speed, Information Sciences 384 (2017) 34–85.
[55] Y. Lu, M. Liang, Z. Ye, L. Cao, Improved particle swarm optimization algorithm and its application in text feature selection, Applied Soft Computing 35 (2015) 629–636.
[56] L. M. Abualigah, A. T. Khader, E. S. Hanandeh, A new feature selection method to improve the document clustering using particle swarm optimization algorithm, Journal of Computational Science 25 (2018) 456–466.
[57] N. Kushwaha, M. Pant, Link based BPSO for feature selection in big data text clustering, Future Generation Computer Systems 82 (2018) 190–199.
[58] X. Bai, X. Gao, B. Xue, Particle swarm optimization based two-stage feature selection in text mining, in: IEEE Congress on Evolutionary Computation, 2018, pp. 1–8.
[59] S. Fong, R. Wong, A. V. Vasilakos, Accelerated PSO swarm search feature selection for data stream mining big data, IEEE Transactions on Services Computing 9 (1) (2016) 33–45.
[60] T. Khadhraoui, S. Ktata, F. Benzarti, H. Amiri, Features selection based on modified PSO algorithm for 2D face recognition, in: International Conference on Computer Graphics, Imaging and Visualization, 2016, pp. 99–104.
[61] P. H. Silva, E. Luz, L. A. Zanlorensi, D. Menotti, G. Moreira, Multimodal feature level fusion based on particle swarm optimization with deep transfer learning, in: IEEE Congress on Evolutionary Computation, 2018, pp. 1–8.
[62] W. Srisukkham, L. Zhang, S. C. Neoh, S. Todryk, C. P. Lim, Intelligent leukaemia diagnosis with bare-bones PSO based feature optimization, Applied Soft Computing 56 (2017) 405–419.
[63] S. Udhaya Kumar, H. Hannah Inbarani, PSO-based feature selection and neighborhood rough set-based classification for BCI multiclass motor imagery task, Neural Computing and Applications 28 (11) (2017) 3239–3258.
[64] S. B. Sakri, N. B. A. Rashid, Z. M. Zain, Particle swarm optimization feature selection for breast cancer recurrence prediction, IEEE Access 6 (2018) 29637–29647.
[65] S.-W. Lin, K.-C. Ying, S.-C. Chen, Z.-J. Lee, Particle swarm optimization for parameter determination and feature selection of support vector machines, Expert Systems with Applications 35 (4) (2008) 1817–1824.
[66] C.-L. Huang, C.-J. Wang, A GA-based feature selection and parameters optimization for support vector machines, Expert Systems with applications 31 (2) (2006) 231–240.
[67] M.-Y. Cho, T. T. Hoang, Feature selection and parameters optimization of SVM using particle swarm optimization for fault classification in power distribution systems (2017).
[68] B. Tran, B. Xue, M. Zhang, A new representation in PSO for discretization-based feature selection, IEEE Transactions on Cybernetics 48 (6) (2018) 1733–1746.
[69] H. B. Nguyen, B. Xue, I. Liu, M. Zhang, PSO and statistical clustering for feature selection: a new representation, in: Simulated Evolution and Learning, Springer, 2014, pp. 569–581.
[70] H. B. Nguyen, B. Xue, I. Liu, P. Andreae, M. Zhang, Gaussian transformation based representation in particle swarm optimization for feature selection, in: Applications of Evolutionary Computation, Springer, 2015, pp. 541–553.
[71] B. Tran, B. Xue, M. Zhang, Variable-length particle swarm optimization for feature selection on high-dimensional classification, IEEE Transactions on Evolutionary Computation (2018) 1–1.
[72] B. P. Flannery, W. H. Press, S. A. Teukolsky, W. Vetterling, Numerical recipes in C, Press Syndicate of the University of Cambridge, New York 24 (1992) 78.
[73] F. Wang, J. Liang, An efficient feature selection algorithm for hybrid data, Neurocomputing 193 (2016) 33–41.
