A Survey on Evolutionary Multiobjective Feature Selection in Classification: Approaches, Applications, and Challenges

Article *in* IEEE Transactions on Evolutionary Computation · January 2023

DOI: 10.1109/TEVC.2023.3292527

Ruwang Jiao
Soochow University
26 PUBLICATIONS **629** CITATIONS

SEE PROFILE

Bach Nguyen
Victoria University of Wellington
41 PUBLICATIONS **1,059** CITATIONS

SEE PROFILE

Bing Xue
Victoria University of Wellington
443 PUBLICATIONS **15,422** CITATIONS

SEE PROFILE

Mengjie Zhang
Victoria University of Wellington
997 PUBLICATIONS **27,520** CITATIONS

SEE PROFILE

A Survey on Evolutionary Multiobjective Feature Selection in Classification: Approaches, Applications, and Challenges

Ruwang Jiao, *Member, IEEE*, Bach Hoai Nguyen, *Member, IEEE*, Bing Xue, *Senior Member, IEEE*, and Mengjie Zhang, *Fellow, IEEE*

***Abstract*—Maximizing the classification accuracy and minimizing the number of selected features are two primary objectives in feature selection, which is inherently a multiobjective task. Multiobjective feature selection enables us to gain various insights from complex data in addition to dimensionality reduction and improved accuracy, which has attracted increasing attention from researchers and practitioners. Over the past two decades, significant advancements in multiobjective feature selection in classification have been achieved in both the methodologies and applications, but have not been well summarized and discussed. To fill this gap, this paper presents a broad survey on existing research on multiobjective feature selection in classification, focusing on up-to-date approaches, applications, current challenges, and future directions. To be specific, we categorize multiobjective feature selection in classification on the basis of different criteria, and provide detailed descriptions of representative methods in each category. Additionally, we summarize a list of successful real-world applications of multiobjective feature selection from different domains, to exemplify their significant practical value and demonstrate their abilities in providing a set of trade-off feature subsets to meet different requirements of decision makers. We also discuss key challenges and shed lights on emerging directions for future developments of multiobjective feature selection.**

***Index Terms*—Evolutionary feature selection, multiobjective learning, classification**

I. Introduction

With the proliferation of big data in today's world, high-dimensional data is commonly encountered in many real-world applications. For example, in cancer classification, a doctor has to decide whether a patient has cancer or not, and a wrong decision may frighten or even kill a person. However, the cancer data is usually high-dimensional and contains a large number of irrelevant, noisy, and/or redundant features [1], which significantly affects the classification performance. In specific, irrelevant or noisy features often degenerate the classification performance due to their misleading information that confuses a classification algorithm. Redundant features cannot improve the classification performance, but will result in longer computational time, because they provide the same information with respect to the class labels. The high-dimensional data also significantly increases the storage space and processing time, and makes machine learning models less comprehensible. In addition, due to the "curse of dimensionality", a machine learning method requires a very large number of training instances to achieve a reliable result. Otherwise, it is easy to be overfitting the machine learning model.

Feature selection (FS) addresses the above problems by selecting a small subset of relevant features which can improve the classification performance, reduce the dimensionality of data, reduce space storage, improve computational efficiency, and facilitate data visualization and understanding [2], [3]. FS plays a critical role in data mining, pattern recognition, and machine learning. Compared with other dimensionality reduction techniques, such as feature construction and feature extraction [4], FS can preserve the original semantics of the data, making it an effective method with interpretability and facilitating human understanding of the results.

Although FS is an essential process, it is also a complex and challenging combinatorial optimization problem. The challenges of FS lie in three perspectives. First, the search space of FS grows exponentially with the number of features, i.e., D features can result in 2^D possible feature subsets. Thus, the exhaustive search that considers all the possible subsets is impractical, particularly when the number of features is large. Second, there are complex interactions among features. For example, two relevant features that contain similar information with respect to a particular class label may lead to redundancy. By contrast, two weakly relevant features can provide significant information regarding the classification task when they are selected together, which is known as complementary features. Third, FS is inherently a multiobjective problem. The two main goals of FS are to maximize the classification performance and minimize the number of selected features. However, these two objectives are usually in conflict. For example, removing relevant and/or complementary features can deteriorate classification performance. There is no single best feature subset, but rather a set of non-dominated subsets showing trade-offs between the two objectives. Optimizing the two objectives can more accurately reflect the decision-making reality of FS problems in practical applications.

A powerful search algorithm is the basis for dealing with the above three challenges in multiobjective FS (MOFS) problems. Thanks to the numerous strengths, such as the global search ability, no prior knowledge is required about the problem, and the population-based search can obtain a set of solutions to trade-off the multiple conflicting objectives, evolutionary computation (EC) has been widely used to ad-

The authors are with the School of Engineering and Computer Science, Victoria University of Wellington, Wellington 6140, New Zealand (E-mail: ruwangjiao@gmail.com; hoai.bach.nguyen@ecs.vuw.ac.nz; bing.xue@ecs.vuw.ac.nz; mengjie.zhang@ecs.vuw.ac.nz).

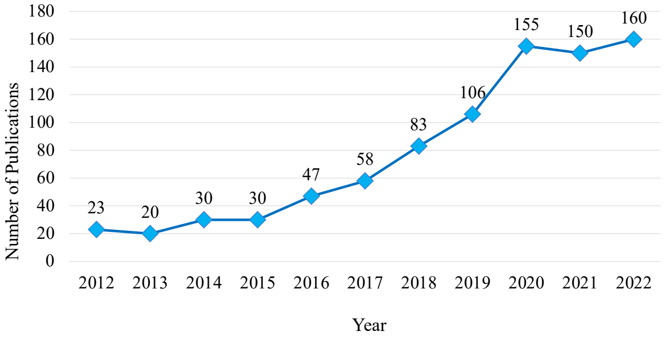


Fig. 1. Number of publications on MOFS in classification from 2012 to 2022. The data is collected from the Web of Science on January 6, 2023, with the topics of "multi-objective feature selection" OR "multiobjective feature selection" AND "classification". The total number of publications from 2012 to 2022 is 862.

dress multiobjective optimization problems, and the derived algorithms are called evolutionary multiobjective optimization (EMO) methods [5]. EMO can also easily handle multiobjective optimization problems with discontinuous and concave Pareto fronts, which are two salient strengths over traditional mathematical programming methods [6]. There have been a lot of research endeavors to use EMO for addressing MOFS problems. As shown in Fig. 1, the number of publications related to MOFS in classification is increasing rapidly in the last decade.

A. *Difference from other surveys*

Aiming to facilitate getting updated with the latest research status and relevant results of MOFS in classification, this paper provides a review of the most relevant evolutionary MOFS studies. In spite of numerous studies on evolutionary MOFS in classification have been conducted, very few surveys are available. Previous surveys focus mainly on evolutionary single-objective FS [7]–[10], where evolutionary MOFS has been touched slightly. In [11], EMO for data mining tasks, such as FS, classification, clustering, and association rule mining have been reviewed. However, only a small portion of the survey is devoted to MOFS. Al-Tashi *et al.* [12] present the most recent survey on MOFS in 2020. However, it focuses mainly on wrapper MOFS approaches, while our survey paper covers a wider range including wrapper, filter, and embedded approaches. To the best of our knowledge, this is the first review of MOFS that widely presents how the studies in MOFS have evolved and the main topics associated with this research field.

B. *Insights*

The overall goal of this survey is to provide a comprehensive survey of the state-of-the-art in MOFS in classification and highlight its advances, gaps, and challenges, and provide a guideline for relevant researchers and practitioners in this active realm, with the expectation to stimulate some new ideas for promoting the development of MOFS. The insights that readers can extract from this paper include:

1) The development of the mainstream methods of existing MOFS studies in the recent two decades. This part provides an in-depth understanding to this research field.
2) The progress of MOFS research in practical applications. This part points out the direction of the specific application of MOFS for relevant practitioners.
3) The current challenges of the MOFS task and how these challenges can be addressed, or partially addressed, and the possible future directions that can be done.

C. *Outline of the survey*

The rest of the paper is organized as follows. Section II presents the preliminaries of the survey. Section III reviews existing MOFS approaches according to their core design components, including solution representations, evaluation functions, initialization mechanisms, offspring generation, environmental selection, and decision making. Section IV describes key applications of MOFS in classification. Section V elaborates the issues and challenges and identifies the emerging research topics. Finally, Section VI gives some concluding remarks.

II. Preliminaries of the Survey

A. *Multiobjective optimization*

In recent years, multiobjective optimization using metaheuristics has gained great popularity [5], [6]. A multiobjective optimization problem could be mathematically expressed as:

$$\begin{array}{ll} \min & \boldsymbol{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_m(\boldsymbol{x}))^T \\ \text{where} & \boldsymbol{x} = (x_1, \cdots, x_D)^T \in \Omega \end{array} \tag{1}$$

where Ω represents the search space. m is the number of objectives, and D indicates the dimension of a candidate solution $\boldsymbol{x}$. T means transpose.

In multiobjective optimization, the multiple objectives normally conflict with each other, which means any improvement of one objective usually results in deterioration in the other objectives. In other words, there is always trade-off between different objectives, and thus there is no single best solution. The Pareto dominance relationship is usually utilized to compare the quality of each pair of solutions:

Definition 1 (Pareto dominance relation): For any two vectors $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$, $\boldsymbol{x}_a$ can be said to dominate $\boldsymbol{x}_b$ (i.e., $\boldsymbol{F}(\boldsymbol{x}_a) \prec \boldsymbol{F}(\boldsymbol{x}_b)$) if and only if the following two conditions hold:

1) $f_i(\boldsymbol{x}_a) \leq f_i(\boldsymbol{x}_b), \forall i \in \{1, \cdots, m\}$;
2) $f_i(\boldsymbol{x}_a) < f_i(\boldsymbol{x}_b), \exists i \in \{1, \cdots, m\}$.

Definition 2 (Pareto optimum): A solution $\boldsymbol{x}^* \in \Omega$ is a Pareto optimum it is not dominated by any other solution.

Definition 3 (Pareto set & Pareto front): The set of all the Pareto optimums is called the Pareto set:

$$PS = \{\boldsymbol{x} | \boldsymbol{x} \in \Omega \text{ and } \boldsymbol{x} \text{ is Pareto optimum}\}.$$

The mapping of the Pareto set in the objective space is called the Pareto front:

$$PF = \{\boldsymbol{F}(\boldsymbol{x}) | \boldsymbol{x} \in PS\}.$$

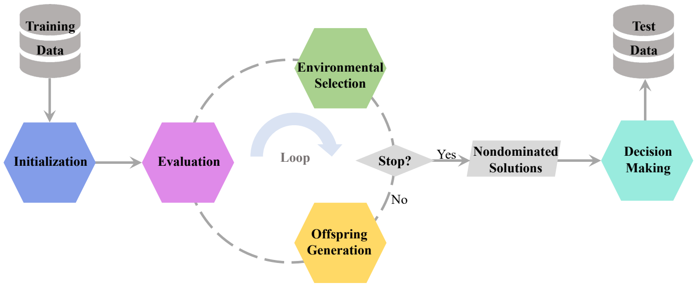


Fig. 2. Framework of MOFS.

B. MOFS

FS in classification is inherently a multiobjective task. Commonly used objectives include maximizing classification accuracy (which can be represented in terms of classification accuracy, classification error, $F1$ measure, or mutual information) and minimizing the size of the feature subset (which can be expressed in terms of the number of selected features or redundancy), as well as some other objectives such as computational cost, model size, etc (more details can be seen in Section III-B). MOFS has the ability to provide a set of non-dominated feature subsets. Practitioners can select their preferred solution from these non-dominated subsets according to their requirements, or utilize different trade-off solutions for different situations.

Over the past two decades, research on MOFS in classification has emerged and increased to explore the consideration of multiple criteria to select features. Traditional methods usually use weight vectors to combine these multiple criteria into a single scalar function to form a single-objective FS [7]. However, such single-objective FS has two main disadvantages: 1) It is not trivial to set an appropriate weight vector to combine the different objectives since their trade-off is problem-dependent; 2) Only one feature subset is obtained, from which little insight into the FS problem could be gained. In contrast, the advantages of regarding FS as a multiobjective optimization problem are manifold: 1) a set of non-dominated feature subsets can be obtained to meet different requirements in real-world applications; 2) by analyzing the Pareto front composed of multiple non-dominated feature subsets, we can understand the FS problem more deeply [13]; and 3) avoid the hyperparameter-setting trouble, i.e., no need to set a weight vector to trade off the multiple objectives.

It is worth noting that deep learning has shown remarkable success in various areas [14]. However, in some aspects, MOFS methods still have their salient strengths over deep learning:

- **Improved interpretability:** Deep learning models are often considered black boxes since it is usually difficult to understand how the deep model arrives at its predictions. By contrast, MOFS can select a small set of features without sacrificing the valuable insights contained within the features, which potentially leads to more interpretable models. Actually, FS has been used to improve the interpretability of deep learning [15]. Furthermore, MOFS facilitates gaining insight of the problem, e.g., interactions between features, which is essential in many real-world domains such as medical [1].
- **Computationally efficient:** While MOFS methods can be more expensive when the number of features/instances is large, they are still relatively cheaper than most deep learning approaches since MOFS methods usually do not require as much data as deep learning methods to obtain reliable results. Furthermore, MOFS methods often do not require as many computational resources or expensive GPUs, making them more accessible and affordable to many researchers and practitioners.

III. MOFS APPROACHES

As shown in Fig. 2, an MOFS framework usually consists of initialization, objective evaluation, environmental selection, offspring generation, and decision making. Note that the evaluation step includes defining the solution representation and evaluation functions. Therefore, as depicted in Fig. 3, this section categorizes and reviews existing MOFS work according to the following six core design components: solution representation, evaluation functions, initialization, offspring generation, environmental selection, and decision making.

A. Solution representation

A suitable data structure that can represent solutions (i.e., feature subsets) to the MOFS problem is the first step in designing a working MOFS method. Based on the existing MOFS research, solution representation can be divided into four categories: vector, tree, graph, and matrix. In what follows, we detail them accordingly.

1) Vector: The vector-based representation is most commonly used in the evolutionary FS community, which can be further grouped into two categories: binary encoding and continuous encoding, as illustrated in Fig. 4(a). In binary encoding, the bits 1 and 0 represent a feature that is selected or discarded, respectively. In continuous encoding, a threshold θ rounds the continuous representation value to 1 and 0 to determine whether the corresponding feature is selected or not. Note that this encoding is often used in EC methods that are based on vector representations, e.g., genetic algorithms (GAs) [16], particle swarm optimization (PSO) [17], differential evolution (DE) [18], and so on.

