A Review on Evolutionary Feature Selection

Nadia Abd-Alsabour
Cairo University
Cairo, Egypt
nadia.abdalsabour@ieee.org

Abstract—This paper presents a review of some of the most recent evolutionary algorithms used for solving feature selection based upon previously published research on feature selection. In addition, we discuss various research issues relating to each of the presented evolutionary algorithm. Evolutionary algorithms present several advantages over traditional search such as they require less domain-specific information. Such advantages have made them very popular within feature selection as explained in this paper. This paper covers the first part only of the evolutionary algorithms for the feature selection problem due to the limitation of the number of pages. The references cited in this paper cover the major theoretical issues, and provide access to the main branches of the literature dealing with such methods.

Keywords— Feature Selection (FS), Genetic Algorithms (GAs), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO)

I. INTRODUCTION

Finding the optimal feature subset is an NP hard optimization problem that involves searching the space of possible feature subsets to identify the optimal one. There are $2^n$ states in the search space (where $n$ is the number of features in the dataset). Each state in the search space specifies a subset of the possible features [1]. For large $n$ values, evaluating all the states is computationally infeasible. If it is known beforehand that exactly $d$ features out of $n$ features should be chosen, then we only need to go through the subsets of size $d$. This results in the number of possible subsets is still too much for most values of $d$ [2].

Exhaustive search (that searches over all the possible feature subsets of a feature set) is usually time-consuming. Can one make some smart choices based on the minimum information available but without looking at the whole picture? This is all what heuristic search is about. The rationale of this non-optimal strategy is three-fold:

- it is quick to find a solution (a subset of features),
- it usually can find near optimal solution if not optimal, and
- the trade-off of optimality with the speed is often worthwhile because of a gain in speed with little loss of optimality [1]. This leads to the use of heuristic search that is much faster than exhaustive search.

Whenever a heuristic algorithm for an optimization problem allows the investment of an arbitrary amount of computation time and improving the current solution, one wishes to know what is likely to happen in the long run; will the current solution get closer and closer to an optimal solution (optimal solution convergence), or is it possible that there will always remain a gap to the optimum solution? [3]. Evolutionary algorithms are one example of heuristic approaches that have tackled feature selection problems. This is explained in Section 3. In this paper, we explain GA, ACO, and PSO algorithms for feature selection problems. It also investigates the numerous recent trends in employing them for feature selection.

The rest of this paper is organized as follows. Section 2 introduces feature selection. Section 3 addresses the fundamentals of the explained evolutionary algorithms used for solving feature selection. Section 4 concludes this paper and highlights future work in this area.

II. FEATURE SELECTION

Feature selection is defined as a process of finding a subset of features from the original set of features according to the criterion of feature selection [4]-[5]. Its main goal is to find a subset of features with predictive performance comparable to the full set of features [6]. Feature selection does not generate unwanted artifacts i.e., it is carried out in the original feature space. This is achieved by removing redundant and/or irrelevant features without losing the original concept of data [7].

Feature selection is one aspect of dimension reduction that is the main theme for simplifying the data. Discard the main question here is can some of the prepared data without sacrificing the quality of results. Dimensionality reduction is the process of taking data in a high dimensional space and mapping it into a new space whose dimensionality is much smaller. There are several reasons to reduce the dimensionality of the data. For example, high dimensional data impose computationally efficiency challenge. Moreover, high dimensional data may lead to poor generalization abilities of the learning algorithm (for example, in nearest neighbor classifiers, the sample complexity increases exponentially with the dimension). Figure 1 [6] illustrates the revised process of data mining with an intermediate step for dimension reduction. In this figure, dimension reduction methods are applied to data and then prediction methods are applied to the reduced data.

Figure 1. The role of dimension reduction in data mining [6].
The two aspects of dimension reduction are delete a column (known as feature selection) and delete a row (known as instance reduction). These aspects attempt to preserve the characteristics of the original data by excluding some data. These aspects are different from feature extraction that reduces dimensions but the new data are unrecognizable when compared to the original data i.e., instead of selecting a subset of features from the original set of features (as in feature selection), new blended features are created [6]. In feature extraction, all available features are used and the data are transformed (using a linear or nonlinear transformation) to a reduced dimension space. Thus, the aim is to replace the original features by a smaller set of features [2].