[74] H. B. Nguyen, B. Xue, P. Andreae, Surrogate-model based particle swarm optimization with local search for feature selection in classification, in: G. Squillero, K. Sim (Eds.), Applications of Evolutionary Computation, Lecture Notes in Computer Science, Springer International Publishing, 2017, pp. 487–505.
[75] H. B. Nguyen, B. Xue, P. Andreae, Pso with surrogate models for feature selection: static and dynamic clustering-based methods, Memetic Computing 10 (3) (2018) 291–300.
[76] T. Butler-Yeoman, B. Xue, M. Zhang, Particle swarm optimisation for feature selection: A hybrid filter-wrapper approach, in: IEEE Congress on Evolutionary Computation, 2015, pp. 2428–2435.
[77] B. Xue, M. Zhang, W. N. Browne, Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms, Applied Soft Computing 18 (2014) 261–276.
[78] B. Tran, B. Xue, M. Zhang, Improved PSO for feature selection on high-dimensional datasets, in: Simulated Evolution and Learning, Springer, 2014, pp. 503–515.
[79] B. Tran, M. Zhang, B. Xue, A PSO based hybrid feature selection algorithm for high-dimensional classification, in: IEEE Congress on Evolutionary Computation, 2016, pp. 3801–3808.

[80] H. Nguyen, B. Xue, I. Liu, M. Zhang, Filter based backward elimination in wrapper based PSO for feature selection in classification, in: IEEE Congress on Evolutionary Computation, 2014, pp. 3111–3118.
[81] K. Mistry, L. Zhang, S. C. Neoh, C. P. Lim, B. Fielding, A micro-GA embedded PSO feature selection approach to intelligent facial emotion recognition, IEEE Transactions on Cybernetics 47 (6) (2017) 1496–1509.
[82] H. B. Nguyen, B. Xue, P. Andreae, M. Zhang, Particle swarm optimization with genetic operators for feature selection, in: IEEE Congress on Evolutionary Computation, 2017, pp. 286–293.
[83] S. Gu, R. Cheng, Y. Jin, Feature selection for high-dimensional classification using a competitive swarm optimizer, Soft Computing 22 (3) (2018) 811–822.
[84] R. Cheng, Y. Jin, A competitive swarm optimizer for large scale optimization, IEEE Transactions on Cybernetics 45 (2) (2015) 191–204.
[85] Y. Shi, R. Eberhart, A modified particle swarm optimizer, in: IEEE International Conference on Evolutionary Computation, 1998, pp. 69–73.
[86] Y. Shi, R. C. Eberhart, Parameter selection in particle swarm optimization, in: International Conference on Evolutionary Programming, Springer, 1998, pp. 591–600.
[87] A. Adeli, A. Broumandnia, Image steganalysis using improved particle swarm optimization based feature selection, Applied Intelligence 48 (6) (2018) 1609–1622.
[88] R. C. Eberhart, Y. Shi, Tracking and optimizing dynamic systems with particle swarms, in: IEEE Congress on Evolutionary Computation, Vol. 1, 2001, pp. 94–100.
[89] Y. Feng, G.-F. Teng, A.-X. Wang, Y.-M. Yao, Chaotic inertia weight in particle swarm optimization, in: International Conference on Innovative Computing, Informatio and Control, IEEE, 2007, pp. 475–475.
[90] B. Xue, M. Zhang, W. Browne, Particle swarm optimization for feature selection in classification: A multi-objective approach, IEEE Transactions on Cybernetics 43 (6) (2013) 1656–1671.
[91] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197.
[92] E. Zitzler, M. Laumanns, L. Thiele, et al., SPEA2: Improving the strength pareto evolutionary algorithm, in: Eurogen, Vol. 3242, 2001, pp. 95–100.
[93] B. Xue, L. Cervante, L. Shang, W. N. Browne, M. Zhang, A multi-objective particle swarm optimisation for filter-based feature selection in classification problems, Connection Science 24 (2-3) (2012) 91–116.