The classical vector-based representation requires one dimension/bit for each feature. If the number of features is very large, a large number of bits are required to represent a solution (feature subset). Moreover, since a feature subset has a lot of 0 bits (i.e., features are not selected), it is a waste for the classical vector-based representation. To this end, a new vector-based representation [19] is suggested which only stores the index of selected features, to save memory for feature subset representation and speed up the evaluations. To solve high-dimensional MOFS problems, the idea of cooperative coevolution from evolutionary large-scale optimization [20] is borrowed, which decomposes the original large-scale solution representation into a set of smaller solution representations through a clustering operator, aiming to address the high-dimensional MOFS problems via a divide-and-conquer manner [21].

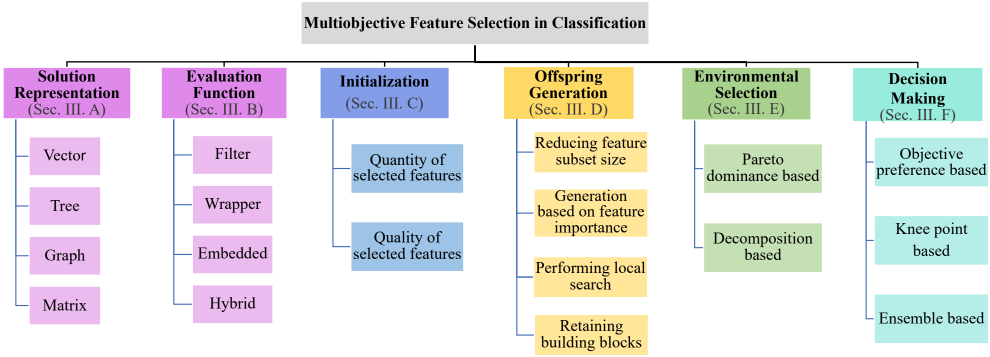
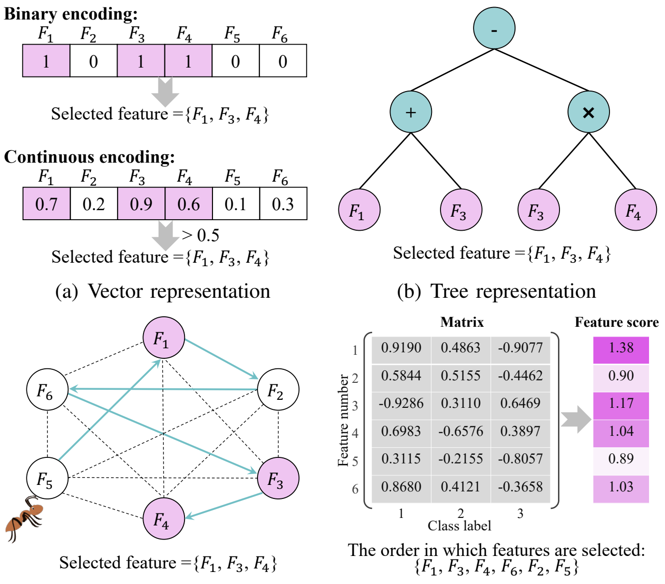

Multiobjective Feature Selection in Classification
Solution Representation (Sec. III. A)
Vector
Tree
Graph
Matrix
Evaluation Function (Sec. III. B)
Filter
Wrapper
Embedded
Hybrid
Initialization (Sec. III. C)
Quantity of selected features
Quality of selected features
Offspring Generation (Sec. III. D)
Reducing feature subset size
Generation based on feature importance
Performing local search
Retaining building blocks
Environmental Selection (Sec. III. E)
Pareto dominance based
Decomposition based
Decision Making (Sec. III. F)
Objective preference based
Knee point based
Ensemble based

Fig. 3. Taxonomy of MOFS according to different criteria.


Binary encoding:
F_1 F_2 F_3 F_4 F_5 F_6
1 0 1 1 0 0
Selected feature $=\{F_1, F_3, F_4\}$
Continuous encoding:
F_1 F_2 F_3 F_4 F_5 F_6
0.7 0.2 0.9 0.6 0.1 0.3
> 0.5
Selected feature $=\{F_1, F_3, F_4\}$
-
+
×
F_1
F_3
F_3
F_4
Selected feature $=\{F_1, F_3, F_4\}$
F_1
F_2
F_3
F_4
F_5
F_6
Selected feature $=\{F_1, F_3, F_4\}$
Matrix
Feature score
Feature number
Class label
The order in which features are selected: $\{F_1, F_3, F_4, F_6, F_2, F_5\}$

(a) Vector representation

(b) Tree representation

Feature number	1	2	3	Feature score
1	0.9190	0.4863	-0.9077	1.38
2	0.5844	0.5155	-0.4462	0.90
3	-0.9286	0.3110	0.6469	1.17
4	0.6983	-0.6576	0.3897	1.04
5	0.3115	-0.2155	-0.8057	0.89
6	0.8680	0.4121	-0.3658	1.03

(c) Graph representation

(d) Matrix representation

Fig. 4. Illustrations of different solution representations.

2) Tree: The tree-based genetic programming (GP) [22]–[24] is the representative method using the tree representation. The built-in ability of GP in detecting important features and implicitly and automatically mining the complex feature interactions by exploring the feature space has made it a valuable method for FS. GP uses a set of functions to represent internal nodes and a set of terminals to represent leaf nodes. An illustrative example is depicted in Fig. 4(b), from which we can figure out that FS can be realized by selecting only the features in the leaf nodes of the GP tree, i.e., features F_1, F_3, and F_4 in Fig. 4(b). In addition to that, the tree-based GP also has the ability to construct new features based on the selected important features, e.g., F_1+F_3, F_3*F_4, and $F_1+F_3-F_3*F_4$ in Fig. 4(b), to discover hidden relationships between features.

3) Graph: In ant colony optimization (ACO) [25], the representation of the solution is usually in the form of a graph. For instance, a decision graph corresponds to the heuristic information and pheromone density is used to represent a feature subset [26]. As shown in Fig. 4(c), suppose an ant selects a subnode (i.e., feature F_5 in Fig. 4(c)) of the graph as a starting point, and then travels to other features. Ants should decide whether to select this feature when traversing a node, depending on a probability function consisting of the pheromone intensity and the heuristic information on the edges. Heuristic information is mainly measured by relevance (features to labels) and redundancy (features to features). It should be noted that a feature can only be visited once by an ant. In Fig. 4(c), an ant travels nodes in the order of F_5, F_1, F_2, F_6, F_3, F_4, but only features F_1, F_3, and F_4 are selected under the assumption that their probability function values are large.

In [27], a solution is represented via a weighted graph $< F, E >$, where the node set $F = \{F_1, F_2, \cdots, F_D\}$ denotes all the features in a dataset, and $E = \{(F_i, F_j) : F_i, F_j \in F\}$ represents the edge set of the graph. A weight w_{ij} is used to indicate the correlation between two features F_i and F_j that are linked by the edge (F_i, F_j). Edges with values less than a threshold are removed.

In [28], a feature graph is defined for MOFS that considers both the feature importance and the complex relationship between features. FS is achieved by considering the links of the feature graph. To obtain better feature graph structures, an EMO algorithm is used to dynamically adjust the structure of the feature graph, which considers the classification accuracy and feature subset size of the adjusted feature graph as two objectives.

4) Matrix: In sparse learning-based FS methods, a transformation matrix is used to fit a sparse learning model [29], which can be represented as a solution. Each row of the transformation matrix represents a feature. After the transformation matrix is obtained, a composite score is calculated for each row of the transformation matrix, i.e., $||\mathbf{w}^i||_2^2$ $(i = 1, \cdots, D)$, and then features with the highest feature scores are selected [30]. Fig. 4(d) gives a schematic illustration with six features, where the order of FS based on their composite scores is F_1, F_3, F_4, F_6, F_2 and F_5.

Summary: The vector-based encoding is straightforward for MOFS since each bit corresponds to a raw feature. Nevertheless, it cannot scale well when applied to high-dimensional datasets because the search space grows exponentially. Besides, it can show which features are selected, but not the interactions between features, i.e., which features can work well with others. For the tree-based encoding, mainly GP, the built-in FS and the ability to achieve both FS and feature construction make it salient among other representations. However, the search space of GP is even larger due to the need of searching both features and function operators, and may only select a very small number of features when limiting the search space (i.e., tree depth). The graph-based encoding (e.g., ACO) is usually based on a fully connected graph, which leads to a large and complex search space, because at each node, an ant has many choices to traverse. For the matrix-based solution representation (i.e., sparse learning-based MOFS), the major challenges posed for EMO are the huge search space and the long computation time, since each matrix consists of $D \times c$ variables that need to be optimized, where D and c represent the number of features and classes, respectively. It is worth noting that there are some evolutionary matrix-based operations that may offset some of the above difficulties of slow computation time. For example, in [31], a population is encoded as a matrix, and then the parallel computing functionalities of the matrix can be directly and easily executed on high-performance computing resources to accelerate the computation speed of evolutionary operators.

There is no thumb rule to choose the most appropriate solution representation for MOFS. The choice of solution representation is highly related to the EC method used, since the offspring generation operations used to produce new solutions are based on the solution representation method.

B. Evaluation functions

As shown in Fig. 5, based on how much a classification algorithm is involved during the FS process, MOFS could be roughly categorized into the filter, wrapper, and embedded methods [2]. In different kinds of MOFS methods, different criteria are used to define evaluation functions.

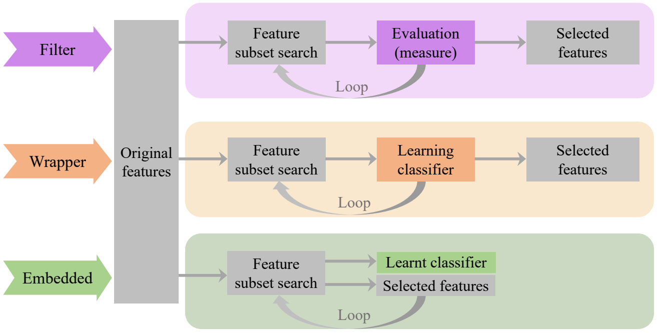


Fig. 5. Taxonomy of MOFS methods according to how much a classification algorithm is involved during the FS process.

1) Filter methods: They usually evaluate the goodness of a feature subset based on the intrinsic characteristics of the data, such as information-theoretic measures, correlation measures, similarity measures, consistency measures, statistical measures, and fuzzy set theory measures. They are usually computationally efficient and have good generalization ability to different classification algorithms, but could have worse classification performance than wrapper methods because their selected features are not integrated with a classification algorithm.

Since FS aims to select a compact feature subset with maximal discriminative ability, where the discriminability requires that the selected features have high relevance to class labels, and the compactness requires that the selected features have low redundancy, so relevance and redundancy are the most common evaluation criteria in filter methods. Traditional methods normally aggregate these criteria into a single-objective metric function, e.g., minimal redundancy maximal relevance (mRMR) [32], and then use a greedy search to select the best feature subset. Nevertheless, as discussed before, it is not a trivial task to set an appropriate weight value to combine these two criteria, and the greedy search mechanism easily renders the solution to fall into local optima.

A large family of evaluation functions in existing filter MOFS methods is based on *information-theoretic measures*, [2], including the entropy principle. Entropy principle is used to quantify the amount of information in a data source and is a useful tool of uncertainty measure for characterizing the distinguishing information of feature subsets. In such methods, the two separate objectives, i.e., the relevance between features to class labels and the redundancy between features, are measured by using some information-theoretic criteria, e.g., mutual information [33], [34]. Mutual information is a measure of the mutual dependence between two random variables, which is obtained through the product of the joint distribution of the two variables and their marginal distributions. It can be used to measure the shared information between a feature to the class label (i.e., relevance) or two features (i.e., redundancy). However, it is intractable to determine the number of selected features by only using information-theoretic methods to measure feature relevance. To automatically determine the optimal number of selected features, maximizing the relevance between the selected feature subset with respect to class labels and minimizing the number of selected features can be regarded as two objectives [35]. In the above methods, mutual information and entropy are used to calculate the feature relevance.

In *similarity*-based filter methods, the feature importance is assessed by data similarity. In this method, relevance and redundancy can be measured using some similarity-based criteria. For example, the Pearson correlation coefficient can be used to measure the redundancy among features, while the relative discrimination criterion [36] or separability index [37] is used to measure the relevance between each feature with respect to class labels. Distance is often used as a straightforward way to measure data similarity. For example, good features facilitate data belonging to different classes to be further apart, while data belonging to the same class are as close as possible, which is undoubtedly an MOFS task, i.e., maximizing the inter-class distance and minimizing the intra-class distance [38]. In [39], in addition to objectives of distance and feature subset size, a new objective of region purity is designed, to retain as many instances of the same class as possible in the local region while excluding instances of different classes that tend to cause misclassification.

Different filter measures have different characteristics and preferences. To study the complementary relationship between different combinations of filter measures belonging to different categories, Spolaôr *et al.* [40] combine the class separability measure (i.e., intra-class distance) with four other filter measures of consistency (i.e., inconsistent example pairs), dependency (i.e., attribute class correlation), distance (i.e., Laplacian score), and information (i.e., representation entropy) as two objective functions that are optimized by NSGA-II [41]. These four methods are compared with methods that treat each filter measure as a single-objective FS problem. Experimental results show that the method treats class separability and dependency measures as two objectives that can achieve better prediction performance and select fewer features.

2) Wrapper methods: They evaluate the quality of candidate feature subsets based on a classification algorithm. They can yield high classification accuracy for a particular classification algorithm often at the cost of high computational time and weaker generalization of the selected features to other classification algorithms.

Maximizing the classification performance and minimizing the number of selected features are the two major objectives in FS, which are commonly used in wrapper-based MOFS [42]. Although in wrapper methods, the classification performance (e.g., classification accuracy or classification error rate) of a feature subset is obtained based on a classification algorithm, in different situations, the classification performance can be represented by different evaluation functions. For example, since many collected data are unbalanced, using the overall classification accuracy as the evaluation function will make the classification results of an MOFS method strongly biased toward the majority class. To address the class imbalance problem, the balanced classification error rate [43]–[45] can be used to evaluate the classification performance of a feature subset.

Except for the above two main objectives, other objectives could also be considered in wrapper-based MOFS. For instance, features such as gender or ethnicity can be seen as sensitive features. Since algorithms that learn from biased data tend to produce biased classifiers, the fairness of decisions made by automated processes comes even more important. Therefore, in addition to considering classification performance, four measures of fairness are aggregated into a second objective to address the issue of fairness [46]. Data is not free in practice, because costs such as money and time are required when acquiring feature values. In this case, such an MOFS task should also minimize the cost of the feature subset as an additional objective [47]–[49]. A tri-objective feature selection method [50] considers the difference between feature subsets in the search space as the third objective, which aims to select feature subsets with low similarities with each other in the population. The selection based on the converted three objectives considers the performance of a feature subset in both the objective and search spaces, and can maintain a trade-off between minimizing the number of selected features, maximizing the classification performance, and keeping the diversity of feature subsets.

