

An efficient discrete rat swarm optimizer for global optimization and feature selection in chemoinformatics

Essam H. Houssein ^{a,*}, Mosa E. Hosney ^b, Diego Oliva ^c, Eman M.G. Younis ^a, Abdelmgeid A. Ali ^a, Waleed M. Mohamed ^a

^a Faculty of Computers and Information, Minia University, Minia, Egypt

^b Faculty of Computers and Information, Luxor University, Luxor, Egypt

^c Depto. de Innovación Basada en la Información y el Conocimiento, Universidad de Guadalajara, CUCFI, Guadalajara, Jal, Mexico



ARTICLE INFO

Article history.

Received 20 May 2022

Received in revised form 31 May 2023

Accepted 5 June 2023

Available online 8 June 2023

Keywords.

Machine learning

Metaheuristic algorithms

Feature selection

Quantitative structure–activity relationship

Transfer functions

ABSTRACT

Machine learning algorithms need feature selection (FS) as a significant step towards filtering unnecessary data. This paper proposes a wrapper FS approach that combines the rat swarm optimization (RSO) algorithm with genetic operators to avoid local optimal. In the proposed approach the transfer functions (TFs) are added to balance local and global search by converting a continuous search space into a discrete space. Eight variants of the bmRSO algorithm were applied for classification purposes using a support vector machine (SVM) to increase accuracy and decrease the number of features over several chemical datasets. The eight bmRSO proposed methods and the original RSO were evaluated using the CEC'20 test suite and twelve datasets (eight chemical and four toxicity effect datasets) to verify their performance in complex optimization problems and FS over real datasets, respectively. Moreover, the binary versions of other stable metaheuristic algorithms such as Harris Hawks Optimization (HHO), Grey Wolf Optimization (GWO), Farmland Fertility Algorithm (FFA), Artificial Gorilla Troops Optimizer (GTO), African Vultures Optimization Algorithm (AVOA), Runge Kutta Optimizer's (RUN), and Slime Mould Algorithm (SMA) were used to compare the results obtained by the best variant of the bmRSO. Eventually, the experimental results have revealed that in most of the tests, the proposed bmRSO1 has achieved efficient search results with higher convergence speeds without increasing additional computational efforts. From the twelve datasets, the MAO dataset reached the highest results compared with other datasets, so the proposed method, bmRSO1-SVM, achieved an accuracy of 98.201% and a 20.001 number of selected features.

© 2023 Elsevier B.V. All rights reserved.

1. Introduction

Cheminformatics is the prediction of molecular features and analysis related to diseases [1]. It uses a database for molecular data such as pharmacogenomics. Compounds interacting with disease receptors have been developed to decrease or deactivate these effects. In these applications, the modeled search space is a critical problem because it grows exponentially as the features in the datasets. The drug development process is a chemoinformatics trend described as rational drug design. Thus, we aimed to find new drugs that could be biological targets [2]. The drug development approach is indicated through the identification and validation of a drug. It is essential to identify the basics of cellular processes, predict protein structures, and how a

chemical compound interacts with normal biological molecular targets. In this regard, a drug is a small organic molecule that activates a disease effect. Drugs can be made by two important methods: docking and quantitative structure–activity relationship (QSAR) [3]. Docking is done by bidding between receptors and ligands, whereas QSAR is based on a mathematical form for calculation descriptors [4,5].

The main task of the drug development process is the toxicity study because it is necessary to avoid drug failures in clinical trials. The Food and Drug Administration (FDA) estimates that computational techniques might save 100 million dollars per drug [6]. Drug toxicity refers to the fatal effect of a drug on the entire organ. However, the best bioassays are costly and time-consuming, making them unsuitable for large-scale toxicity testing in the early phases of drug development. Thus, automated approaches avoid *in vivo* or *in vitro* toxicity disadvantages. Furthermore, computational models have been applied as alternatives to experimental materials and animals. Chemical toxicity compounds are determined automatically, thereby classifying the effects of most

* Corresponding author.

E-mail addresses: essam.halim@mu.edu.eg (E.H. Houssein), mosa94@luxor.edu.eg (M.E. Hosney), diego.oliva@ceci.udg.mx (D. Oliva), eman.younas@mu.edu.eg (E.M.G. Younis), a.ali@mu.edu.eg (A.A. Ali), waleedmakram@mu.edu.eg (W.M. Mohamed).

toxicity as mutagenic, irritating, tumorigenic, reproductive, and neurotoxic [7]. These are the most popular toxicity risks that are essential benchmarks for the DataWarrior open-source tool [8]. Many models for computational toxicity, such as Case DEREK and Multicase, were used. The main task of these models is to predict chemical compound toxicity and toxicological endpoints.