[94] H. B. Nguyen, B. Xue, I. Liu, P. Andreae, M. Zhang, New mechanism for archive maintenance in PSO-based multi-objective feature selection, Soft Computing 20 (10) (2016) 3927–3946.
[95] Y. Zhang, D.-w. Gong, J. Cheng, Multi-objective particle swarm optimization approach for cost-based feature selection in classification, IEEE/ACM Transactions on Computational Biology and Bioinformatics 14 (1) (2017) 64–75.
[96] M. Amoozegar, B. Minaei-Bidgoli, Optimizing multi-objective PSO based feature selection method using a feature elitism mechanism, Expert Systems with Applications 113 (2018) 499–514.
[97] Y. Zhang, D. Gong, Y. Hu, W. Zhang, Feature selection algorithm based on bare bones particle swarm optimization, Neurocomputing 148 (2015) 150–157.
[98] P. Ghamisi, M. S. Couceiro, J. A. Benediktsson, A novel feature selection approach based on FODPSO and SVM, IEEE Transactions on Geoscience and Remote Sensing 53 (5) (2015) 2935–2947.
[99] M. Mafarja, R. Jarrar, S. Ahmad, A. A. Abusnaina, Feature selection using binary particle swarm optimization with time varying inertia weight strategies, in: International Conference on Future Networks and Distributed Systems, ACM, 2018, pp. 18:1–18:9, event-place: Amman, Jordan.
[100] A. A. Naeini, M. Babadi, S. M. J. Mirzadeh, S. Amini, Particle swarm optimization for object-based feature selection of VHSR satellite images, IEEE Geoscience and Remote Sensing Letters 15 (3) (2018) 379–383.
[101] S. Yadav, A. Ekbal, S. Saha, Feature selection for entity extraction from multiple biomedical corpora: A PSO-based approach, Soft Computing 22 (20) (2018) 6881–6904.
[102] O. S. Qasim, Z. Y. Algamal, Feature selection using particle swarm optimization-based logistic regression model, Chemometrics and Intelligent Laboratory Systems 182 (2018) 41–46.
[103] L.-Y. Chuang, H.-W. Chang, C.-J. Tu, C.-H. Yang, Improved binary PSO for feature selection using gene expression data, Computational Biology and Chemistry 32 (1) (2008) 29–38.
[104] C.-S. Yang, L.-Y. Chuang, C.-H. Ke, C.-H. Yang, Boolean binary particle swarm optimization for feature selection, in: IEEE Congress on Evolutionary Computation, 2008, pp. 2093–2098.
[105] I. Jain, V. K. Jain, R. Jain, Correlation feature selection based improved-binary particle swarm optimization for gene selection and cancer classification, Applied Soft Computing 62 (2018) 203–215.
[106] P. Moradi, M. Gholampour, A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy, Applied Soft Computing 43 (2016) 117–130.
[107] Y. Chen, Y. Wang, L. Cao, Q. Jin, An effective feature selection scheme for healthcare data classification using binary particle swarm optimization, in: International Conference on Information Technology in Medicine and Education, 2018, pp. 703–707.
[108] H. Dong, J. Sun, T. Li, L. Li, An improved niching binary particle swarm optimization for feature selection, in: IEEE International Conference on Systems, Man, and Cybernetics, 2018, pp. 3571–3577.
[109] S. M. Vieira, L. F. Mendonça, G. J. Farinha, J. M. Sousa, Modified binary PSO for feature selection using SVM applied to mortality prediction of septic patients, Applied Soft Computing 13 (8) (2013) 3494–3504.
[110] A. Boubezoul, S. Paris, Application of global optimization methods to model and feature selection, Pattern Recognition 45 (10) (2012) 3676–3686.
[111] M. C. Lane, B. Xue, I. Liu, M. Zhang, Particle swarm optimization and statistical clustering for feature selection, in: Advances in Artificial Intelligence, Springer, 2013, pp. 214–220.