Datasets in real-world scenarios might contain noisy data, e.g., erroneous, missing, unknown, and incomplete feature values [51]. In addition to classification performance, data reliability is particularly important for noisy datasets. When encountering a classification dataset with missing data, a tri-objective optimization problem can be formulated by considering classification performance, feature subset size, and missing rate as three objectives [52], to consider the reliability objective when searching for non-dominated feature subsets. Sometimes, the degree of reliability of features can be obtained explicitly. In this way, maximizing the classification performance and reliability of data can be taken as a bi-objective optimization problem, to promote the reliability of the selected features [53].

3) Embedded methods: They incorporate FS and model training into a single process, which usually delivers better classification performance than filter methods and is more efficient than wrapper methods. However, embedded methods generally have weaker generalization ability than filter methods and worse classification performance than wrapper methods.

GP has built-in FS and feature construction abilities, and the evolved GP tree can also be used as a classifier [54], which makes GP a mainstream embedded method. When using GP for classification, in addition to the classification performance of the GP tree, we should sometimes consider the tree size (e.g., number of nodes), the complexity (e.g., the difficulty of the function operators or tree depth), and the number of features in the leaf nodes, which makes it a multiobjective task. The multiobjective GP methods [23], [24] divide a classification task with c classes into c binary classification tasks, and evolve c sets of genetic programs to create ensembles, which have been used for simultaneous FS and classification [23], and also extracting and selecting linearly separable features [24]. These two methods define three objectives. The first two objectives, i.e., false positives and false negatives, are related to classification performance and focus on the data imbalance issue. The third objective, i.e., the number of features in the leaf nodes, is used to reduce the number of selected features and alleviate the bloating issue of GP.

Support vector machines (SVMs) are powerful classification algorithms. Their classification ability largely depends on the choice of the kernel function and the kernel parameters, as well as the features it uses. Therefore, finding the optimal SVM model and the optimal feature subset must occur simultaneously, which is a multiobjective optimization task. To achieve this, the classification performance, the number of support vectors, the margin, and the feature subset size are considered as four objectives, to simultaneously optimize the classification performance, the parameters of SVMs, and the feature subset using different kernel functions [55]. Note that strictly speaking, this work belongs to many-objective FS, i.e. the number of objectives is greater than 3. However, there are not many works on many-objective FS, so we consider it as MOFS here.

Neural networks (NNs) are capable of dealing with complex high-dimensional data. The output of an NN can be used for classification, and FS can be implicitly achieved by finding the input nodes that are not connected to the hidden layer in the NN [56], which can form an embedded MOFS method that optimizes at least two objectives, i.e., maximizing the NN classification accuracy and minimizing the number of selected features. Generally, the complexity of the NN structure should also be considered. For example, the complexity of the NN structure can be controlled by minimizing the number of hidden neurons to make the structure as compact as possible [56]. In [57], the third objective, namely the regularization of output layer weights, is used to prevent the overfitting of NNs.

TABLE I
A GENERAL COMPARISON OF DIFFERENT MOFS METHODS.

	Classification performance	**Computational cost**	**Generality to different classifiers**
Filter	Low	Low	High
Wrapper	High	High	Low
Embedded	Medium	Medium	Medium

4) Hybrid methods: As shown in Table I, different types of MOFS methods (filter, wrapper, and embedded) have different strengths and weaknesses. To take full advantages of different kinds of MOFS methods, some hybrid MOFS methods have been developed. Although relying on the combination of different MOFS methods, hybrid methods are usually relatively complex but the classification performance is better than a single method.

Typical examples are the combination of filter methods with high efficiency and wrapper methods with high classification accuracy. A tri-objective FS method [58], which regards the feature subset size, the relevance of features, and the classification performance, as three objectives. In this method, most evaluations are based on the filter measure (i.e., mutual information), while only a small part of evaluations (performing a local search for the non-dominated solutions evaluated by the first two objectives) uses a wrapper measure (i.e., the kNN classification algorithm), which can improve the efficiency through reducing the number of classification performance evaluations.

Gene expression datasets usually contain a large number of features, which are not suitable for directly using wrapper methods because of the high computational cost. The filter methods for such problems usually have poor classification performance, since each of them is based on a specific measure and can only explore one aspect of the data. To take advantage of both filter and wrapper methods, a two-stage MOFS method is proposed in [59]. In the first stage, five filter measures are used to test the importance of each feature, and then an ensemble method is utilized to select a small number of important features. In the second stage, by taking the classification performance and the number of selected features as two objectives, a wrapper method is used to further select useful features from the previously selected features in the first stage to improve the classification performance.

Summary: A summary of representative evaluation functions used for MOFS in classification is listed in Table II. For readers interested in more details, please refer to their original papers. As can be seen from the table, although different evaluation functions have been used in different literature, the evaluation functions in most research work take into account classification performance and feature subset size. For example, in filter methods, relevance, redundancy, and/or complementarity can be used to express effectiveness. In wrapper methods, the effectiveness can be represented by the classification performance of the wrapped classification algorithm. In embedded methods such as sparse learning models, the effectiveness can be measured by the loss value, while minimizing the feature subset size is achieved by forcing the regularization as small as possible.

C. Initialization

Unlike single-objective FS which only needs to consider one objective value, the initialization in MOFS needs to consider the distribution of the population in the multi-dimensional objective space. To provide better starting points than that of the traditional random initialization method for the population in MOFS, most existing improved initialization strategies focus on either the *quantity* or *quality* of the selected features.

1) From the perspective of the ***quantity*** *of the selected features:* When treating the number of the selected features as an objective, the initial solutions generated by the traditional random initialization strategy are typically located around the middle area of the objective space. Since each feature has the same probability (i.e., p=0.5) of being selected, the number of features selected in each solution would tend to converge to $p \times D$, where D represents the number of features in a dataset. Thus, as shown in Fig. 6, the initial population is mostly concentrated in the central region in terms of the objective of the number of selected features, while solutions with a low (i.e., $\ll p \times D$) or high number (i.e., $\gg p \times D$) of selected features in the initial population will be rare, resulting in a poor distribution of the initial population in the objective space.

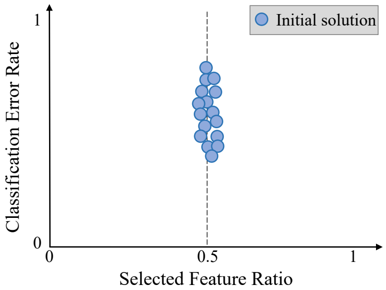


Fig. 6. An example of the initial population distribution in the objective space obtained by the traditional random initialization method.

To address the above issue, some efforts have been devoted to sampling the initial solutions more uniformly (i.e., with a diverse number of selected features) in the objective space. For example, borrowing the idea of the forward and backward initialization methods [67], the segmented initialization method [68] divides the initial population into three subpopulations. Solutions in these three subpopulations select features on the basis of probabilities of one-quarter, one-half, and three-quarters, respectively. By this means, solutions can be sampled around the forward, middle, and backward regions of the objective space (i.e., regions with a small, medium, and large number of selected features, respectively). The interval-based initialization method [69] divides the number of selected features into five intervals, and the FS probabilities of these five intervals decrease as the number of selected features increases. In this way, the number of selected features can cover the whole objective space in terms of the feature subset

TABLE II
SUMMARY OF EVALUATION FUNCTIONS USED IN MOFS.

Category	Evaluation functions	References
Filter	Relevance (relative discrimination criterion) and redundancy (Pearson correlation coefficient)	[36]
	Relevance (mutual information) and redundancy (mutual information)	[33], [34]
	Relevance (mutual information or information gain) and feature subset size	[35]
	Relevance (fuzzy mutual information) and feature subset size	[60]
	Relevance (separability index), redundancy (Pearson correlation coefficient) and feature subset size	[37]
	Inter-class distance and intra-class distance	[38]
	Region purity, region-based distance, and feature subset size	[39]
Wrapper	Classification error rate (or classification accuracy) and feature subset size	[42], [61]
	Balanced classification error rate and feature subset size	[44], [62], [63]
	Classification accuracy and fairness	[46]
	Classification error rate, feature subset size, and missing rate	[52]
	Classification accuracy and reliability value of the selected features	[53]
	Classification error rate and feature cost	[47]–[49]
	Balanced classification error rate, feature subset size, and similarity between feature subsets	[50]
	Specificity, sensitivity, and feature subset size	[64]
	Classification accuracy, feature subset size, and instance subset size	[65]
Embedded	Loss and a predefined number of selected features (reference points)	[30]
	False positives, false negatives, and the number of features at the leaf nodes	[23], [24]
	Square error, the number of hidden neurons, and the number of selected features in the input layer	[56]
	Square error, the number of selected features in the input layer, and generalization (regularization term)	[57]
	Classification performance, the number of support vectors, the margin, and feature subset size	[55]
Hybrid	Classification accuracy, feature subset size, and relevance (mutual information)	[58]
	Classification accuracy, feature subset size, relevance (symmetric uncertainty), and redundancy (symmetric uncertainty)	[66]

size objective, and most solutions only select a small number of features to benefit the population convergence. Different from the above methods which generate initial solutions that select specific proportions of features, in [50], the number of features selected from different feature subsets is randomly generated, which results in a more diverse solution set in the objective space.

2) From the perspective of the ***quality*** *of the selected features:* In addition to improving the objective of the number of selected features, some initialization methods focus on improving the objective of the quality of initial solutions, e.g., classification performance.

The information entropy-based initialization method [70] incorporates the conditional entropy of each feature with respect to class labels into the initialization process, to generate high-quality initial solutions. In MOCDE [71], half of the solutions in the population employ a filter measure to remove some unpromising features, while the remaining solutions in the population select features based on the probability to make relevant features have a high probability to be selected.

Summary: There are also some works that consider both the quality and quantity of the selected features. For instance, the initialization in SparseEA [72] first evaluates the relevance (i.e., classification performance) of every single feature, and then adopts binary tournament selection to select features based on the evaluated relevance for each solution. Note that the number of selected features for each initial solution is randomly generated between 1 and the total number of features in a dataset, which results in various numbers of selected features. One major issue is that if a dataset contains a large number of features, this method will cost a large number of classification performance evaluations. It has been experimentally demonstrated that the traditional random initialization method with uniform sampling is not applicable for solving large-scale multiobjective sparse optimization problems (e.g., MOFS problems) [73]. Adjusting the probability of each feature being selected can improve the diversity of initial feature subsets in the objective space. On the one hand, selecting a small number of features for the initial population (which is similar to the forward search) is beneficial for population convergence, particularly for high-dimensional datasets [74]. On the other hand, if only too few features are selected, it is not beneficial to the exploration of the whole search space, since the wide distribution of the initial population in the objective space could promote the exploration width. For initialization methods that focus on the quality of the selected features, they usually only focus on the relevance of every single feature, but ignore the correlations between features. Considering the selection of features with high relevance and good complementarity can greatly improve the quality of initial solutions [75].

D. Offspring generation

Offspring generation in MOFS is utilized to discover more promising feature subsets. The commonly-used operators include crossover and/or mutation. Crossover, also called recombination, is used to exchange and combine genetic information from multiple parent feature subsets to produce new feature subset(s) in MOFS. It enhances the exploration capabilities of an MOFS method. Mutation is usually applied to perturb a single feature subset to promote population diversity and avoid getting stuck in local optima. However, the canonical offspring generation operators, such as in GA, PSO, DE, and ACO, are not so effective for MOFS tasks, since they rarely consider the characteristics of the MOFS problems, particularly for high-dimensional datasets. To this end, in addition to the traditional offspring generation operators, improved versions focus mainly on the following aspects.

1) Reducing the number of selected features: Minimizing the number of selected features is a major objective in MOFS. From this perspective, FS can be taken as a sparse multiobjective optimization problem, that is, forcing many decision variables to be 0 (forcing many features not to be selected). To address such problems, a pattern mining method [76] and an unsupervised NNs method [77] are used to detect and learn the sparse subspace (0 bits) of the Pareto set, respectively. When generating new offspring solutions, crossover and mutation are performed on the reduced subspace, which greatly reduces the search space and ensures the sparsity of new solutions (i.e., as many features as possible are not selected). In [63], an irrelevance learning method is used to detect irrelevant features based on the population distribution in the objective space. Specifically, if all the features selected by one solution are also selected by another solution that exhibits worse classification performance, the additional selected features can be considered irrelevant. After performing crossover and mutation operations to generate new candidate solutions, if new solutions select some of these irrelevant features, these irrelevant features will be removed according to a certain probability (i.e., flipping their bits from 1 to 0) to reduce the number of selected features for these new solutions.

2) Offspring generation based on feature importance: During the evolutionary process, good solutions in a population, such as non-dominated solutions or solutions with high classification accuracy, usually contain more useful information, i.e., more relevant features or features with better interactions.

Utilizing information from these solutions can accelerate the convergence of the population. The importance of each feature can be assessed by feedback from the population or by some filter measures. For example, the particle ranking-based PSO method [78] divides the objective space into several sub-regions by uniform and non-uniform partitioning methods, and the particle rank and the feature rank are calculated and used to update the velocity and the position of particles in each generation, to speed up the optimization process and steer particles towards the best solutions. Some methods combine filter measures to guide the offspring's generation. The interactive offspring generation method [79] simultaneously evolves a 'wrapper' population (evaluation using the wrapper method) and a 'filter' population (evaluation using the filter method). In this method, the wrapper-to-filter steering strategy uses good feature subsets in the 'wrapper' population to guide the 'filter' population to a better direction, and the filter-to-wrapper repair strategy is employed to utilize the promising information in the 'filter' population to repair some features in the 'wrapper' population, thereby avoiding the 'wrapper' population from falling into a local optimum.

The selection frequency of each feature by good solutions can also reflect the importance of each feature to some extent. If a feature is always selected by many good solutions, it implies that this feature is more important. To use such information during the evolutionary process, in [80], the features are ranked according to each feature's selection frequency of the solutions in the archive. Afterward, the feature rank is used not only to generate new solutions in the archive, but also to update the particles' position in PSO. In [81], the offspring generation operators are based not only on the FS frequency of elite solutions, i.e., non-dominated solutions, but also on the FS frequency of inferior solutions, i.e., solutions at the last front, to synthesize both the positive and negative information of population.

3) Performing local search: Local search is utilized to search for more promising solutions around the current solutions to further improve their performance. EC algorithms usually achieve good performance in global search because their offspring generation operators are based on population. However, their ability to further exploit the identified areas is not particularly efficient. Therefore, it is necessary to combine the offspring generation operators with local search to further improve the search performance of MOFS algorithms.

The one-bit purifying search [82] is suggested to refine the non-dominated solutions in the population, to exploit more promising feature subsets along the elite solutions. In order to find neighboring feature subsets with better classification performance than the current one, in [83], after mutation and crossover operations, three local search operators of insertion, deletion, and swap are proposed under the guidance of the ReliefF indicator [84]. To spend more effort to compensate for the diversity loss of non-dominated solutions in sparse objective regions, a local search combined with GA, called direct multi-search, is performed for the uncrowded non-dominated solution to explore more promising solutions in this area [85].