Optimization is used to solve many problems by maximizing or minimizing the solutions. Metaheuristic algorithms (MAs) are optimization tools in which different methodologies are used to increase the effectiveness of search processes [9–12]. Although finding the exact solution is difficult in most cases, MA could give the optimal global best [13,14]. Depending on the search space, optimization can be described as continuous, discrete, and binary [15]. The solutions can be divided into three groups: 1- Continuous variables, which are constantly changing and have a continuous range of values; 2- Discrete variables, which can take variables with either an integer or a binary value; 3- Mixed variables, which can have many real, or integers, thereby making it a mixed problem. Most optimization algorithms use a continuous search space that causes a local search problem. Sometimes, it is necessary to convert the continuous search space to discrete. Several optimization problems, such as manufacturing, use discrete variables. For example, discrete variables are integers or binary values in production lines.

Data mining is increasingly used to extract essential data and predict the best results. Feature selection (FS) is a critical step in data mining for filtering unnecessary features and selecting only a representative subset from the given training data [16]. Feature selection is used as a preprocessing classification. Its benefits include reducing model creation time, avoiding overfitting problems, and enhancing model generalization for several medical issues, such as medical imaging and drug development [17]. Feature size can be obtained from medical resources; thus, more literature has shown that data from medical applications use FS. Many FS strategies are classified as wrapper, filter, and embedded methods.

Filter methods can evaluate the feature subset using an objective function depending on the feature function. However, wrapper strategies search for a subset of features depending on the specific classifier's performance. A hybrid FS approach is a combination of filter and wrapper methods. An example of hybrid FS is ridge regression, which has built-in penalization. FS is considered an NP-hard problem with many possible solutions, especially for a large feature space [18]. Several MAs methods have a binary version for solving problems, such as wrapper techniques for FS [19,20].

Feature selection is increasingly used in many fields as bioinformatics with sequence and signal analysis, which are the main challenges of FS [16]. Content analysis checks the sequence properties and productivity for protein coding. Recently, many applications have been built using machine learning (ML) algorithms, especially classification methods, such as genomics research, cancer detection, medical diagnostic applications, and computational chemistry are just a few examples. Many attribute numbers in these data affect the dataset performance. However, most of these features are redundant, irrelevant, and distracting. Feature selection strategies have been applied to declare the influence of such features by selecting the significant ones and deleting the others. This improves classification performance and reduces data computation and processing time.

The rat swarm optimizer (RSO) is a MA that can solve many real-world problems in engineering applications [21]. The algorithm performs excellently in different search spaces, such as real variable single-objective and multi-objective optimization problems. Its behaviors were tested and showed many flaws that affect its quality. The standard RSO suffers drawbacks such as;

(1) its exploitation is weaker than its exploration ability, and (2) it is trapped in local solutions. Therefore, this paper proposes a modified version of RSO that uses genetic operators and eight Transfer Functions (TFs) called bmRSO to convert the search space from continuous to discrete [22]. With the genetic operators and TFs, bmRSO can avoid local optimal and control the exploration and exploitation of the search space.

Besides, eight variants of bmRSO based on eight TFs and the original RSO algorithm were evaluated on the IEEE CEC'20 test suite and some chemical datasets. Transfer functions were used to boost the bmRSO with a Support Vector Machine (SVM) for solving binary problems with high accuracy. Transfer functions are used to convert velocity into probability [23]. We also presented a wrapper FS approach that uses a classifier (SVM) [24]. Eight datasets were considered for testing the proposed FS algorithm. For FS, TFs were used to modify RSO with the operators, and the genetic operators were used to control local and global searches. Results show that bmRSO is balanced regarding local and global search and the stagnation in suboptimal solutions is avoided. The experimental results indicate that the boosting methods outperform the original algorithm and can be applied to real FS applications.