[112] M. C. Lane, B. Xue, I. Liu, M. Zhang, Gaussian based particle swarm optimization and statistical clustering for feature selection, in: Evolutionary Computation in Combinatorial Optimisation, Springer, 2014, pp. 133–144.
[113] J. Liu, Y. Mei, X. Li, An analysis of the inertia weight parameter for binary particle swarm optimization, IEEE Transactions on Evolutionary Computation 20 (5) (2016) 666–681.
[114] H. Wang, R. Ke, J. Li, Y. An, K. Wang, L. Yu, A correlation-based binary particle swarm optimization method for feature selection in human activity recognition, International Journal of Distributed Sensor Networks 14 (4) (2018) 1550147718772785.
[115] B. H. Nguyen, B. Xue, P. Andreae, A novel binary particle swarm optimization algorithm and its applications on knapsack and feature selection problems, in: G. Leu, H. K. Singh, S. Elsayed (Eds.), Intelligent and Evolutionary Systems, Proceedings in Adaptation, Learning and Optimization, Springer International Publishing, 2017, pp. 319–332.
[116] D. Karaboga, An idea based on honey bee swarm for numerical optimization, Tech. rep., Technical report-tr06, Erciyes university, engineering faculty, computer (2005).
[117] M. Akila, V. Suresh Kumar, N. Anusheela, K. Sugumar, A novel feature subset selection algorithm using artificial bee colony in keystroke dynamics, in: K. Deep, A. Nagar, M. Pant, J. C. Bansal (Eds.), International Conference on Soft Computing for Problem Solving, Advances in Intelligent and Soft

Computing, Springer India, 2012, pp. 813–820.
[118] M. Y. SyarifahAdilah, R. Abdullah, I. Venkat, ABC algorithm as feature selection for biomarker discovery in mass spectrometry analysis, in: Conference on Data Mining and Optimization, 2012, pp. 67–72.
[119] M. S. Uzer, N. Yilmaz, O. Inan, Feature selection method based on artificial bee colony algorithm and support vector machines for medical datasets classification (2013).
[120] R. J. Kuo, S. B. L. Huang, F. E. Zulvia, T. W. Liao, Artificial bee colony-based support vector machines with feature selection and parameter optimization for rule extraction, Knowledge and Information Systems 55 (1) (2018) 253–274.
[121] H. M. Alshamlan, G. H. Badr, Y. A. Alohali, ABC-SVM: artificial bee colony and SVM method for microarray gene selection and multi-class cancer classification, International Journal of Machine Learning and Computing 6 (3) (2016) 184–190.
[122] P. Rakshit, S. Bhattacharyya, A. Konar, A. Khasnobish, D. N. Tibarewala, R. Janarthanan, Artificial bee colony based feature selection for motor imagery EEG data, in: J. C. Bansal, P. Singh, K. Deep, M. Pant, A. Nagar (Eds.), Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012), Advances in Intelligent Systems and Computing, Springer India, 2013, pp. 127–138.
[123] H. Alshamlan, G. Badr, Y. Alohali, mRMR-ABC: a hybrid gene selection algorithm for cancer classification using microarray gene expression profiling (2015).
[124] H. Peng, F. Long, C. Ding, Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy, IEEE Transactions on Pattern Analysis & Machine Intelligence (8) (2005) 1226–1238.
[125] P. Shunmugapriya, S. Kanmani, A hybrid algorithm using ant and bee colony optimization for feature selection and classification, Swarm and Evolutionary Computation 36 (2017) 27–36.
[126] W. A. H. M. Ghanem, A. Jantan, Novel multi-objective artificial bee colony optimization for wrapper based feature selection in intrusion detection, International Journal of Advance Soft Computing Applications.
[127] E. Hancer, B. Xue, M. Zhang, D. Karaboga, B. Akay, A multi-objective artificial bee colony approach to feature selection using fuzzy mutual information, in: IEEE Congress on Evolutionary Computation, 2015, pp. 2420–2427.