4) Retaining building blocks of feature subsets: There are complex interactions between features, e.g., redundancy, interaction, and complementarity. Traditional offspring generation operators in EC cannot capture these complex interactions among features. If features with positive interactions can be retained or combined together during the offspring generation process, the quality of new feature subsets will be greatly improved.

To prevent important partial solutions (building blocks) from being broken and capture the most expressive interactions among features, a Bayesian network is employed as the probability model to replace the traditional offspring generation operators to generate new feature subsets [86]. The offspring generation entails considering the interactions between offspring solutions, both to explore around good feature subsets and to guide the population to find more good feature combinations. To achieve this, the crossover operator in [87] considers not only solutions in the current population, but also non-dominated solutions across generations, to mine better feature combinations from non-dominated solution sets of different generations and speed up the convergence of the population.

To improve the search efficiency of genetic operators for high-dimensional datasets, the variable granularity search [88] combines multiple features into a bit, which greatly reduces the search space. Under this representation, the granularity offspring generation operator is proposed to crossover and mutate different feature combinations (granularity bits). As the evolution proceeds, the number of features represented by a bit is gradually reduced, to eventually find high-quality feature subsets.

Summary: The feature subsets obtained by the method that tends to reduce the number of selected features usually contain a small number of selected features, but this method does not place the objective of maximizing the classification performance at a higher priority than the objective of minimizing the feature subset size. The method based on local search aims to refine a feature subset in its neighborhood. However, it is important to determine when and where to perform the local search operator, because they will affect not only the computational burden but also the search efficiency. The offspring generation operators based on the feature importance can exploit the information of individual features, so that highly-relevant features have a high probability of being selected. However, this family ignores the complex interactions among features, which could break good feature combinations. The method of retaining the building blocks of feature subsets tends to combine features with positive interactivity/complementarity. However, the number of such studies is limited, and most of the existing studies are based on filter measures to explore feature interactions, which can only reflect one aspect of the data, and cannot mine the interactions among features based on the performance of a classification algorithm. A desirable approach to detect feature interactions is to consider both the classification performance and distribution of feature subsets, and then maintain the building blocks of these interactive features in offspring generation, e.g., by enabling positively interactive features to be selected together in crossover and avoiding complementary features being separated in mutation.

E. Environmental selection

Environmental selection (population update) plays a vital role in evolution as it determines which candidate solutions can survive, and is the driving force of population evolution. Depending on the selection criterion used [89], the current EMO methods for MOFS could be mainly classified as Pareto dominance-based and decomposition-based approaches.

1) Pareto dominance-based: Due to FS is often formulated as a bi- or tri-objective optimization problem, Pareto dominance-based EMO methods are widely used for MOFS problems [90], since their weakness of the deterioration of the selection pressure in the many-objective space does not appear in the bi-objective or tri-objective space. They also consume fewer computational resources and require no additional parameters.

In Pareto dominance-based EMO, the fitness assignment usually combines the non-dominated sorting with a density estimator (e.g., crowding distance, fitness sharing, entropy, adaptive grids, parallel coordinates) [6]. When employing Pareto dominance-based EMO for MOFS problems, some work replaces the selection criteria from density estimator to the feature subset size [74], to keep a more diversified number of selected features. Some work also uses reference points as the density estimator [91], which calculates the perpendicular distance between a solution and a reference line, and the solutions have the minimum distance with their corresponding reference points are selected. Note that in this method, the reference points are not set uniformly, but more reference points are set on the objective close to the classification error rate to put more emphasis on the classification performance.

MOFS often suffers from the issue of solution duplication in the objective space, which could degenerate the diversity and thus lead to the population stagnation. Some efforts have integrated duplication-handling strategies into the environmental selection process, to consider the quality of solutions in both the objective and search spaces. For example, in [74], before non-dominated sorting, the Manhattan distances among all solutions in the search space are calculated to estimate dissimilarity degrees of solutions. A threshold is used to determine whether a duplicated solution should be deleted. Specifically, among multiple duplicated solutions with dissimilarity less than the threshold, only one randomly selected solution survives. However, this method is computationally expensive because the distance of each pair of solutions in the population needs to be calculated. The duplication-handling method in [62] is implemented after the non-dominated sorting. It only considers the distance between duplicated solutions and the non-dominated solutions with the same or similar classification performance or the number of selected features as them, to filter out some unpromising duplicated solutions in the objective space that contain more redundant features.

Multiple optimal (non-dominated) feature subsets with different features selected can achieve similar or even the same classification performance, and this problem is called multimodal MOFS. To find such multiple optimal feature subsets, the Pareto-dominance relationship is relaxed in [92] to drag the solutions with the same number of selected features and the similar classification performance into the same front level. The density estimator in most Pareto dominance-based EMO methods is based on the objective values of solutions. In order to solve the multimodal MOFS problems, some works [71], [93] modify the density estimator (i.e., crowding distance) by considering the distance of solutions in both the objective and search spaces, to distinguish and identify such multiple optimal feature subsets.

2) Decomposition-based: The core idea of the decomposition-based EMO is to transform a multiobjective optimization problem into several single-objective optimization problems which are simultaneously solved using information from their neighboring subproblems. The performance of decomposition-based EMO relies on the scalarizing function that they adopt. They are also sensitive to weight generation methods. The salient strength of decomposition-based EMO is that they can be easily scaled in the many-objective space.

The Pareto front of MOFS problems is highly discontinuous, and the predefined weight vectors are not suitable for such problems. To amend this issue, an adaptive penalty mechanism based on the penalty boundary interaction (PBI) scalarizing function [94] is proposed. In this method, the penalty value in PBI for each solution is not constant, but adaptively adjusted according to the distance between the solution and the weight vector, to enhance the selection pressure of the archive. To mitigate the effect of the Pareto front shape, instead of setting weight vectors, in [95], a set of reference points on the axis of feature subset size are set, to put more emphasis on the classification performance. During the evolutionary process, the reference points are dynamically adjusted according to the conflicting relationship between the two objectives.

Summary: Note that as a major branch of EMO, i.e., the indicator-based EMO, there has been little effort to apply them to MOFS tasks, even though they have exhibited excellent performance on EMO benchmark problems [96]. Although the other two methods, i.e., Pareto dominance-based and decomposition-based, have been widely used in MOFS problems, they both have their limitations. For example, for the Pareto dominance-based approach, since there is no selection pressure directly toward the two objectives, the obtained solutions usually have a poor spread and are concentrated in the central region of the Pareto front, which makes it hard to find feature subsets with high classification performance and/or a very small number of selected features. By contrast, by setting weight vectors along the two objectives, the decomposition-based methods can find feature subsets with a small number of selected features and high classification performance. Nevertheless, for decomposition-based methods, the predefined weight vectors do not work well because multiple objectives for FS do not always conflict with each other at different regions of the search space. On the contrary, the Pareto dominance-based methods can capture the conflicting information about objectives, as they can easily distinguish dominant relationships between solutions through the Pareto dominance principle.

Some work has been devoted to leveraging the strengths of both Pareto dominance-based and decomposition-based methods while mitigating their weaknesses. A multiform method [44] is proposed to address an MOFS task assisted with its auxiliary single-objective FS tasks in a multi-task environment. In this method, the Pareto dominance method is used for the MOFS task while the weighted sum scalarizing function is used for single-objective FS tasks. Through assigning specific weight vectors for single-objective FS tasks, good feature subsets from single-objective FS tasks could be transferred and shared to promote the search of the MOFS task.

F. Decision making

The goal of MOFS is to provide a representative subset of the non-dominated solutions for decision makers. However, from the point of view of users, among the multiple non-dominated solutions obtained during the training phase, only one single solution is needed. Without specific constraints, the choice of a single solution from the non-dominated set for the unseen test data is not a trivial task. There are three primary decision-making techniques for multiobjective machine learning [97], including objective preference-based, knee point-based, and ensemble-based methods.

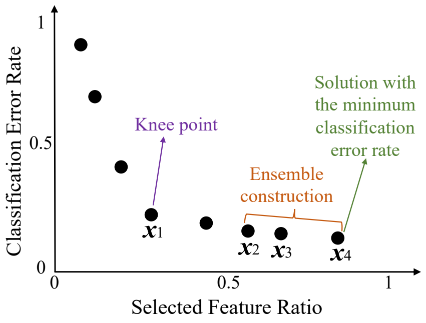


Fig. 7. Examples of different decision-making methods in MOFS.

1) Objective preference-based: Usually, there are different priorities among different objectives. In objective preference-based decision-making methods, among multiple trade-offs, the solution with the best value on the objective with a higher priority preference can be selected. For example, the solution with the minimal classification error rate [62], [74] (e.g., feature subset $\boldsymbol{x}_4$ in Fig. 7) can be chosen when the classification error rate and the selected feature ratio[1] are treated as two objectives, since classification performance is more important than the number of selected features.

2) Knee point-based: Without any prior preference information, knee point (e.g., feature subset $\boldsymbol{x}_1$ in Fig. 7) is considered to be a naturally preferred compromised solution that is attractive to decision makers in MOFS [83]. Knee point is a solution that incurs a large loss in at least one objective and gains a small amount of improvement in other objectives, which makes it intriguing to decision makers in posterior decision making [98].

3) Ensemble-based: The selection of the final solution for the above two methods is based on the characteristics of the MOFS task and the shape of the obtained approximated Pareto front, respectively. Different from the above two methods that only pick up one solution, the ensemble-based methods endeavor to extract and leverage useful information from multiple trade-off solutions, and generate a more reliable and robust solution based on these non-dominated solutions, e.g., feature subsets $\boldsymbol{x}_2$, $\boldsymbol{x}_3$, and $\boldsymbol{x}_4$ in Fig. 7, which can avoid the overfitting issue to some extent and enhance the credibility of the selected features [99], [100].

Summary: Different decision-making methods for MOFS are applicable to different situations. For example, the decision-making method based on objective preference assumes that different objectives for FS have different priorities, and decision makers can easily determine the solution of interest according to preferences. The decision-making method based on the knee point is based on the shape of the non-dominated feature subsets, which is applicable to the situation where the prior preference information is unavailable, or decision makers have no special preference for the MOFS problem. In this case, they can regard the knee point as a favorable feature subset to achieve the smallest trade-off loss at all objectives. The ensemble-based methods undoubtedly take a longer time and the ensemble size is larger than a single solution. However, this is generally a small price to pay considering the accuracy gained from the ensemble on the unseen test set.

IV. Applications

As shown in Fig. 8, this section will provide six representative domain applications where MOFS approaches have been successfully applied. We will briefly describe each application and focus on its problem characteristics and the major motivations for employing the MOFS approach in its specific context.

[1] Selected feature ratio is the ratio of the number of selected features to the total number of features.

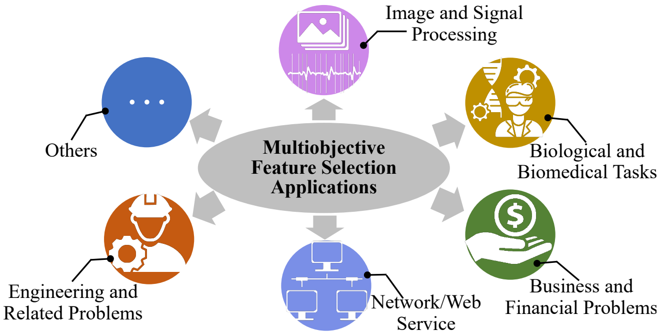


Image and Signal Processing
Others
Multiobjective Feature Selection Applications
Biological and Biomedical Tasks
Business and Financial Problems
Engineering and Related Problems
Network/Web Service

Fig. 8. Representative applications of MOFS.

A. Image and signal processing

Facial emotion recognition reveals emotional state by analyzing facial expressions from images or videos. Feature extraction and FS play a vital role in the emotion recognition process. The features obtained by feature extraction are usually high-dimensional and they are always treated equally in the training of a classification algorithm, which leads to a slow recognition speed and a low emotion recognition rate. In this case, MOFS can be adopted to obtain a number of discriminative features with different trade-offs between the number of selected features and the emotion recognition accuracy [101], [102]. A similar idea is applied for handwritten numeral classification [103].

In hyperspectral image classification, such as target detection, medical imaging, and earth monitoring, by selecting fewer decisive bands, hyperspectral band (feature) selection can reduce the information redundancy, retain the inherent physical information of hyperspectral image data, and improve prediction accuracy. Noise resistance and critical information preservation are the main factors to be considered in band selection, which is a multiobjective task. To achieve this, hyperspectral band selection is formulated as a tri-objective optimization problem, which considers the information, noise, and correlation of the bands as three objectives [104]. Additionally, a target-oriented evaluation strategy is designed to choose the final solution from the obtained non-dominated solution set. An open issue in band selection is how to set the appropriate number of bands that need to be selected to retain most of the information. By setting the number of selected bands as an objective, an EMO algorithm can automatically determine the number of bands to be selected from a group of trade-off candidates [105].

Another application of MOFS is on the brain-computer interface which identifies the brain activity based on electroencephalogram signals [106]. The two objectives are the number of selected features and the inter/intra class distances, which are optimized by NSGA-II [41]. The results demonstrate that this method is much more efficient and selects smaller numbers of features than other benchmark methods. MOFS has also been used to process audio signals for analyzing music segments [107], instrument recognition [108], and smoothing the movement of amputees supporting disabled people [109].

B. Biological and biomedical tasks

Many microarray technology-based cancer detection systems employ machine learning models to classify microarray datasets to improve the accuracy of cancer diagnosis. The microarray datasets typically contain a small number of instances but a large number of genes (features), and many of these genes are irrelevant or redundant. MOFS can improve the diagnostic accuracy by selecting a small number of marker genes from the microarray datasets that play a key role in inducing cancers [110]. However, biologists would gain more information if the relevance of each gene with a *specific* cancer type could be revealed. For this purpose, two new definitions of genes are defined in [111], namely full class relevant (FCR) and partial class relevant (PCR) features, which indicate genes that can distinguish all cancer types and part of cancer types, respectively. To select both FCR and PCR genes simultaneously, a Markov Blanket embedded MOFS method is used to facilitate such cancer identification and prediction.

One of the most common biomedical applications is drug discovery, which usually has many conflicting objectives such as toxicity risk and efficacy (i.e., classification accuracy). Soto *et al.* [112] propose an early work on applying NSGA-II [41] and SPEA2 [113] to select chemical drugs. Artificial NN ensemble measures the efficacy of candidate drugs. Both NSGA-II and SPEA2 can significantly reduce the complexity of drug structures (via FS) while achieving comparative efficacy in comparison with traditional drug discovery methods. Abd Elaziz *et al.* [114] further improve the efficiency by using statistical dependency instead of the classification performance. The proposed algorithm is not only more efficient but also able to evolve more stable drugs with promising results across different drug metrics. The interpretation of results is very important in the context of drug discovery, and the number of selected features affects the difficulty of interpretation. Jiménez *et al.* [115] apply MOFS to build small and accurate decision trees for predicting a drug's efficacy. The key point is the evolved trees are interpretable and easily understood by chemists.