The results obtained by bmRSO are compared with eight variants of bmRSO and the original RSO algorithm. After that, some binary metaheuristic algorithms such as Harris Hawks Optimization (HHO) [14], Grey Wolf Optimization (GWO) [25], Farmland fertility Algorithm (FFA) [26], Artificial Gorilla Troops Optimizer (GTO) [27], African Vultures Optimization Algorithm (AVOA) [28], Runge Kutta Optimizer (RUN) [29], and Slime Mould Algorithm (SMA) [30] are used in the comparison. The proposed bmRSO method is enhanced using two mechanisms to avoid the most popular problems of local search and to balance between local and global search, so the main contributions of this paper that make our methods differ from others are the following:

1. We proposed different and efficient modifications of RSO.
2. The proposed bmRSO introduces the mechanisms of genetic operators (crossover, mutation, and selection) to enhance RSO's performance by increasing the diversity of the population and avoiding local search.
3. Transfer functions were applied to convert continuous search spaces to binary search spaces to balance local and global search space.
4. The proposed bmRSO was adopted to address the ten global optimization tasks from the CEC'20 test suite and was compared with variants of bmRSO and the original RSO algorithm.
5. The developed bmRSO was transformed into a binary model for tackling FS problems for the first time. Furthermore, the bmRSO versions were evaluated using several datasets in chemoinformatics. The proposed models were applied to classification using SVM to increase accuracy and decrease the number of features in different datasets to propose the best chemical compound for making a suitable drug.
6. The best bmRSO version result is compared with other MAs.

The rest of the paper is organized as follows: Section 2 presents the related work and highlights several recent studies. Section 3 introduces the preliminaries of RSO, genetic operators, quantitative structure–activity relationship (QSAR), and Support Vector Machine (SVM). Section 4 discusses the proposed bmRSO algorithm. Section 5 explains the counterparts' results over the CEC'20 test suite. 6 presents the bmRSO for solving FS problems with two experiment results for chemical and toxicity data. Section 6.4 explains several binary metaheuristic algorithms compared with the

best version of bmRSO-SVM over several datasets. Section 7 discusses the results, advantages, and disadvantages of the proposed bmRSO. Finally, Section 8 concludes the paper and introduces future work directions.

2. Related work

The related literature on cheminformatics is vast, with a huge number of possible ML applications. This section will analyze previous work in this domain. For example, the authors of [3] created medical molecules, diagnosed several diseases, and determined suitable drugs. However, one major challenge is collecting medical data. First, ZINC and protein bank databases were used to select suitable crystals for the protein structure of medical molecules. Here, CADD is the most efficient chemical method.

The authors of [31] proposed a method that can be used for drug design identification. QSAR and docking are the most important tools for CADD and chemical compounds found on the Pubchem website. In [32], docking is the binding protein and ligand. Proteins and ligands are related and must be separated. The optimal drug based on the best ligands with less energy and Pymol software [33] is used for the separation. The energy calculation is conducted using AutoDock software.

In [34], sentiment analysis was used to predict user sentiments on the effectiveness of specific drugs based on user reviews. To illustrate the commonalities across domains, transfer learning methodologies are offered. Here, classification-based sentiment analysis is used as an n-grams approach to conclude prediction tasks to multiple user reviews. The classification model of this dataset has an accuracy of 92.24%.

The authors of [35] presented a method that depends on biclustering and was applied to reduce the number of molecular descriptors to predict chemical compound biodegradation. Three classifiers were used to evaluate the biodegradation task: Random Committee (RC), Neural Network (NN), and Random Forest (RF). The experimental results show that the best classifier was RF, which had 88.81% accuracy with only 19 Molecular Descriptors (MD) for the QSAR biodegradation dataset. Thus, we can see that artificial intelligence significantly improved the development of many horizons depending on ML for QSAR modeling [36,37].

Many opportunities have appeared in the FS field for selecting significant features. In [38–41], FS is defined by four steps (1) choose the suitable features, (2) analyze the subset using various metrics, (3) identify other sets, and (4) feature validation. Other FS methods [42] include wrapper, filter, and hybrid techniques. However, the wrapper method produces more accurate results than the filter approach, although it is more time-consuming. The hybrid approach integrates two methods. The quality of each set of features is evaluated using the fitness function of MAs [43], which is an integral part of FS and ML algorithms. The success of such systems highly depends on the relevance of features to the target domain.

Many MAs [44–46] are applied for solving the FS problem, such as Search and Rescue optimization algorithm (SAR) that is hybridized with k-NN, and a wrapper FS method [47]. It is applied to minimize the search space size, find the best subset features, and increase classification accuracy. In the same context, Henry Gas Solubility Optimization (HGSO) is combined with a boosted Harris Hawks optimization based on Heavy-tailed distributions to improve the search process [48]. This approach has been applied to FS problems for some chemical and machine learning repository UCI datasets. In this study, a developed (MAs) technique addresses the FS problem and avoids the demerits of classic FS methods. Thus, more hybrid MA techniques are increasingly being developed for this purpose and to improve the quality of results [48].