Those techniques that delete features could be considered methods for data preparation. Similarly, methods that transform data into a new set of features can also be considered data preparation methods. The subset of data that results from these deletions or transformations maintains the same number of cases (rows). For most applications, the dataset will have many rows than columns. For example, customers and sales may increase but the number of features per customer may not increase much. Thus, deleting a column has a more dramatic effect on data reduction than deleting a row [6].

Feature selection can be used in many applications from choosing the most important social-economic parameters for determining whoever a person that can return a bank loan to dealing with a chemical process and selecting the best set of ingredients [8]. It is used in many applications in order to simplify their datasets by eliminating the redundant and irrelevant features without affecting the prediction accuracy. Some of these applications are: face recognition [9]-[14], face detection [13], [15]-[16], bioinformatics [17]-[19], web page classification [20]-[21], text categorization [22]-[23], speaker recognition [24], and customer relationship management [25].

III. EVOLUTIONARY ALGORITHMS

According to Shukla et al. [26], looking to any species in the natural world can give a clear idea of evolution. The most complex species are results of continuous evolution over time. Evolution started with simple species. Over time, these species evolved into newer species. In general, each new species is better suited to the changing environments when compared with the earlier ones. This makes it possible for the continuous generation of new species from the older ones to keep performing better. The results are the most adapted species that we find today. Artificial systems keep getting better over time as a result of continuous adjustments and adaptation [26].

Evolutionary algorithms are population based (i.e., a set of individuals representing various candidate solutions to the optimization problem being solved) heuristic search techniques. In FS problems, each individual represents a feature subset. The quality of each candidate solution is evaluated using a fitness function. The selected individuals are subjected to the action of genetic operators to obtain new individuals that constitute the next generation [27].

A. Genetic Algorithms

Genetic algorithms are one of the evolutionary algorithms that have been used in solving feature selection problems. This is the oldest, most established, and well used of the evolutionary algorithms. Hence, its mechanics and its application to feature selection will be described in the following.

Conversion of one generation to the next generation is done by means of genetic operators. The first operator is the selection operator that selects the pairs of individuals that will reproduce to generate newer populations. Selection decides the parents that are responsible for the individuals being generated. Selection is done on the basis of the individual's fitness value. An individual having a higher fitness has a high reproductive capability and is thus likely to be selected numerous times. On the other hand, the weaker individuals may not be selected at all for reproduction [26]. Mutation and crossover are two of the most common operators used with genetic algorithms that represent individuals as binary strings. Mutation operates on a single string and generally changes a bit at random. Thus, a string 11010 may get changed to 11110 by inverting the third bit. Crossover operates on two parent strings to produce two offspring. For example, with a randomly chosen crossover position four, the two strings 01101 and 11000 yield the offspring 01100 and 11001 as a result of crossover. Other genetic representations require the use of appropriately designed genetic operators [27].

The idea in using genetic algorithms for solving optimization problems is that they start with a population of solutions. Each individual in the population represents a solution to the given optimization problem. Initially, the population contains all randomly generated solutions that are initially generated. This is called the first generation of the population. Then the various genetic operators are applied over the population to produce a new population set from the old population set. The goodness of the solution varies from individual to individual within the population. To measure the goodness of a solution, a fitness function is used [26].

1) Genetic Algorithms for Feature Selection

The task of using genetic algorithms for feature selection using binary representation is to find the optimal binary vector (an individual in a population) in which each bit corresponds to a feature. A one or zero means that the feature is selected or not respectively. The aim is to find the binary vector with the smallest number of ones that achieve the best performance [28].

Abbasimehr and Alizadeh [29] used a genetic algorithm for feature selection in churn prediction with extra evaluation criteria besides the model accuracy that is the comprehensibility measure. Where comprehensibility measure is measured as the number of rules extracted from decision tree (their used model). Nahook and Eftekhari [30] solved feature selection based on $\cap$ - fuzzy similarity measures that were found using multi objective genetic algorithm and they tested their work using benchmark datasets from the UCI machine-learning repository [31]. Zhu and Hu [32] adapted a filter method to remove redundancy.
features based on mutual information and then they adopted the wrapper based feature selection method based on a genetic algorithm i.e., the samples are first filtered by conditional mutual information and then sent to the genetic algorithm to be optimized according to the used classifier. They used their algorithm in intrusion detection system. This is illustrated in Figure 2 [32]. Salcedo-Sanz et al. [33] improved the performance of their genetic algorithm for the feature selection problem by using a novel genetic operator that fixes in each iteration the number of features to be selected among the available ones and consequently reduces the size of the search space. This leaded to training became faster and achieving higher performance. They also used the Walsh expansion of the feature selection fitness function in order to perform features ranking. They tested their algorithm using real biological datasets. Vignolo et al. [34] used multi-objective genetic algorithm for face recognition that led on maximizing the face classification accuracy, while minimizing the number of the selected features and the mutual information. Besides, the authors proposed two different strategies for representing the candidate solutions.