MOFS has also been applied to a wide range of medical diagnoses such as breast cancer [116], heart diagnosis [117], [118], and Parkinson detection [119]. Sohrabi *et al.* [120] apply MOFS to select informative clinical and genetic characteristics to predict the amount of warfarin for a particular patient.

C. Business and financial problems

Credit scoring is an essential task in many banking businesses, which predicts the creditworthiness of a customer based on the customer's information. However, the large amount of customer behavior data collected and stored by financial institutions faces high costs and contains many irrelevant and redundant features, which reduces the prediction performance and unnecessarily increases the model complexity. Kozodoi *et al.* [121] apply NSGA-II to improve the performance of the scoring system by minimizing the number of required features and maximizing the profit. According to the results, all the best solutions evolved by single-objective algorithms are dominated by the solutions evolved by NSGA-II. It is worth noting that minimizing the number of selected features in credit scoring can also reduce the cost of data acquisition.

Credit card fraud detection is a major challenge for banks and card issuers. It is extremely important to detect whether a transaction is fraudulent, because wrong predictions will hit consumers' financial confidence and cause huge losses to banks. MOFS can be used for credit card fraud detection [122]. Through selecting important financial information such as corporate financial statements, customer transactions, and repayment records, it can improve the accuracy of predicting business performance or credit risk of individual customers, and reduce economic losses and consumer financial uncertainty.

D. Network/Web service

Recently, network security is becoming increasingly important, which detects and prevents any network intrusion. The intrusion detection system is a precautionary model based on the network traffic data, to monitor the system against threats and protect network security. However, the network traffic data are generally very large and complex. Improving detection effectiveness and reducing the overhead of the classification problem are two pressing issues that need to be addressed in intrusion detection, which is a multiobjective task. Recent results show that MOFS can select small numbers of features which allows the system network to detect abnormal traffic in real time [123]–[125].

Sousa *et al.* [126] apply NSGA-II to improve Learning to Ranking (L2R) which is an essential component in information retrieval applications such as web search engines. L2R learns a function mapping from a feature vector (extracted from a query) to the ranking of possible answers to the query. The results suggest that MOFS has the ability to select a small number of important features such that the query becomes simpler, and thus the answers are more correct.

E. Engineering and related problems

The purpose of software defect prediction is to predict potential defect modules by learning a defect prediction model. In the process of collecting the defect prediction datasets, researchers have designed different features that are closely related to software defects. However, not all of these features are conducive to learning an effective defect prediction model. MOFS can be used for software defect prediction by minimizing the number of selected features and maximizing the performance of software defect prediction models simultaneously [127], [128]. By analyzing which features are frequently selected by an MOFS method, it can help software testers to conclude which features are more important to the effectiveness of software defect prediction.

MOFS has also been successfully applied to many other complex engineering problems, such as churn prediction in telecommunications [129] and energy consumption prediction [130].

F. Others

Other applications of MOFS, such as in environmental monitoring, mitigate liquefaction damage through soil liquefaction sensitivity prediction [131], and improve the environment health through air quality forecasting and indoor temperature forecasting [132]; in public health, reduce the health risk through recommendation system [133], and so on.

There are many other applications of MOFS that we are unable to cover due to the space limit. We hope that the above example is enough to show a common pattern, that is, MOFS plays an important and practical role in many application fields.

V. Challenges and future directions

Although prior research endeavors have reported promising achievements in MOFS in classification, there are still some issues and challenges that need to be addressed. Note that in addition to the challenges specific to MOFS in classification that will be described below, there are also some common challenges for both single-objective FS and MOFS, such as the low search efficiency on high-dimensional data, high computational cost on a large number of instances, and unbalanced classification, which will not be elaborated here as they have been discussed in detail in another survey [7].

A. Challenges

1) Online MOFS with streaming data: Most of the existing studies for MOFS are limited to batch learning, which assumes that the FS task is performed in an offline/batch learning manner, i.e., all features of training instances can be available in advance [134]. This assumption might not always be applicable to real-world applications because data can arrive sequentially, or collecting full information of training data can be very expensive. If training data cannot be processed in a single pass, their complexity increases with time and memory, leading to a degradation in classification performance.

To improve the classification accuracy and reduce the memory cost, EMO can be used to solve online FS tasks by considering the classification performance and feature subset size simultaneously, either for features and instances arriving individually or in batches [135]. However, online FS brings new challenges to existing MOFS methods. For instance, when a set of new features arrives online, due to shifting data distributions (i.e., concept drift), an MOFS method should not only consider the correlation and redundancy between the new features, but also between the new features and the previously selected features. Additionally, the previously selected features that are no longer relevant to the current state should also be removed [136].

2) Multi-modal MOFS: The existence of interactive and redundant features can result in multiple optimal feature subsets in the dataset, i.e., different feature subsets with the same number of selected features have similar or even the same classification performance. Searching for multiple optimal feature subsets with the same number of different features selected has important meanings, e.g., biological hints [71]. Also, in real-world applications, the cost or difficulty of collecting different features can be different. For feature subsets with similar or identical classification performance, users can make choices based on their own preferences or real-world situations.

The goal of multi-modal MOFS is to find as many optimal feature subsets as possible in the search space. Although the idea of niching [137] borrowed from evolutionary multi-modal optimization has been applied to solve multimodal MOFS problems [66], [71], [92], [138], there are still some challenges in this topic. For example, measuring the diversity of solutions in both the objective space and search space is usually computationally expensive, and how to balance and combine the diversity contribution of these two spaces is also an issue, which requires further work.

3) MOFS in multi-label classification: In canonical classification tasks, each instance in the dataset belongs to only one class label. Nevertheless, in some real-world applications, e.g., for semantic image and video annotation, each instance can be associated with multiple class labels simultaneously, and these class labels are not mutually exclusive. Such classification tasks are often known as multi-label classification problems [139].

MOFS for multi-label classification is more challenging than MOFS for single-label classification [140], [141]. This is because when using EMO to perform FS for multi-label classification, in addition to the relevance between features and class labels and the correlation between features, the correlation between class labels also needs to be considered to improve both the quality and scalability of the classification.

4) Transfer learning for MOFS: Existing work on MOFS tends to solve each MOFS task independently. Nevertheless, some MOFS tasks have related feature spaces and/or related classification tasks. These MOFS tasks are closely connected to each other, such that they share common knowledge of problem-solving and have related optimal feature subsets [142]. Recently, evolutionary transfer optimization [143], [144], a paradigm that combines EC with knowledge learning, transfer, and sharing from related domains, to boost the optimization efficiency and performance, can benefit MOFS. Transfer learning for MOFS tasks can be a sequential method or a multi-task method, which are suitable for the following two situations respectively:

- **Sequential method:** we are more interested in the target classification task, so we can solve the source classification task first and then extract the transferred knowledge to help solve the target classification task.
- **Multi-task method:** it focuses on addressing the problem of joint MOFS across a group of classification tasks, and these multiple classification tasks have equal or similar priority.

Note that for evolutionary multi-task FS, there are some works that use EC methods to solve a specific FS task by combining with its artificially constructed auxiliary FS tasks in a multi-task environment, which can be broadly categorized into the following three categories:

- Methods focus on the **number of features:** It converts a high-dimensional FS task into several related low-dimensional FS tasks, and then searches for an optimal feature subset by transferring knowledge between these FS tasks [145]–[148].
- Methods focus on the **number of instances :** It decomposes the original large-scale data into several small-scale data, so as to promote evolutionary feature subset search by using small-scale data to quickly optimize for the large-scale dataset [144].
- Methods focus on the **problem formulation:** It constructs multiple single-objective FS tasks based on the distribution of the MOFS task, to facilitate improving the performance of the MOFS task by using the useful knowledge provided by the constructed single-objective FS tasks [44].

Only the third of the above methods is dedicated to solving the MOFS problems. However, this method is not really to address multiple MOFS tasks, but uses the *multiform* to construct auxiliary tasks for a specific MOFS task.

If the source MOFS task and the target MOFS task have different feature sets, their search spaces are also different. How to effectively learn and transfer knowledge between related MOFS tasks with different but overlapping search spaces is pivotal to the success of transfer learning for MOFS. In addition, useful knowledge in multiobjective spaces is usually in the form of non-dominated solutions, how to capture and utilize the complex interactions among these non-dominated feature subsets remains a challenge.

5) Many-objective FS in classification: Many-objective optimization is a hot topic in the realm of EC [149]. Most existing research endeavors treat FS as a bi-objective or tri-objective optimization task. Many-objective FS enables users to analyze different aspects of complex data by considering more criteria simultaneously (typically greater than three), such as correlation, redundancy, feature subset size, robustness, etc [150], and has recently emerged in many fields [151]. For instance, in imbalanced learning, classifiers built from unbalanced datasets are often biased toward the majority class. Unfortunately, the minority class is often the class of interest. Therefore, by taking the classification performance of each class as an objective, a many-objective FS problem can be formulated to alleviate the issue of class imbalance [124].

Similar to that in the many-objective optimization domain, many-objective FS also poses challenges to the existing EMO methods, including the selection operators, computational cost, and visualization of feature subsets in the many-objective space. Since the number of approximate solutions required grows exponentially with the number of objectives, it is a promising approach to focus on a small promising region of the Pareto front by utilizing the preference information of decision makers. However, it is still challenging since the prior knowledge about the Pareto front for many-objective FS problems is unknown.

6) MOFS for multi-modal learning: Nowadays, massive amounts of data with high-capacity and high-variety features are generated in many domains. These data often involve multiple data modalities, e.g., sight, sound, movement, touch, and smell, which are known as multi-modal big data. They contain abundant inter-modality and cross-modality information, and span multiple disciplines such as signal processing, pattern recognition, and computer vision, which have emerged as an extensively researched topic. Multi-modal learning provides more accuracy and robustness than single-modality learning, as it combines information from multiple sources at the signal level and/or the semantic level [152].

Since these data come from multiple modalities, the data after feature fusion could be high-dimensional and contain a lot of redundant and irrelevant features, leading to a decrease in classification performance [153]. By considering multiple criteria, EMO has the strength of identifying relevant and informative feature subsets to represent applications across all domains, to improve the classification accuracy and reduce the data size simultaneously. However, how to design evaluation functions and develop new EMO to better represent global and local patterns for these multi-modal data are worth further investigation. MOFS for multi-modal learning is a research topic that is just beginning to blossom.

7) Multiobjective optimization for simultaneously FS and instance selection: The advancement of data collection technology has enabled data to reach an unprecedented scale. Both the number of features and the number of training instances have an impact on the computational cost and performance of the classification algorithm. In increasingly large classification datasets, not only features, but also training instances are not all useful for classification. Removing redundant and noisy training instances and only selecting representative instances can be helpful in reducing the execution time without affecting the FS results, especially for instance-based classification algorithms, e.g., kNN, which uses the entire training set.

EMO can be used for simultaneously FS and instance selection to improve the classification performance and reduce the computational cost. However, it is not a trivial task since FS and instance selection are interrelated [154]. Representative instances contribute to the selection of important and useful features, and important and useful features in turn contribute to the selection of representative instances.

B. Future directions

1) Handling different preferences between objectives: Most existing works treat the commonly-used two objectives (i.e., classification performance and feature subset size) as equally important. However, the preferences for these two objectives are typically different. The classification performance is clearly more important than the number of selected features. This could also be revealed from the widely-used evolutionary single-objective FS scalar function, such that the weight assigned to the classification performance is far greater than that of the number of selected features [67]. More emphasis should be placed on the objective of maximizing the classification performance in the optimization process.

There are some potential future directions to address the above issue. First, the preference-inspired EMO [63], [155] can be adopted to introduce the objective preference information into the optimization process (e.g., a priori, interactive, or a posteriori) to steer the search towards regions of interest, i.e., prioritize the objective of maximizing the classification performance over the objective of minimizing the feature subset size. Second, constraints can be imposed on solutions with a small number of selected features but poor classification performance, to preferentially select solutions with better classification performance [62]. Third, for decomposition-based methods, the weight vectors or reference points can be specifically designed in a subtle way to put more emphasis on the objective of classification performance [95]. Fourth, for Pareto dominance-based methods, instead of using canonical Pareto dominance criterion, a more general form of dominance relationship, e.g., α dominance [156], can be used to adjust the trade-off rates between objectives.

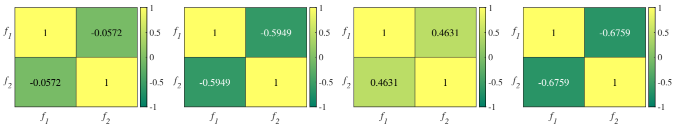
2) Handling highly-discrete Pareto front: The Pareto front of MOFS problems is highly-discrete. The setting of weight vectors can greatly affect the performance of decomposition-based EMO, as they determine the search direction and the distribution of the obtained solution set. Specifying a suitable set of weight vectors for solving the MOFS task is very challenging, mainly because $a\ priori$ information about the geometry of the Pareto front is unknown, and the Pareto front may be only composed of several sparse discrete points in the objective space. This differs from many current benchmarks in the EMO field, most of which are designed for continuous numerical optimization problems with known Pareto fronts in advance, even those known Pareto fronts are not used during the optimization process.

To solve the MOFS problem with the discrete Pareto front, it is necessary to adaptively adjust not only the position but also the number of weight vectors, to respond to the distribution changes of the population in the objective space in time. Future work can consider the above direction to further explore the potential of decomposition-based EMO for MOFS.

3) Handling partially conflicting objectives: The multiple objectives in FS tasks are not always conflicting with each other, e.g., classification performance versus feature subset size. For example, Fig. 9 plots the correlation matrix between the classification error rate and the number of selected features on the Yale dataset [157] at generations of 10, 40, 70, and 100, respectively. The population obtained by MOEA/D[2] [158] is used to estimate the correlation matrix (i.e., Pearson correlation coefficients), which implies the conflicting degree between the two objectives. A correlation value close to -1 (close to 1) indicates a strong conflicting relationship (or positive correlation) between the two objectives, and a correlation value close to 0 suggests that there is no dependency between the two objectives. It can be observed from the figure that the correlation value between the classification performance and the number of selected features varies with generations, which suggests their conflicting relationships and conflicting degree are varies at different objective regions. This phenomenon arises probably from the complex interactions between features. On the one hand, removing some irrelevant and redundant features may reduce the classification error rate, so that the two objectives no longer conflict. On the other hand,

[2] In our experiments, the population size is 200 and the maximal number of generations is 100; the maximal number of solutions replaced by each offspring is 2; the probability that parent solutions are selected from the neighborhood is 0.9. The Tchebycheff approach is used as the aggregation function.

selecting some relevant, complementary, or interactive features can also decrease the classification error rate, thus making the two objectives conflict with each other.



(a) g=10 (b) g=40 (c) g=70 (d) g=100

Fig. 9. Correlation matrix between objectives obtained by MOEA/D at different generations on the Yale dataset.