[128] T. Prasartvit, B. Kaewkamnerdpong, T. Achalakul, Dimensional reduction based on artificial bee colony for classification problems, in: D.-S. Huang, Y. Gan, P. Premaratne, K. Han (Eds.), Bio-Inspired Computing and Applications, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012, pp. 168–175.
[129] S. Palanisamy, S. Kanmani, Artificial bee colony approach for optimizing feature selection, International Journal of Computer Science Issues 9 (3) (2012) 432–438.
[130] F. G. Mohammadi, M. S. Abadeh, Image steganalysis using a bee colony based feature selection algorithm, Engineering Applications of Artificial Intelligence 31 (2014) 35–43.
[131] G. Yavuz, D. Aydin, Angle modulated artifical bee colony algorithms for feature selection, Applied Computational Intelligence and Soft Computing 2016 (2016) 7.
[132] M. Schiezaro, H. Pedrini, Data feature selection based on artificial bee colony algorithm, EURASIP Journal on Image and Video Processing 2013 (1) (2013) 47.
[133] E. Hancer, B. Xue, M. Zhang, D. Karaboga, B. Akay, Pareto front feature selection based on artificial bee colony optimization, Information Sciences 422 (2018) 462–479.
[134] E. Hancer, B. Xue, D. Karaboga, M. Zhang, A binary ABC algorithm based on advanced similarity scheme for feature selection, Applied Soft Computing 36 (2015) 334–348.
[135] Z. B. zger, B. Bolat, B. Dr, A comparative study on binary Artificial Bee Colony optimization methods for feature selection, in: International Symposium on Innovations in Intelligent Systems and Applications, 2016, pp. 1–4.
[136] D. Jia, X. Duan, M. K. Khan, Binary artificial bee colony optimization using bitwise operation, Computers & Industrial Engineering 76 (2014) 360–365.
[137] L. Wei, C. Hanning, BABC: a binary version of artificial bee colony algorithm for discrete optimization, International Journal of Advancements in Computing Technology 4 (14) (2012) 307–14.
[138] L. Wei, N. Ben, C. Hanning, Binary artificial bee colony algorithm for solving 0-1 knapsack problem, Adv Inf Sci Serv Sci 4 (22) (2012) 464–470.
[139] M. Mandala, C. Gupta, Binary artificial bee colony optimization for GENCOs' profit maximization under pool electricity market, International Journal of Computer Applications 90 (19).
[140] E. Zorarpac, S. A. zel, A hybrid approach of differential evolution and artificial bee colony for feature selection, Expert Systems with Applications 62 (2016) 91–103.
[141] X. Li, M. Li, M. Yin, Multiobjective ranking binary artificial bee colony for gene selection problems using microarray datasets, IEEE/CAA Journal of Automatica Sinica (2018) 1–16.
[142] M. Dorigo, G. Di Caro, Ant colony optimization: a new meta-heuristic, in: IEEE Congress on Evolutionary Computation, Vol. 2, 1999, pp. 1470–1477.
[143] M. Dorigo, T. Stützle, Ant Colony Optimization: Overview and Recent Advances, Springer International Publishing, Cham, 2019, pp. 311–351.
[144] P. Balaprakash, M. Birattari, T. Stützle, Z. Yuan, M. Dorigo, Estimation-based ant colony optimization and local search for the probabilistic traveling salesman problem, Swarm Intelligence 3 (3) (2009) 223–242.
[145] C. Blum, Beam-ACOhybridizing ant colony optimization with beam search: an application to open shop scheduling, Computers and Operations Research 32 (6) (2005) 1565 – 1591.
[146] C. Blum, M. Y. Valls, M. J. Blesa, An ant colony optimization algorithm for DNA sequencing by hybridization, Computers and Operations Research 35 (11) (2008) 3620 – 3635, part Special Issue: Topics in Real-time Supply Chain Management.