Assessing the risk aspects of unknown biotransformed medicines in drug development is critical [6,7]. However, the experimental methods used to complete this work are time-consuming and computationally expensive, making them unsuitable for evaluating a large dataset of compounds early in drug development. Computational techniques can be used to predict the risk aspects of unknown biotransformed medications to avoid these problems. Toxicity data were extracted from 5909 pharmaceuticals with 31 chemical descriptors. Toxicology testing is an important process in drug development. Thus, computational models were applied to the toxicity effects of the proposed medications. It was used a dataset containing four toxicity effects, mutagenic, tumorigenic, irritating, and reproductive impacts. Here whale swarm optimization with SVM was applied to make the prediction.

Another technique for predicting medication development is a rough set FS [5,40]. Crude set-based approaches pick the most discriminative characteristic traits during the FS phase. Using different rough set-based methods, such as Quick Reduct Feature Selection (QRFS), Discernibility Matrix-Based Feature Selection (DMFS), and Entropy-Based Feature Selection (EBFS), many features from the feature vector were selected in this step. These algorithms aim to reduce the number of features, thereby speeding up the classification and improving accuracy.

A binary version of the Particle Swarm Optimization (PSO) has been introduced for the binary problems [49]. TF is applied here to map a continuous to discrete search space. Since it is the most important aspect of the binary version, six new TFs were used and assessed in this study. They were split into two families: S and V-shaped. In the same context, in [22], the Binary Dragonfly Algorithm (BDA) was used to develop a wrapper FS algorithm. The TF, which converts a continuous search space into a discrete space, is the most crucial part of the BDA. This study integrated eight TFs into BDA and evaluated them using eighteen benchmark datasets from the UCI data repository, divided into two families (S and V-shaped functions). The proposal of time-varying S and V-shaped TF to leverage the impact of the step vector on balancing exploration and exploitation is the key contribution of this study.

There is no certainty that all classification problems will have the best solution (for example, a subset of features) due to the stochastic nature of MAs. Additionally, according to the No-Free-Lunch (NFL) theorem for optimization, no single optimization strategy can solve every optimization problem [50]. As a result, the performance of the current stochastic-nature FS techniques may be degraded when used for particular optimization tasks.

This motivated us to suggest Rat Swarm Optimization (RSO) enhancements and use them for FS in chemical data classification. Even though RSO has excelled in various applications, it is still prone to local optimal stagnation when dealing with high-dimensional tasks. Therefore, developing new mechanisms that might significantly enhance or maintain a better trade-off between exploration and exploitation is desirable. This paper proposed a new version of the RSO called bmRSO algorithm. The proposed approach was improved by two schemes, namely, genetic operator and TFs, to evolve the global search and local search during optimization to enhance the search tendency in the original RSO. In the first strategy, the genetic operators are hybridized with RSO to obtain more diversity find the population and to avoid local optimal. For the second strategy, TFs are added to convert continuous to discrete search space. The convergence control parameter (cp) is integrated to govern the transition from exploration at the start of the optimization process to exploitation at the end. The proposed bmRSO-based FS generates subsets of features evaluated using the SVM as an internal classifier. Moreover, the proposed bmRSO is validated on twelve challenging chemical datasets. bmRSO1-SVM outperformed other wrapper FS techniques, such as different bmRSO-SVM versions and binary MAs algorithms (bHHO, bRUN, bSMA, bAVOA, bFFA, bGWO, and bGTO).

3. Preliminaries

This section explains the steps of the standard version of the Rat Swarm Optimization (RSO). Besides the methodology, the Quantitative Structure–Activity Relationship (QSAR) is also discussed. Finally, the description of a Support Vector Machine (SVM) is also provided.

3.1. Rat swarm optimization

Rats are medium-sized rodents with long tails and are different in size and weight [21]. Black and brown rats are the two most common rat species. Depending on the gender, they are called bucks and does, respectively. Generally, rats are sociable animals by nature. They groom each other and participate in jumping, chasing, tumbling, and punching. They are territorial creatures that live in groups of bucks and does. Their behavior is highly aggressive in certain conditions, resulting in several deaths. Here, we study their aggressive behavior while chasing and fighting with prey.

3.1.1. Mathematical model and optimization

Here, the behaviors of rats, such as chasing and fighting, are examined, and the proposed RSO algorithm is then described.