The above issue also poses a challenge to canonical decomposition-based EMO, as they cannot identify the non-conflicting regions in the objective space. Predefined weight vectors for such problems will lose their effectiveness and efficiency, because a large number of weight vectors in non-conflicting regions will lead to wasted computational costs [95]. A desirable way for future work is to detect and estimate the conflicting degree between objectives based on the evolutionary status of the population, and adaptively adjust the weight vectors accordingly. It is noteworthy that the Pareto dominance-based methods seem to be exempt from such issues. This can be attributed to the case that the non-dominated sorting operator utilized in most Pareto dominance-based methods could easily distinguish the dominance relationship between each pair of solutions, and select non-dominated solutions could provide convergence pressure to the Pareto front.

4) Designing more appropriate performance metrics: Generally, the solution set obtained by MOFS methods can be assessed from two aspects: 1) efficiency and 2) effectiveness. Efficiency can be simply measured by the time associated to search for the best feature subsets, such as training time.

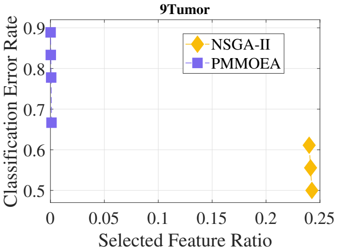


Fig. 10. Distributions of non-dominated solutions obtained by NSGA-II and PMMOEA on the 9Tumor dataset, respectively.

Effectiveness is used to measure the quality of the selected subset of features. The commonly used performance metrics in EMO, such as Inverted Generation Distance (IGD) [159] and Hypervolume (HV) [160], can measure both the diversity and convergence of the obtained non-dominated solutions. However, they do not really reflect the quality of the non-dominated solution set according to the characteristics of the MOFS problem. For instance, Fig. 10 plots the non-dominated solutions on the test data obtained by NSGA-II [41] and PMMOEA [76] when kNN is used as a classification algorithm on the 9Tumor dataset[3]. Compared with NSGA-II, it can be observed that the solutions obtained by PMMOEA select a smaller number of features, but achieve larger classification error rates (worse classification performance). The HV and IGD values for these two sets of solutions show that PMMOEA can outperform NSGA-II. However, the minimum classification error rate of the solution obtained by NSGA-II is about 20% smaller/better than that of PMMOEA. In reality, the solution set obtained by PMMOEA is much worse than that of NSGA-II, since the classification performance is more important than the number of selected features. Therefore, the performance metrics commonly used in EMO cannot really reflect the quality of the solution set for MOFS tasks.

Classification performance is the most direct and effective way to assess the effectiveness of MOFS methods for classification. Some performance metrics, such as minimum classification error rate [74], are only based on one solution. Such results do not seem to represent the performance of all the non-dominated solutions obtained by an MOFS algorithm.

In order to compare the results of different MOFS methods more fairly and to provide more information to decision makers, better performance metrics based on the characteristics of the MOFS problem are needed. Using scalar functions or setting specific reference points by combining preference information to bias the evaluation towards the desired region of the Pareto front can be potential future directions to go.

5) Handling objective selection bias: From the optimization perspective, another challenge in MOFS is the objective selection bias [62], that is, it is more likely to obtain solutions that select a small number of features with poor classification performance than the case of selecting a large number of features with better classification performance (e.g., Fig. 10). This can be attributed to the fact that maximizing the classification performance is much more difficult than minimizing the feature subset size, as the latter could be easily realized through selecting fewer features. Thus the obtained solutions are easily biased towards the objective of minimizing the number of selected features during the search process. By contrast, maximizing the classification performance requires considering the relevance of selected features and the complex interactions among features. It is very challenging to select useful features to improve classification performance from high-dimensional data which usually contains lots of irrelevant and redundant features. These irrelevant and redundant features usually degenerate the classification performance and mislead the classification task.

Unfortunately, the objective selection bias issue in MOFS has seldom been explored, so more work on detecting and correcting the objective selection bias is needed in the future.

6) Handling solution duplication: Since FS belongs to a kind of combinatorial optimization problem, the issue of solution duplication in both the search space and objective space frequently occurs in MOFS, especially when the dataset contains lots of redundant features and/or very few training instances. Solution duplication in the search space can be easily solved by reserving only one of the many duplicated solutions. However, solution duplication in the objective space is more complex to address. The presence of redundant and interactive features leads to multiple feature subsets selecting the same number of but different features and exhibiting the

[3] http://www.gems-system.org

same classification accuracy, e.g., solutions $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, and $\boldsymbol{x}_3$ in Fig. 11, and solutions $\boldsymbol{x}_4$ and $\boldsymbol{x}_5$ in Fig. 11. A large number of duplicated solutions that exhibit the same objective values in the objective space occupy the population, which can result in poor diversity and stagnation of the population. Some duplicated solutions in the objective space should be removed from the population in time. Nevertheless, it is not an easy task to determine which duplicated solutions in the objective space should be discarded or saved.

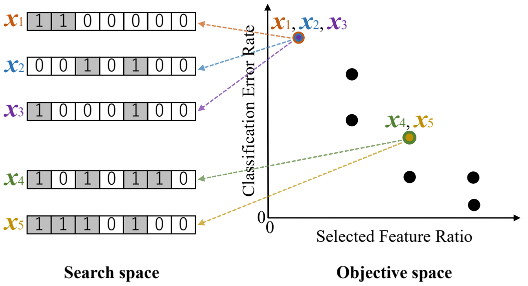


Fig. 11. Illustration of solution duplication in the objective space.

A possible way to deal with the above challenge is to consider the distance (similarity) between solutions not only in the objective space, but also in the search space. Although there have been some attempts to alleviate this issue [44], [74], most of them incur a high time complexity. There is still great potential for developing novel duplication-handling strategies for MOFS.

7) How to visualize final results: Due to the stochastic nature of EC, the non-dominated solutions obtained by an EMO algorithm for MOFS are likely to be different in different runs. How to visualize these solutions in the objective space from the multiple runs so that readers can easily understand and compare these results is a challenge.

Commonly-used methods for visualizing final results focus on the following two. The first is to plot the non-dominated solution set with the median performance metrics [62], e.g., HV. Obviously, this is the result of one representative run out of several runs. However, it cannot reflect the overall results obtained by an EMO algorithm in all runs and the robustness of these results. The second is to show the average non-dominated solution set [42], [161]. Specifically, the multiple non-dominated solution sets obtained by an EMO algorithm in the multiple independent runs are first combined into a union set. Then, the classification performance of all solutions with the same number of features is averaged. In this way, by combining all possible numbers of selected features and their corresponding average classification performance, an "average non-dominated solution set" can be obtained. However, strictly speaking, the "average non-dominated solutions" are actually not non-dominated solutions anymore. Such results can be considerably affected by extreme solutions. Obviously, the above two methods have their limitations. Unfortunately, there is not much work focused on this issue. Novel visualization methods will become a necessity.

8) Applying EMO to sparse learning-based FS: Sparse learning-based FS has gained great popularity due to its promising performance and good interpretability [29]. It usually performs sparse learning model training and FS through minimizing a loss function penalized by a regularization term. The sparse regularizer forces many feature coefficients to be smaller or exactly zero, and then the corresponding features with large feature coefficients can be selected. However, it is difficult for canonical methods to address some forms of regularization (e.g., $\ell_{2,0}$-norm), due to the difficulties such as non-linearity, discreteness, non-differentiability, and non-convexity. Moreover, setting an appropriate regularization parameter that balances the loss function and the regularization is not easy. Finally, the number of selected features needs to be predefined, which is not known in advance and is usually problem dependent.

Minimizing the loss function value and forcing feature sparsity are two main objectives in sparse learning-based FS, and they usually conflict with each other to some extent, which is inherently a multiobjective task. In this case, EMO has great potential as an alternative tool to offset some of the above-mentioned difficulties in sparse learning-based FS, and to produce a set of approximately optimal solutions.

9) Discretization-based MOFS: Data discretization is to find the smallest set of cut-points that best discretize the continuous features to facilitate classification. Through data discretization, some noise disturbances on the continuous data can also be eliminated, resulting in more robust results. Currently, most MOFS methods do not consider discretization when dealing with continuous data. However, some classifiers, such as decision trees and Naive Bayes, are primarily designed for discrete data.

Recently, applying EC methods to perform data discretization and FS simultaneously has shown promising results. For example, to automatically choose a cut-point for each feature, in [162], after obtaining a cut-point table (all possible cut-points for each feature) through an entropy-based discretization method, the best cut-point is selected by using PSO to perform discretization and feature selection. Said *et al.* [163] treat FS and data discretization as a bi-level optimization problem, where FS is performed on the upper level and data discretization is performed on the lower level. Note that in the above two methods, a distance measure used to explore the boundary between the instances belonging to different classes is integrated into the objective function through a weighted sum method. Zhou *et al.* [99] analyze the conflict relationship between the distance measure, the classification performance, and the number of selected features, and regard FS and data discretization as a multiobjective optimization problem. Moreover, they encode the number of selected cut-points for each feature as the particle's position in PSO, thus allowing the selection of multiple cut-points.

Discretization-based MOFS deserves more attention in the future. More efforts should be devoted to investigating the dependencies between data discretization and FS, and how to integrate multiple cut-points generation and selection into EMO (e.g., solution representation and evaluation functions) to better discretize data and select features.

VI. Conclusions

In this paper, we have conducted a comprehensive survey for evolutionary MOFS in classification, in which essential components such as solution representation, evaluation function (wrapper/filter/embedded), population initialization, offspring generation, environmental selection, and decision making have been discussed extensively, and the strength and weakness of each category of methods have been summarized. In addition, we have discussed the applications of MOFS in various fields, such as image and signal processing, biological and biomedical tasks, business and financial problems, network/web service, and engineering problems, and illustrated the necessity of MOFS for these fields. While state-of-the-art techniques have made significant progress in solving MOFS, we have also identified and summarized the major issues and challenges when using EMO for MOFS, and suggested some possible future research directions.

Although evolutionary MOFS has been widely applied to many real-world applications such as medical and financial problems, it is still a young and growing research realm that will continue to play a pivotal role in data mining, machine learning, and EC. We hope that this survey can facilitate related researchers to understand the state of the art in this area, and inspire more fruitful future research and applications.

References

[1] I. Joanito, P. Wirapati, N. Zhao, Z. Nawaz, G. Yeo, F. Lee, C. L. Eng, D. C. Macalinao, M. Kahraman, H. Srinivasan *et al.*, "Single-cell and bulk transcriptome sequencing identifies two epithelial tumor cell states and refines the consensus molecular classification of colorectal cancer," *Nature Genetics*, vol. 54, no. 7, pp. 963–975, 2022.

[2] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM Computing Surveys*, vol. 50, no. 6, pp. 1–45, 2017.

[3] B. Ahadzadeh, M. Abdar, F. Safara, A. Khosravi, M. B. Menhaj, and P. N. Suganthan, "SFE: A simple, fast and efficient feature selection algorithm for high-dimensional data," *IEEE Transactions on Evolutionary Computation*, 2023, doi:10.1109/TEVC.2023.3238420.

[4] S. Ding, H. Zhu, W. Jia, and C. Su, "A survey on feature extraction for pattern recognition," *Artificial Intelligence Review*, vol. 37, no. 3, pp. 169–180, 2012.

[5] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.

[6] C. A. C. Coello, S. G. Brambila, J. F. Gamboa, M. Tapia, and G. Castillo, "Multi-objective evolutionary algorithms: Past, present, and future," in *Black Box Optimization, Machine Learning, and No-Free Lunch Theorems*. Springer, 2021, pp. 137–162.

[7] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 4, pp. 606–626, 2015.

[8] L. Brezočnik, I. Fister, and V. Podgorelec, "Swarm intelligence algorithms for feature selection: a review," *Applied Sciences*, vol. 8, no. 9, p. 1521, 2018.

[9] B. H. Nguyen, B. Xue, and M. Zhang, "A survey on swarm intelligence approaches to feature selection in data mining," *Swarm and Evolutionary Computation*, vol. 54, pp. 1–16, 2020.

[10] T. Dokeroglu, A. Deniz, and H. E. Kiziloz, "A comprehensive survey on recent metaheuristics for feature selection," *Neurocomputing*, vol. 494, pp. 269–296, 2022.

[11] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, and C. A. C. Coello, "A survey of multiobjective evolutionary algorithms for data mining: Part i," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 4–19, 2013.

[12] Q. Al-Tashi, S. J. Abdulkadir, H. M. Rais, S. Mirjalili, and H. Alhussian, "Approaches to multi-objective feature selection: A systematic literature review," *IEEE Access*, vol. 8, pp. 125 076–125 096, 2020.

[13] Y. Jin and B. Sendhoff, "Pareto-based multiobjective machine learning: An overview and case studies," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 3, pp. 397–415, 2008.

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[15] S. Ö. Arik and T. Pfister, "Tabnet: Attentive interpretable tabular learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 8, 2021, pp. 6679–6687.

[16] H. Jh, "Adaptation in natural and artificial systems," *Ann Arbor*, 1975.

[17] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *International Conference on Neural Networks*, vol. 4. IEEE, 1995, pp. 1942–1948.

[18] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[19] J. González, J. Ortega, M. Damas, P. Martín-Smith, and J. Q. Gan, "A new multi-objective wrapper method for feature selection–accuracy and stability analysis for BCI," *Neurocomputing*, vol. 333, pp. 407–418, 2019.

[20] X. Ma, X. Li, Q. Zhang, K. Tang, Z. Liang, W. Xie, and Z. Zhu, "A survey on cooperative co-evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 421–441, 2019.

[21] H. Li, F. He, Y. Chen, and Y. Pan, "MLFS-CCDE: multi-objective large-scale feature selection by cooperative coevolutionary differential evolution," *Memetic Computing*, vol. 13, no. 1, pp. 1–18, 2021.

[22] K. Neshatian and M. Zhang, "Pareto front feature selection: using genetic programming to explore feature space," in *Annual Conference on Genetic and Evolutionary Computation*, 2009, pp. 1027–1034.

[23] K. Nag and N. R. Pal, "A multiobjective genetic programming-based ensemble for simultaneous feature selection and classification," *IEEE Transactions on Cybernetics*, vol. 46, no. 2, pp. 499–510, 2015.

[24] K. Nag and N. R. Pal, "Feature extraction and selection for parsimonious classifiers with multiobjective genetic programming," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 3, pp. 454–466, 2020.

[25] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29–41, 1996.

[26] Z. Wang, S. Gao, M. Zhou, S. Sato, j. Cheng, and J. Wang, "Information-theory-based nondominated sorting ant colony optimization for multiobjective feature selection in classification," *IEEE Transactions on Cybernetics*, pp. 1–14, 2022, doi: 10.1109/TCYB.2022.3185554.

[27] M. Rostami, S. Forouzandeh, K. Berahmand, M. Soltani, M. Shahsavari, and M. Oussalah, "Gene selection for microarray data classification via multi-objective graph theoretic-based method," *Artificial Intelligence in Medicine*, vol. 123, p. 102228, 2022.