![Feature Selection Flow Diagram in the Intrusion Detection based on Improved GA](image)

Figure 2. Feature Selection Flow Diagram in the Intrusion Detection based on Improved GA [32].

2) Research Issues

Research directions in genetic algorithms for solving feature selection can be divided into two main research directions:

- Applying genetic algorithms to new types of datasets. For example, Pali and Bhaiya [35] used a GA for feature selection in face database with the use of neural network classifier. Another example is He et al. [36] that applied a genetic algorithm for feature selection in the field of financial market to find out the most significant factors to the stock market that is very important in this field. As shown from these examples, testing these genetic algorithms was using specific type of datasets that makes testing is not sufficient and make comparisons with them is difficult.

- Hybridizing the used GA with another evolutionary algorithm in order to take advantage of these two approaches and to handle large datasets [1] such as Oh et al. [37] where they embedded local search operations after devising them into a GA in order to fine-tune the search. This leaded to a significant improvement in the performance and the acquisition of subset size control. They performed the experiments on standard datasets.

B. Ant Colony Optimization

The ant colony (rather than individual ants) can be seen as an intelligent entity for its great level of self-organization and the complexity of the tasks it performs. Natural ant colony systems inspired many researchers in computer sciences to develop new solutions for optimization problems [38].

ACO algorithms simulate the foraging behaviour of some ant species [39]. They use two factors for guiding the search process. These are: the pheromone values and heuristic information. Good-quality solutions can only emerge as the result of the collective interaction between the artificial ants. This is obtained via indirect communication mediated by the information ants read or write in the variables storing pheromone trail values. This is a distributed learning process in which the single ants are not adaptive themselves but adaptively modify the way the problem is represented and perceived by other ants [1].

The early applications of ACO have been mainly concerned with solving ordering problems. One of the recent trends in ACO is to solve industrial problems proving that it is useful for real-world applications [40]-[41]. More recent applications include for example bioinformatics, multi-objective, and dynamic problems [42].

1) Ant Colony Optimization for Feature Selection

ACO algorithms for feature selection differ mainly in the solutions approaches for representing it such as whether solving it using the constructive graph or not, using one ACO algorithm or more, suitable pheromone representation, or dealing with it as a single or multi-objective optimization problem. This is explained bellow.

Much of the ACO literature considers the development of a constructive graph for the problem to be solved as essential to the application of ACO as it is a graph based shortest path problem that ants solve when moving from the nest to a food source [43].

It is evident that ACO has been applied to numerous problems that are not naturally described in terms of graphs. One example is in Lee et al. [44] where they adopted the graph based ant system to solve feature selection where candidate solutions can be represented in directed graphs and each traversed path by an ant in a cycle represents a candidate solution to the feature selection. The selected features are represented as a combination of arcs where ants have traversed through the graph. Every ant must visit every node at most once. Every ant starts at a specific node and ends at a specific node checking every node in between with a given sequence. Every node has two arcs connected to its next visiting node, each representing either selection or exclusion of the feature it is assigned to. Therefore, combining traversed arcs together gives a full representation of a candidate solution. This is illustrated in Figure 3 [44].

Another example is in Bello et al. [45] where a feature selection problem is viewed as a network in which nodes represent features and all nodes are connected by bidirectional links. Pheromones are associated with nodes. Ants perform a forward selection in which each ant expands its subset step-by-step by adding new features. Jensen and
Shen [46] represented feature selection as a graph where the nodes represent features and the edges between the nodes denote the choice of the next feature. The search for the optimal feature subset is then an ant traversal through the graph where a minimum number of nodes are visited that satisfies the traversal stopping criterion. Figure 4 [46] illustrates this process. Similarly, Aghdam et al. [47] follow similar strategy to Jensen and Shen [46] but the pheromone is associated with the nodes instead of with the edges. Each ant starts with a random feature. From these initial positions, they traverse edges probabilistically until a traversal stopping criterion is satisfied. Abd-Alsabour et al. [52] studied the effect of fixing the length of the selected feature subsets. This contradicts with the main concepts of ant algorithms as heuristic algorithms. Abd-Alsabour and Randall [48] did not use the constructive graph. Instead, they implemented a binary ant colony system with the use of support vector machine classifier. The algorithm was tested using benchmark datasets from the UCI repository [31].