[147] A. J. Talln-Ballesteros, J. C. Riquelme, Tackling ant colony optimization meta-heuristic as search method in feature subset selection based on correlation or consistency measures, in: E. Corchado, J. A. Lozano, H. Quintin, H. Yin (Eds.), Intelligent Data Engineering and Automated Learning, Lecture Notes in Computer Science, Springer International Publishing, 2014, pp. 386–393.
[148] a. and, Ant colony optimization based network intrusion feature selection and detection, in: International Conference on Machine Learning and Cybernetics, Vol. 6, 2005, pp. 3871–3875 Vol. 6.
[149] H. R. Kanan, K. Faez, S. M. Taheri, Feature Selection Using Ant Colony Optimization (ACO): A New Method and Comparative Study in the Application of Face Recognition System, in: P. Perner (Ed.), Advances in Data Mining. Theoretical Aspects and Applications, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2007, pp. 63–76.
[150] H. Huang, H.-B. Xie, J.-Y. Guo, H.-J. Chen, Ant colony optimization-based feature selection method for surface electromyography signals classification, Computers in Biology and Medicine 42 (1) (2012) 30–38.
[151] T. Mehmod, H. B. M. Rais, Ant colony optimization and feature selection for intrusion detection, in: Advances in machine learning and signal processing, Springer, 2016, pp. 305–312.
[152] R. Joseph Manoj, M. D. Anto Praveena, K. Vijayakumar, An ACO-ANN based feature selection algorithm for big data, Cluster Computing.
[153] H. Peng, C. Ying, S. Tan, B. Hu, Z. Sun, An improved feature selection algorithm based on ant colony optimization, IEEE Access 6 (2018) 69203–69209.
[154] K. Lutvica, S. Konjicija, Alternative pheromone laying strategyan improvement for the ACO algorithm, in: IEEE International Conference on Information, Communication and Automation Technologies, 2017, pp. 1–6.

[155] D. Meier, I. Tullumi, Y. Stauffer, R. Dornberger, T. Hanne, A novel backup path planning approach with ACO, in: IEEE International Symposium on Computational and Business Intelligence, 2017, pp. 50–56.
[156] M. M. Kabir, M. Shahjahan, K. Murase, A new hybrid ant colony optimization algorithm for feature selection, Expert Systems with Applications 39 (3) (2012) 3747–3763.
[157] R. R. Rajoo, R. A. Salam, Ant colony optimization based subset feature selection in speech processing: Constructing graphs with degree sequences, International Journal on Advanced Science, Engineering and Information Technology 8 (4-2) (2018) 1728–1734.
[158] R. Forsati, A. Moayedikia, R. Jensen, M. Shamsfard, M. R. Meybodi, Enriched ant colony optimization and its application in feature selection, Neurocomputing 142 (2014) 354–371.
[159] A. Rashno, S. Sadri, H. SadeghianNejad, An efficient content-based image retrieval with ant colony optimization feature selection schema based on wavelet and color features, in: International Symposium on Artificial Intelligence and Signal Processing (AISP), 2015, pp. 59–64.
[160] R. N. Khushaba, A. Al-Ani, A. AlSukker, A. Al-Jumaily, A combined ant colony and differential evolution feature selection algorithm, in: M. Dorigo, M. Birattari, C. Blum, M. Clerc, T. Sttzle, A. F. T. Winfield (Eds.), Ant Colony Optimization and Swarm Intelligence, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2008, pp. 1–12.
[161] Y. Chen, D. Miao, R. Wang, A rough set approach to feature selection based on ant colony optimization, Pattern Recognition Letters 31 (3) (2010) 226–233.
[162] A. H. Hamamoto, L. F. Carvalho, M. L. Proenca, ACO and GA metaheuristics for anomaly detection, in: International Conference of the Chilean Computer Science Society, 2015, pp. 1–6.