[28] F. Cheng, C. Zhou, X. Liu, Q. Wang, J. Qiu, and L. Zhang, "Graph-based feature selection in classification: Structure and node dynamic mechanisms," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2022, doi:10.1109/TETCI.2022.3225550.

[29] X. Li, Y. Wang, and R. Ruiz, "A survey on sparse learning models for feature selection," *IEEE Transactions on Cybernetics*, vol. 52, no. 3, pp. 1642–1660, 2022.

[30] K. Demir, B. H. Nguyen, B. Xue, and M. Zhang, "Multi-objective feature selection with a sparsity-based objective function and gradient local search for multi-label classification," in *International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2021, pp. 823–832.

[31] Z.-H. Zhan, J. Zhang, Y. Lin, J.-Y. Li, T. Huang, X.-Q. Guo, F.-F. Wei, S. Kwong, X.-Y. Zhang, and R. You, "Matrix-based evolutionary computation," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 6, no. 2, pp. 315–328, 2022.

[32] H. Peng, F. Long, and 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*, vol. 27, no. 8, pp. 1226–1238, 2005.

[33] Z. Wang, M. Li, and J. Li, "A multi-objective evolutionary algorithm for feature selection based on mutual information with a new redundancy measure," *Information Sciences*, vol. 307, pp. 73–88, 2015.

[34] H. Dong, J. Sun, T. Li, R. Ding, and X. Sun, "A multi-objective algorithm for multi-label filter feature selection problem," *Applied Intelligence*, vol. 50, no. 11, pp. 3748–3774, 2020.

[35] B. Xue, L. Cervante, L. Shang, W. N. Browne, and M. Zhang, "Multi-objective evolutionary algorithms for filter based feature selection in classification," *International Journal on Artificial Intelligence Tools*, vol. 22, no. 04, p. 1350024, 2013.

[36] M. Labani, P. Moradi, and M. Jalili, "A multi-objective genetic algorithm for text feature selection using the relative discriminative criterion," *Expert Systems with Applications*, vol. 149, p. 113276, 2020.

[37] M. Rostami, S. Forouzandeh, K. Berahmand, and M. Soltani, "Integration of multi-objective PSO based feature selection and node centrality for medical datasets," *Genomics*, vol. 112, no. 6, pp. 4370–4384, 2020.

[38] S. Paul and S. Das, "Simultaneous feature selection and weighting–an evolutionary multi-objective optimization approach," *Pattern Recognition Letters*, vol. 65, pp. 51–59, 2015.

[39] Y. Zhou, Y. Qiu, and S. Kwong, "Region purity-based local feature selection: A multi-objective perspective," *IEEE Transactions on Evolutionary Computation*, 2022, doi:10.1109/TEVC.2022.3222297.

[40] N. Spolaôr, A. C. Lorena, and H. D. Lee, "Multi-objective genetic algorithm evaluation in feature selection," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2011, pp. 462–476.

[41] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[42] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: A multi-objective approach," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1656–1671, 2012.

[43] G. Patterson and M. Zhang, "Fitness functions in genetic programming for classification with unbalanced data," in *Australasian Joint Conference on Artificial Intelligence*. Springer, 2007, pp. 769–775.

[44] R. Jiao, B. Xue, and M. Zhang, "Benefiting from single-objective feature selection to multiobjective feature selection: A multiform approach," *IEEE Transactions on Cybernetics*, pp. 1–14, 2022, doi: 10.1109/TCYB.2022.3218345.

[45] K. Chen, B. Xue, M. Zhang, and F. Zhou, "Correlation-guided updating strategy for feature selection in classification with surrogate-assisted particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 5, pp. 1015–1029, 2021.

[46] J. Brookhouse and A. Freitas, "Fair feature selection with a lexicographic multi-objective genetic algorithm," in *International Conference on Parallel Problem Solving from Nature (PPSN)*. Springer, 2022, pp. 151–163.

[47] Y. Zhang, D.-W. Gong, and J. Cheng, "Multi-objective particle swarm optimization approach for cost-based feature selection in classification," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 14, no. 1, pp. 64–75, 2017.

[48] Y. Zhang, S. Cheng, Y. Shi, D.-W. Gong, and X. Zhao, "Cost-sensitive feature selection using two-archive multi-objective artificial bee colony algorithm," *Expert Systems with Applications*, vol. 137, pp. 46–58, 2019.

[49] Y. Hu, Y. Zhang, and D.-W. Gong, "Multiobjective particle swarm optimization for feature selection with fuzzy cost," *IEEE Transactions on Cybernetics*, vol. 51, no. 2, pp. 874–888, 2020.

[50] R. Jiao, B. Xue, and M. Zhang, "A tri-objective method for bi-objective feature selection in classification," *Evolutionary Computation (MIT Press)*, pp. 1–30, 2023.

[51] X. Zhu and X. Wu, "Class noise vs. attribute noise: A quantitative study," *Artificial Intelligence Review*, vol. 22, no. 3, pp. 177–210, 2004.

[52] Y. Xue, Y. Tang, X. Xu, J. Liang, and F. Neri, "Multi-objective feature selection with missing data in classification," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 6, no. 2, pp. 355–364, 2022.

[53] Y. Zhang, D.-W. Gong, and Z. Wan-Qiu, "Feature selection of unreliable data using an improved multi-objective PSO algorithm," *Neurocomputing*, vol. 171, pp. 1281–1290, 2016.

[54] Y. Mei, Q. Chen, A. Lensen, B. Xue, and M. Zhang, "Explainable artificial intelligence by genetic programming: A survey," *IEEE Transactions on Evolutionary Computation*, 2022, doi: 10.1109/TEVC.2022.3225509.

[55] A. Bouraoui, S. Jamoussi, and Y. BenAyed, "A multi-objective genetic algorithm for simultaneous model and feature selection for support vector machines," *Artificial Intelligence Review*, vol. 50, no. 2, pp. 261–281, 2018.

[56] L. Bai, H. Li, W. Gao, J. Xie, and H. Wang, "A joint multiobjective optimization of feature selection and classifier design for high-dimensional data classification," *Information Sciences*, vol. 626, pp. 457–473, 2023.

[57] F. Coelho, M. Costa, M. Verleysen, and A. P. Braga, "LASSO multi-objective learning algorithm for feature selection," *Soft Computing*, vol. 24, no. 17, pp. 13 209–13 217, 2020.

[58] M. Hammami, S. Bechikh, C.-C. Hung, and L. Ben Said, "A multi-objective hybrid filter-wrapper evolutionary approach for feature selection," *Memetic Computing*, vol. 11, no. 2, pp. 193–208, 2019.

[59] A. Chaudhuri and T. P. Sahu, "Multi-objective feature selection based on quasi-oppositional based Jaya algorithm for microarray data," *Knowledge-Based Systems*, vol. 236, p. 107804, 2022.

[60] E. Hancer, B. Xue, M. Zhang, D. Karaboga, and B. Akay, "A multi-objective artificial bee colony approach to feature selection using fuzzy mutual information," in *IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2015, pp. 2420–2427.

[61] T. M. Hamdani, J.-M. Won, A. M. Alimi, and F. Karray, "Multi-objective feature selection with NSGA-II," in *International Conference on Adaptive and Natural Computing Algorithms*. Springer, 2007, pp. 240–247.

[62] R. Jiao, B. Xue, and M. Zhang, "Solving multiobjective feature selection problems in classification via problem reformulation and duplication handling," *IEEE Transactions on Evolutionary Computation*, pp. 1–15, 2022, doi:10.1109/TEVC.2022.3215745.

[63] R. Jiao, B. Xue, and M. Zhang, "Handling different preferences between objectives for multi-objective feature selection in classification," in *Australasian Joint Conference on Artificial Intelligence*. Springer, 2022, pp. 237–251.

[64] C. J. Tan, C. P. Lim, and Y.-N. Cheah, "A multi-objective evolutionary algorithm-based ensemble optimizer for feature selection and classification with neural network models," *Neurocomputing*, vol. 125, pp. 217–228, 2014.

[65] H. Ishibuchi and T. Nakashima, "Multi-objective pattern and feature selection by a genetic algorithm," in *Annual Conference on Genetic and Evolutionary Computation*, 2000, pp. 1069–1076.

[66] G. Karakaya, S. Galelli, S. D. Ahipaşaoğlu, and R. Taormina, "Identifying (quasi) equally informative subsets in feature selection problems for classification: a max-relevance min-redundancy approach," *IEEE Transactions on Cybernetics*, vol. 46, no. 6, pp. 1424–1437, 2016.

[67] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms," *Applied Soft Computing*, vol. 18, pp. 261–276, 2014.

[68] H. Xu, B. Xue, and M. Zhang, "Segmented initialization and offspring modification in evolutionary algorithms for bi-objective feature selection," in *Genetic and Evolutionary Computation Conference (GECCO)*, 2020, pp. 444–452.

[69] Y. Xue, X. Cai, and F. Neri, "A multi-objective evolutionary algorithm with interval based initialization and self-adaptive crossover operator for large-scale feature selection in classification," *Applied Soft Computing*, vol. 127, p. 109420, 2022.

[70] J. Luo, D. Zhou, L. Jiang, and H. Ma, "A particle swarm optimization based multiobjective memetic algorithm for high-dimensional feature selection," *Memetic Computing*, vol. 14, no. 1, pp. 77–93, 2022.

[71] P. Wang, B. Xue, J. Liang, and M. Zhang, "Multiobjective differential evolution for feature selection in classification," *IEEE Transactions on Cybernetics*, vol. 53, no. 7, pp. 4579–4593, 2023.

[72] Y. Tian, X. Zhang, C. Wang, and Y. Jin, "An evolutionary algorithm for large-scale sparse multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 380–393, 2019.

[73] I. Kropp, A. P. Nejadhashemi, and K. Deb, "Benefits of sparse population sampling in multi-objective evolutionary computing for large-scale sparse optimization problems," *Swarm and Evolutionary Computation*, vol. 69, p. 101025, 2022.

[74] H. Xu, B. Xue, and M. Zhang, "A duplication analysis-based evolutionary algorithm for biobjective feature selection," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 2, pp. 205–218, 2020.

[75] X.-F. Song, Y. Zhang, D.-W. Gong, and X.-Z. Gao, "A fast hybrid feature selection based on correlation-guided clustering and particle swarm optimization for high-dimensional data," *IEEE Transactions on Cybernetics*, vol. 52, no. 9, pp. 9573–9586, 2022.

[76] Y. Tian, C. Lu, X. Zhang, F. Cheng, and Y. Jin, "A pattern mining-based evolutionary algorithm for large-scale sparse multiobjective optimization problems," *IEEE Transactions on Cybernetics*, vol. 52, no. 7, pp. 6784–6797, 2022.

[77] Y. Tian, C. Lu, X. Zhang, K. C. Tan, and Y. Jin, "Solving large-scale multiobjective optimization problems with sparse optimal solutions via unsupervised neural networks," *IEEE Transactions on Cybernetics*, vol. 51, no. 6, pp. 3115–3128, 2021.

[78] A. Rashno, M. Shafipour, and S. Fadaei, "Particle ranking: An efficient method for multi-objective particle swarm optimization feature selection," *Knowledge-Based Systems*, vol. 245, p. 108640, 2022.

[79] Z. Liu, B. Chang, and F. Cheng, "An interactive filter-wrapper multi-objective evolutionary algorithm for feature selection," *Swarm and Evolutionary Computation*, vol. 65, p. 100925, 2021.

[80] M. Amoozegar and B. Minaei-Bidgoli, "Optimizing multi-objective PSO based feature selection method using a feature elitism mechanism," *Expert Systems with Applications*, vol. 113, pp. 499–514, 2018.

[81] J. Qiu, X. Xiang, C. Wang, and X. Zhang, "A multi-objective feature selection approach based on chemical reaction optimization," *Applied Soft Computing*, vol. 112, p. 107794, 2021.

[82] Y. Zhang, D.-W. Gong, X.-Z. Gao, T. Tian, and X.-Y. Sun, "Binary differential evolution with self-learning for multi-objective feature selection," *Information Sciences*, vol. 507, pp. 67–85, 2020.

[83] K. Demir, B. H. Nguyen, B. Xue, and M. Zhang, "A decomposition based multi-objective evolutionary algorithm with ReliefF based local search and solution repair mechanism for feature selection," in *IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2020, pp. 1–8.

[84] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of ReliefF and RReliefF," *Machine Learning*, vol. 53, no. 1, pp. 23–69, 2003.

[85] A.-D. Li, B. Xue, and M. Zhang, "Multi-objective feature selection using hybridization of a genetic algorithm and direct multisearch for key quality characteristic selection," *Information Sciences*, vol. 523, pp. 245–265, 2020.

[86] P. A. Castro and F. J. Von Zuben, "Multi-objective feature selection using a Bayesian artificial immune system," *International Journal of Intelligent Computing and Cybernetics*, vol. 3, no. 2, pp. 235–256, 2010.

[87] T. Li, Z.-H. Zhan, J.-C. Xu, Q. Yang, and Y.-Y. Ma, "A binary individual search strategy-based bi-objective evolutionary algorithm for high-dimensional feature selection," *Information Sciences*, vol. 610, pp. 651–673, 2022.

[88] F. Cheng, J. J. Cui, Q. J. Wang, and L. Zhang, "A variable granularity search based multi-objective feature selection algorithm for high-dimensional data classification," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 2, pp. 266–280, 2023.

[89] M. Li, S. Yang, and X. Liu, "Pareto or non-Pareto: Bi-criterion evolution in multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 645–665, 2015.

[90] E. Hancer, B. Xue, M. Zhang, D. Karaboga, and B. Akay, "Pareto front feature selection based on artificial bee colony optimization," *Information Sciences*, vol. 422, pp. 462–479, 2018.

[91] A. A. Bidgoli, H. Ebrahimpour-Komleh, and S. Rahnamayan, "Reference-point-based multi-objective optimization algorithm with opposition-based voting scheme for multi-label feature selection," *Information Sciences*, vol. 547, pp. 1–17, 2021.

[92] P. Wang, B. Xue, J. Liang, and M. Zhang, "Differential evolution based feature selection: A niching-based multi-objective approach," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 2, pp. 296–310, 2023.

[93] C. Yue, J. J. Liang, B.-Y. Qu, K. Yu, and H. Song, "Multimodal multiobjective optimization in feature selection," in *IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2019, pp. 302–309.

[94] F. Han, W.-T. Chen, Q.-H. Ling, and H. Han, "Multi-objective particle swarm optimization with adaptive strategies for feature selection," *Swarm and Evolutionary Computation*, vol. 62, p. 100847, 2021.

[95] B. H. Nguyen, B. Xue, P. Andreae, H. Ishibuchi, and M. Zhang, "Multiple reference points-based decomposition for multiobjective feature selection in classification: Static and dynamic mechanisms," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 1, pp. 170–184, 2020.

[96] J. G. Falcón-Cardona and C. A. C. Coello, "Indicator-based multi-objective evolutionary algorithms: A comprehensive survey," *ACM Computing Surveys*, vol. 53, no. 2, pp. 1–35, 2020.