![Figure 3](image3.png)

**Figure 3.** FS using an ACO algorithm with DistAl [44].

![Figure 4](image4.png)

**Figure 4.** ACO problem representation for FS [46].

As a subset problem, solutions for feature selection problems do not have fixed length i.e., different ants may have solutions of different lengths. However, some literature (such as Nemati et al. [49], Sivagaminathan and Ramakrishnan [50], and Gunthuri et al. [51]) solved feature selection problems using ant algorithms by fixing the length of the selected feature subsets. This contradicts with the main concepts of ant algorithms as heuristic algorithms. Abd-Alsabour et al. [52] studied the effect of fixing the length of the selected feature subsets on the performance of ACO algorithms for feature selection. They showed that fixing the length of the selected feature subsets will affect the ability of ACO algorithms to converge to the optimal solution (that guarantees that there always remains a possibility to escape from sub-optimal solutions) even if this fixed length equals to the best length.

One more point in this issue is about where the pheromone should be related with (suitable pheromone representation). According to Montgomery [43], two main pheromones strategies may be considered for solving subset selection problems using ACO algorithms. These are: 1) Associate a pheromone trail $\tau_i$ with each object $i$ so that $\tau_i$ represents the desirability of selecting object $i$, and 2) Associate pheromone trail $\tau_{ij}$ with each pair of different objects so that $\tau_{ij}$ representing the desirability that objects $i$ and $j$ belong to the same subset. According to Leguizamón and Michalewicz [53], in ordering problems, the pheromone is laid on paths while for subset problems no paths exist connecting the items. Therefore, the idea of "the more pheromone trail on a particular path, the more profitable is that path" is adapted to "the more pheromone trail on a particular item, the more profitable that item is" i.e., the pheromone is put on items not paths. A value should be assigned to each element without any considerations about possible connections between them (ordering is not important any longer) i.e., items only should be considered instead of connections between them [53].

The last point in this issue is about the evaluation approaches used for evaluating the feature subsets. The main approaches can be classified into filter or wrapper approaches, depending on whether or not feature selection is done independently of the classifier [1]. Normally most of literature on ACO for feature selection uses either wrapper or filter approach as an evaluation criterion but Al-Ani [54] follows a hybrid evaluation measure that is able to estimate the overall performance of subsets as well as the local importance of features. A classification algorithm is used to estimate the performance of the subsets (wrapper evaluation function). On the other hand, the local importance of a given feature is measured using the mutual information evaluation function (which is a filter evaluation function).

2) **Research Issues**

While combining ACO with other evolutionary algorithms has been performed previously, hybridization between ACO and other evolutionary algorithms is still the topic of recent research. An example is the work of Rami et al. [55] where they combined an ACO algorithm with differential evolution (DE) after adapting it to deal with discrete problems. The DE optimization technique can be viewed as an enhanced version of the real valued GA that employs a differential mutation operator with faster convergence properties [56]-[57]. More comparisons and extra feature selection domains are required since the authors reported only comparisons with two evolutionary algorithms and used only one domain that is bio-signal driven applications. The question arise here is if there are ACO implementations (without hybridization) that are able to solve feature selection and get good results, why going to such a research direction that leads to adding extra computation. According to El-Sawy et al. [58], in recent years it has become evident that the concentration on a sole metaheuristic is rather restrictive. A skilled combination of meta-heuristic with other optimization techniques, hybrid meta-heuristic, can provide a more efficient behavior and a higher flexibility when dealing with real-world and large scale problems. However, the field of hybrid meta-heuristics is still in its early days [58] and a substantial amount of further research is necessary in order to show the benefits against the cost of such hybridization and the real role the added evolutionary algorithm will play to the host one. For example, Sun and Li [59] added a GA to an ACO algorithm.
for feature selection just to determine the distribution of pheromones on the paths.