[163] K. Menghour, L. Souici-Meslati, Hybrid ACO-PSO based approaches for feature selection, International Journal of Intelligent Engineering and Systems 9 (3) (2016) 65–79.
[164] A. Al-Ani, Ant colony optimization for feature subset selection., in: WEC (2), 2005, pp. 35–38.
[165] B. Chen, L. Chen, Y. Chen, Efficient ant colony optimization for image feature selection, Signal Processing 93 (6) (2013) 1566–1576.
[166] X. Zhao, D. Li, B. Yang, C. Ma, Y. Zhu, H. Chen, Feature selection based on improved ant colony optimization for online detection of foreign fiber in cotton, Applied Soft Computing 24 (2014) 585–596.
[167] A. Rashno, B. Nazari, S. Sadri, M. Saraee, Effective pixel classification of mars images based on ant colony optimization feature selection and extreme learning machine, Neurocomputing 226 (2017) 66–79.
[168] A. Naseer, W. Shahzad, A. Ellahi, A hybrid approach for feature subset selection using ant colony optimization and multi-classifier ensemble, International Journal of Advanced Computer Science and Applications 9 (1).
[169] S. Tabakhi, P. Moradi, F. Akhlaghian, An unsupervised feature selection algorithm based on ant colony optimization, Engineering Applications of Artificial Intelligence 32 (2014) 112–123.
[170] B. Z. Dadaneh, H. Y. Markid, A. Zakerolhosseini, Unsupervised probabilistic feature selection using ant colony optimization, Expert Systems with Applications 53 (2016) 27–42.
[171] R. Mehmood, W. Shahzad, E. Ahmed, Maximum relavancy minimum redundancy based feature subset selection using ant colony optimization, Journal of Applied Environmental and Biological Sciences 7 (4) (2017) 118–130.
[172] P. Moradi, M. Rostami, Integration of graph clustering with ant colony optimization for feature selection, Knowledge-Based Systems 84 (2015) 144–161.
[173] P. R. K. Varma, V. V. Kumari, S. S. Kumar, Feature selection using relative fuzzy entropy and ant colony optimization applied to real-time intrusion detection system, Procedia Computer Science 85 (2016) 503–510.
[174] Z. Yan, C. Yuan, Ant colony optimization for feature selection in face recognition, in: D. Zhang, A. K. Jain (Eds.), Biometric Authentication, Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2004, pp. 221–226.
[175] H. Yu, G. Gu, H. Liu, J. Shen, J. Zhao, A modified ant colony optimizatoin algorithm for tumor marker gene selection, Genomics, Proteomics & Bioinformatics 7 (4) (2009) 200–208.
[176] O. Kadri, L. H. Mouss, M. D. Mouss, Fault diagnosis of rotary kiln using SVM and binary ACO, Journal of Mechanical Science and Technology 26 (2) (2012) 601–608.
[177] N. G. R. Chawla, Improved feature subset selection using hybrid ant colony and perceptron network, International Journal of Scientific Research and Management 5 (8) (2017) 6764–6770.
[178] S. Kashef, H. Nezamabadi-pour, A new feature selection algorithm based on binary ant colony optimization, in: IEEE Conference on Information and Knowledge Technology, 2013, pp. 50–54.
[179] S. Cheng, B. Liu, Y. Shi, Y. Jin, B. Li, Evolutionary computation and big data: key challenges and future directions, in: International Conference on Data Mining and Big Data, Springer, 2016, pp. 3–14.
[180] A. Jaszkiewicz, On the computational efficiency of multiple objective metaheuristics. the knapsack problem case study, European Journal of Operational Research 158 (2) (2004) 418–433.
[181] B. Xue, M. Zhang, W. N. Browne, Particle swarm optimization for feature selection in classification: A multi-objective approach, IEEE Transactions on Cybernetics 43 (6) (2013) 1656–1671.
[182] Y. Zhang, S. Wang, G. Ji, A comprehensive survey on particle swarm optimization algorithm and its applications, Mathematical Problems in Engineering 2015.