[97] A. Telikani, A. Tahmassebi, W. Banzhaf, and A. H. Gandomi, "Evolutionary machine learning: A survey," *ACM Computing Surveys*, vol. 54, no. 8, pp. 1–35, 2021.

[98] G. Yu, L. Ma, Y. Jin, W. Du, Q. Liu, and H. Zhang, "A survey on knee-oriented multi-objective evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 6, pp. 1452–1472, 2022.

[99] Y. Zhou, J. Kang, S. Kwong, X. Wang, and Q. Zhang, "An evolutionary multi-objective optimization framework of discretization-based feature selection for classification," *Swarm and Evolutionary Computation*, vol. 60, p. 100770, 2021.

[100] Y. Bi, B. Xue, and M. Zhang, "Multitask feature learning as multiobjective optimization: A new genetic programming approach to image classification," *IEEE Transactions on Cybernetics*, vol. 53, no. 5, pp. 3007–3020, 2023.

[101] L. D. Vignolo, D. H. Milone, and J. Scharcanski, "Feature selection for face recognition based on multi-objective evolutionary wrappers," *Expert Systems with Applications*, vol. 40, no. 13, pp. 5077–5084, 2013.

[102] U. Mlakar, I. Fister, J. Brest, and B. Potočnik, "Multi-objective differential evolution for feature selection in facial expression recognition systems," *Expert Systems with Applications*, vol. 89, pp. 129–137, 2017.

[103] R. Guha, M. Ghosh, P. K. Singh, R. Sarkar, and M. Nasipuri, "M-HMOGA: a new multi-objective feature selection algorithm for handwritten numeral classification," *Journal of Intelligent Systems*, vol. 29, no. 1, pp. 1453–1467, 2020.

[104] M. Song, S. Liu, D. Xu, and H. Yu, "Multiobjective optimization-based hyperspectral band selection for target detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–22, 2022.

[105] M. Gong, M. Zhang, and Y. Yuan, "Unsupervised band selection based on evolutionary multiobjective optimization for hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 1, pp. 544–557, 2016.

[106] P. Martín-Smith, J. Ortega, J. Asensio-Cubero, J. Q. Gan, and A. Ortiz, "A supervised filter method for multi-objective feature selection in EEG classification based on multi-resolution analysis for BCI," *Neurocomputing*, vol. 250, pp. 45–56, 2017.

[107] I. Vatolkin, F. Ostermann, and M. Müller, "An evolutionary multi-objective feature selection approach for detecting music segment boundaries of specific types," in *Genetic and Evolutionary Computation Conference (GECCO)*, 2021, pp. 1061–1069.

[108] I. Vatolkin, M. Preuß, G. Rudolph, M. Eichhoff, and C. Weihs, "Multi-objective evolutionary feature selection for instrument recognition in polyphonic audio mixtures," *Soft Computing*, vol. 16, no. 12, pp. 2027–2047, 2012.

[109] G. Khademi, H. Mohammadi, and D. Simon, "Gradient-based multi-objective feature selection for gait mode recognition of transfemoral amputees," *Sensors*, vol. 19, no. 253, pp. 1–23, 2019.

[110] J. S. Dussaut, P. J. Vidal, I. Ponzoni, and A. C. Olivera, "Comparing multiobjective evolutionary algorithms for cancer data microarray feature selection," in *IEEE Congress on Evolutionary Computation (CEC)*, 2018, pp. 1–8.

[111] Z. Zhu, Y.-S. Ong, and J. M. Zurada, "Identification of full and partial class relevant genes," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 7, no. 2, pp. 263–277, 2010.

[112] A. J. Soto, R. L. Cecchini, G. E. Vazquez, and I. Ponzoni, "Multi-objective feature selection in QSAR using a machine learning approach," *QSAR & Combinatorial Science*, vol. 28, no. 11-12, pp. 1509–1523, 2009.

[113] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," *TIK-report*, vol. 103, 2001.

[114] M. Abd Elaziz, Y. S. Moemen, A. E. Hassanien, and S. Xiong, "Toxicity risks evaluation of unknown FDA biotransformed drugs based on a multi-objective feature selection approach," *Applied Soft Computing*, vol. 97, pp. 1–17, 2020.

[115] F. Jiménez, H. Pérez-Sánchez, J. Palma, G. Sánchez, and C. Martínez, "A methodology for evaluating multi-objective evolutionary feature selection for classification in the context of virtual screening," *Soft Computing*, vol. 23, no. 18, pp. 8775–8800, 2019.

[116] Z. Zhou, S. Li, G. Qin, M. Folkert, S. Jiang, and J. Wang, "Multi-objective-based radiomic feature selection for lesion malignancy classification," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 1, pp. 194–204, 2019.

[117] L. A. Kurgan, K. J. Cios, R. Tadeusiewicz, M. Ogiela, and L. S. Goodenday, "Knowledge discovery approach to automated cardiac SPECT diagnosis," *Artificial Intelligence in Medicine*, vol. 23, no. 2, pp. 149–169, 2001.

[118] M. Habib, I. Aljarah, H. Faris, and S. Mirjalili, "Multi-objective particle swarm optimization: theory, literature review, and application in feature selection for medical diagnosis," *Evolutionary Machine Learning Techniques*, pp. 175–201, 2020.

[119] M. Little, P. McSharry, E. Hunter, J. Spielman, and L. Ramig, "Suitability of dysphonia measurements for telemonitoring of Parkinson's disease," *Nature Precedings*, pp. 1–1, 2008.

[120] M. K. Sohrabi and A. Tajik, "Multi-objective feature selection for warfarin dose prediction," *Computational Biology and Chemistry*, vol. 69, pp. 126–133, 2017.

[121] N. Kozodoi, S. Lessmann, K. Papakonstantinou, Y. Gatsoulis, and B. Baesens, "A multi-objective approach for profit-driven feature selection in credit scoring," *Decision Support Systems*, vol. 120, pp. 106–117, 2019.

[122] S. Han, K. Zhu, M. Zhou, and X. Cai, "Competition-driven multimodal multiobjective optimization and its application to feature selection for credit card fraud detection," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 12, pp. 7845–7857, 2022.

[123] E. De la Hoz, E. De La Hoz, A. Ortiz, J. Ortega, and A. Martínez-Álvarez, "Feature selection by multi-objective optimisation: Application to network anomaly detection by hierarchical self-organising maps," *Knowledge-Based Systems*, vol. 71, pp. 322–338, 2014.

[124] W. Wei, S. Chen, Q. Lin, J. Ji, and J. Chen, "A multi-objective immune algorithm for intrusion feature selection," *Applied Soft Computing*, vol. 95, p. 106522, 2020.

[125] M. Roopak, G. Y. Tian, and J. Chambers, "Multi-objective-based feature selection for DDoS attack detection in IoT networks," *IET Networks*, vol. 9, no. 3, pp. 120–127, 2020.

[126] D. X. Sousa, S. Canuto, M. A. Goncalves, T. C. Rosa, and W. S. Martins, "Risk-sensitive learning to rank with evolutionary multi-objective feature selection," *ACM Transactions on Information Systems*, vol. 37, no. 2, pp. 1–34, 2019.

[127] X. Chen, Y. Shen, Z. Cui, and X. Ju, "Applying feature selection to software defect prediction using multi-objective optimization," in *IEEE Annual Computer Software and Applications Conference*, vol. 2, 2017, pp. 54–59.

[128] C. Ni, X. Chen, F. Wu, Y. Shen, and Q. Gu, "An empirical study on Pareto based multi-objective feature selection for software defect prediction," *Journal of Systems and Software*, vol. 152, pp. 215–238, 2019.

[129] B. Huang, B. Buckley, and T.-M. Kechadi, "Multi-objective feature selection by using NSGA-II for customer churn prediction in telecommunications," *Expert Systems with Applications*, vol. 37, no. 5, pp. 3638–3646, 2010.

[130] D. Moldovan and A. Slowik, "Energy consumption prediction of appliances using machine learning and multi-objective binary grey wolf optimization for feature selection," *Applied Soft Computing*, vol. 111, pp. 1–23, 2021.

[131] S. K. Das, R. Mohanty, M. Mohanty, and M. Mahamaya, "Multi-objective feature selection (MOFS) algorithms for prediction of liquefaction susceptibility of soil based on in situ test methods," *Natural Hazards*, vol. 103, no. 2, pp. 2371–2393, 2020.

[132] R. Espinosa, F. Jiménez, and J. Palma, "Multi-surrogate assisted multi-objective evolutionary algorithms for feature selection in regression and classification problems with time series data," *Information Sciences*, vol. 622, pp. 1064–1091, 2023.

[133] M. Kuanr and P. Mohapatra, "Outranking relations based multi-criteria recommender system for analysis of health risk using multi-objective feature selection approach," *Data & Knowledge Engineering*, p. 102144, 2023.

[134] F. BenSaid and A. M. Alimi, "Online feature selection system for big data classification based on multi-objective automated negotiation," *Pattern Recognition*, vol. 110, p. 107629, 2021.

[135] D. Paul, A. Jain, S. Saha, and J. Mathew, "Multi-objective PSO based online feature selection for multi-label classification," *Knowledge-Based Systems*, vol. 222, p. 106966, 2021.

[136] D. Paul, R. Kumar, S. Saha, and J. Mathew, "Multi-objective cuckoo search-based streaming feature selection for multi-label dataset," *ACM Transactions on Knowledge Discovery from Data*, vol. 15, no. 6, pp. 1–24, 2021.

[137] X. Li, M. G. Epitropakis, K. Deb, and A. Engelbrecht, "Seeking multiple solutions: An updated survey on niching methods and their applications," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 4, pp. 518–538, 2016.

[138] Y. Hu, J. Wang, J. Liang, Y. Wang, U. Ashraf, C. Yue, and K. Yu, "A two-archive model based evolutionary algorithm for multimodal multi-objective optimization problems," *Applied Soft Computing*, vol. 119, p. 108606, 2022.

[139] W. Siblini, P. Kuntz, and F. Meyer, "A review on dimensionality reduction for multi-label classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 3, pp. 839–857, 2021.

[140] Y. Zhang, D.-W. Gong, X.-Y. Sun, and Y.-N. Guo, "A PSO-based multi-objective multi-label feature selection method in classification," *Scientific Reports*, vol. 7, no. 1, pp. 1–12, 2017.

[141] G. N. Karagoz, A. Yazici, T. Dokeroglu, and A. Cosar, "A new framework of multi-objective evolutionary algorithms for feature selection and multi-label classification of video data," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 1, pp. 53–71, 2021.

[142] J. Lin, Q. Chen, B. Xue, and M. Zhang, "Multi-task optimisation for multi-objective feature selection in classification," in *Genetic and Evolutionary Computation Conference Companion (GECCO)*, 2022, pp. 264–267.

[143] K. C. Tan, L. Feng, and M. Jiang, "Evolutionary transfer optimization-a new frontier in evolutionary computation research," *IEEE Computational Intelligence Magazine*, vol. 16, no. 1, pp. 22–33, 2021.

[144] N. Zhang, A. Gupta, Z. Chen, and Y.-S. Ong, "Evolutionary machine learning with minions: A case study in feature selection," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 1, pp. 130–144, 2022.

[145] B. Zhang, A. K. Qin, and T. Sellis, "Evolutionary feature subspaces generation for ensemble classification," in *Genetic and Evolutionary Computation Conference*, 2018, pp. 577–584.

[146] K. Chen, B. Xue, M. Zhang, and F. Zhou, "Evolutionary multitasking for feature selection in high-dimensional classification via particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 3, pp. 446–460, 2022.

[147] K. Chen, B. Xue, M. Zhang, and F. Zhou, "An evolutionary multitasking-based feature selection method for high-dimensional classification," *IEEE Transactions on Cybernetics*, vol. 52, no. 7, pp. 7172–7186, 2022.

[148] L. Li, M. Xuan, Q. Lin, M. Jiang, Z. Ming, and K. C. Tan, "An evolutionary multitasking algorithm with multiple filtering for high-dimensional feature selection," *IEEE Transactions on Evolutionary Computation*, 2023, doi:10.1109/TEVC.2023.3254155.

[149] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Computing Surveys*, vol. 48, no. 1, pp. 1–35, 2015.

[150] K. Jha and S. Saha, "Incorporation of multimodal multiobjective optimization in designing a filter based feature selection technique," *Applied Soft Computing*, vol. 98, p. 106823, 2021.

[151] D. Rodrigues, V. H. C. de Albuquerque, and J. P. Papa, "A multi-objective artificial butterfly optimization approach for feature selection," *Applied Soft Computing*, vol. 94, p. 106442, 2020.

[152] Y. Huang, C. Du, Z. Xue, X. Chen, H. Zhao, and L. Huang, "What makes multi-modal learning better than single (provably)," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, pp. 10 944–10 956, 2021.

[153] R. Karthiga and S. Mangai, "Feature selection using multi-objective modified genetic algorithm in multimodal biometric system," *Journal of Medical Systems*, vol. 43, no. 7, pp. 1–11, 2019.

[154] X.-H. Wang, Y. Zhang, X.-Y. Sun, Y.-L. Wang, and C.-H. Du, "Multi-objective feature selection based on artificial bee colony: An acceleration approach with variable sample size," *Applied Soft Computing*, vol. 88, p. 106041, 2020.

[155] Y. Zhou, W. Zhang, J. Kang, X. Zhang, and X. Wang, "A problem-specific non-dominated sorting genetic algorithm for supervised feature selection," *Information Sciences*, vol. 547, pp. 841–859, 2021.

[156] K. Ikeda, H. Kita, and S. Kobayashi, "Failure of Pareto-based MOEAs: Does non-dominated really mean near to optimal?" in *IEEE Congress on Evolutionary Computation*, vol. 2. IEEE, 2001, pp. 957–962.

[157] A. Asuncion and D. Newman, "UCI machine learning repository," 2007, http://archive.ics.uci.edu/ml.

[158] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[159] P. A. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Compution*, vol. 7, no. 2, pp. 174–188, 2003.

[160] E. Zitzler, *Evolutionary algorithms for multiobjective optimization: Methods and applications*. Citeseer, 1999, vol. 63.

[161] F. Cheng, F. Chu, Y. Xu, and L. Zhang, "A steering-matrix-based multiobjective evolutionary algorithm for high-dimensional feature selection," *IEEE Transactions on Cybernetics*, vol. 52, no. 9, pp. 9695–9708, 2022.

[162] B. Tran, B. Xue, and M. Zhang, "A new representation in PSO for discretization-based feature selection," *IEEE Transactions on Cybernetics*, vol. 48, no. 6, pp. 1733–1746, 2017.

[163] R. Said, M. Elarbi, S. Bechikh, C. A. C. Coello, and L. B. Said, "Discretization-based feature selection as a bi-level optimization problem," *IEEE Transactions on Evolutionary Computation*, 2022, doi: 10.1109/TEVC.2022.3192113.