C. Particle Swarm Optimization

PSO was inspired by the social behavior of bird flocking and fish schooling by Eberhart and Kennedy in 1995 [60]. It utilizes a population of particles to represent potential solutions within the search space that fly through a multi-dimensional search space with given velocities. Its position, velocity, and a record of its past performance characterize each particle. Each particle encodes a single intersection of all search dimensions. The associated position and velocity of each particle are randomly generated. At each generation, the velocity of the particle is stochastically adjusted according to the historical best position for the particle itself and the neighborhood best position, i.e., to discover the optimal solution, each particle changes its searching direction according to two factors; its own best previous experience (pbest- called the cognition part) and the best experience of all other members (gbest- called the social part) [61]. This is accomplished by the fitness function [62]. Their leaders are the best performers either from the entire swarm or from their neighborhood influence particles. At each flight cycle, the objective function is evaluated for each particle (with respect to its current position) in order to measure the quality of the particle and to determine the leader in the sub-swarms and the entire population [62]-[63].

Variants of PSO have been proposed such as canonical particle swarm [64] and fully informed particle swarm [65]. Not all of them have been applied for feature selection yet.

1) Particle Swarm Optimization for Feature Selection

Wahono and Suryana [66] used a combination of PSO and a bagging technique for improving the prediction accuracy. PSO was applied to deal with the feature selection and bagging technique was employed to deal with the class imbalance problem. The proposed method was evaluated using datasets from NASA metric data repository. Bagging is an ensemble technique where many classifiers are built and the final classification decision is made based on some forms of voting of the committee of classifiers. It is used in order to improve the classification accuracy [67]. Pan et al. [68] used PSO to replace the exhaustive search used in the Adaboost (the used classifier) on CMU+MIT frontal face dataset. Chuang et al. [69] proposed a variant of PSO called complementary PSO (CPSO) algorithm with the use of K-Nearest Neighbor classifier in the field of DNA Micro-array data. Zahran and Kanaan [70] implemented a binary PSO for feature selection for Arabic text categorization. Jacob and Vishwanath [71] proposed multi-objective PSO for face recognition that outperformed a multi-objective GA in the authors' experiments. Abdi et al. [72] used a PSO to form a novel weighted support vector machine (the used classifier) for gene selection and tumor classification problems. Where PSO not only discards redundant genes, but also takes into account the degree of importance of each gene and assigns diverse weights to the different genes. It was also used to find the appropriate kernel parameters since the choice of gene weights influences the optimal kernel parameters. Yan et al. [73] proposed a new discrete PSO algorithm with a multiplicative likeliness enhancement rule for unordered feature selection for face recognition. They tested their algorithm on the FERET database. Features are selected by their assigned likeliness, which is enhanced by the agreement between a particle and its attractors (its previous location, pbest and gbest). The multiplicative updating rule achieves higher fitness and smaller standard deviation than the additive likeliness enhancement rule. Revathi1 and Malathi [74] proposed a novel simplified swarm optimization algorithm as a rule-based classifier and for feature selection to classify intrusion data. Sivakumar and Chandrasekar [75] used a modified continuous PSO for feature selection in wrapper-based method with k-nearest neighbor classifier that served as a fitness function for the PSO. This algorithm was tested using Lung CT scan images.

2) Research Issues

Besides developing new variants of PSO to suit feature selection as explained in the previous section, integrating PSO with other evolutionary algorithms is the topic of its recent trends as with GAs and ACO. For example, Kumar [76] combined a PSO with a GA to produce good classification rules by which the used intrusion detection system detected the intrusion with very low false alarm rate and high detection accuracy rate. Another example is the work of Nazir et al. [77] that combined a PSO and a GA to perform together feature selection for gender classification using real-world face images.

IV. CONCLUSION AND FUTURE WORK

This paper describes the best-known evolutionary techniques for feature selection in a relative detail. We should remark that the used list of references is not a comprehensive list of papers discussing these algorithms. Rather, our aim was to produce a critical review of the key ideas and guide the researchers in interesting research directions in this area.

The endless inspiration from nature will be an ongoing source for many new optimization approaches that may be applied to feature selection. Therefore, as a research direction for future work in this area, more evolutionary algorithms for feature selection should be added to this review. Also, comparisons between all of the explained evolutionary algorithms should be conducted.

REFERENCES