Chasing the prey. Rats are gregarious creatures that use their social agnostic behavior to chase their prey in groups. The mathematical model defines the behavior, and we suppose that the optimal search agent is aware of the location of the prey. The other search agents can revise their places considering the best search agent thus far. This technique proposes equations from Eq. (1) to Eq. (4):

$$\vec{P}_i = A \cdot \vec{P}_i(x) + C \cdot (\vec{P}_r(x) - \vec{P}_i(x)) \quad (1)$$

where $\vec{P}_i(x)$ denotes the rat position, $\vec{P}_r(x)$ denotes the optimal rat position. However, the following formula is used to determine the A and C parameters:

$$A = R - x \times \left(\frac{R}{\text{Max}_{\text{Iteration}}} \right) \quad (2)$$

Where $x = 0, 1, 2, \dots, \text{Max}_{\text{Iteration}}$

$$C = 2 \cdot \text{rand} \ 0 \quad (3)$$

As a result, R and C are both random numbers in the range of $[1, 5]$ and $[0, 2]$ respectively. Over the path of iterations, the parameters A and C are responsible for better exploration and exploitation.

Fighting with prey. Eq. (4) has been developed to define the fighting process of rats with their prey mathematically:

$$\vec{P}_i(x+1) = \left| \vec{P}_r(x) - \vec{P}_i \right| \quad (4)$$

$\vec{P}_i(x+1)$ determines future position. It saves the best solution and repeatedly updates the locations of other search agents to find the best search agent. The effect of the equations is used from Eq. (1) to Eq. (4). Here, the position can be updated by adjusting the parameters, Eqs. (2) and (3) are used for several positions that can be obtained using the current position. This concept, however, can be expanded to an n -dimensional environment. Thus, the altered values of parameters A and C ensure exploration and exploitation. These RSO equations help find the optimal solution with the fewest operators.

3.2. Quantitative structure–activity relationship

Quantitative structure–activity relationship (QSAR) has the fundamentals of several cellular processes and determines the connections between normal biological molecules [51]. It is a critical tool in the early stages of drug development and can be applied to evaluate the effects of materials, chemicals, and materials on human and environmental health. Consequently, QSAR has grown in popularity in recent years because it can analyze datasets containing large compound numbers and different molecular structures. The QSAR model's primary goal is to discover a link between the molecular descriptors of drugs and their biological activity.

The QSAR approach has been developed in various ways. It may be divided into two categories: traditional methods, such as principal component analysis (PCA), and nontraditional methods, such as partial least-squares. The quality of the final result is affected by various restrictions in these procedures. The connections between the independent and dependent descriptors cannot be comprehended. The nonlinear group, which comprises k-NN and Artificial Neural Networks (ANNs), determines nonlinear mappings. However, when the dataset contains many features, the performance of these QSAR groups reduces. The authors of [52] developed a PSO-based QSAR model in which the fitness function is derived using the Bayesian information criterion (BIC). A QSAR model was proposed in [35] that focuses on bi-clustering analysis as a postprocessing stage for FS outputs acquired using a combination of FS approaches. In [35], to improve the prediction of influenza neuraminidase, they constructed a QSAR model (H1N1). The GA-PLS model, which is a mix of the Genetic Algorithms (GA) and partial least squares, is used as an FS method to check the relevant descriptor in the model. Furthermore, SVM based on GA predicts activity depending on the selected descriptors. In [53], to improve the prediction of H1N1 activity, the penalized (PSVM) was proposed as an FS and classification method. Then, the PSVM with adaptive L1-norm is used in this model to assign a weight to each descriptor that shows its importance.

3.3. Support vector machine

Support Vector Machine (SVM) is a machine learning (ML) classifier based on mapping data using kernel functions to get the optimal solution [54]. SVM is a linear model for several classification or regression problems. It is done by drawing a line to separate the data into classes. It increases by several margins to the nearest point; the optimal line is the algorithm's output. Some parameters influence SVM results as tuning parameters, which are the parameters that are defined when the classifier is designed. The C parameter governs the balance between a smooth decision boundary and correctly identifying training points. Meanwhile, gamma determines the range of influence of a single training session.

We choose to utilize SVM as the classifier in our study over other classifiers due to the following reasons:

- Effective in high-dimensional data: SVMs have demonstrated strong performance in high-dimensional data where the number of features (dimensions) is large.
- Interpretability: SVMs offer interpretability, as the decision boundary is defined by a subset of the training data called support vectors. This feature enables researchers and domain experts to gain insights into the key features and their contributions to the classification process, which can be crucial in biomedical research.

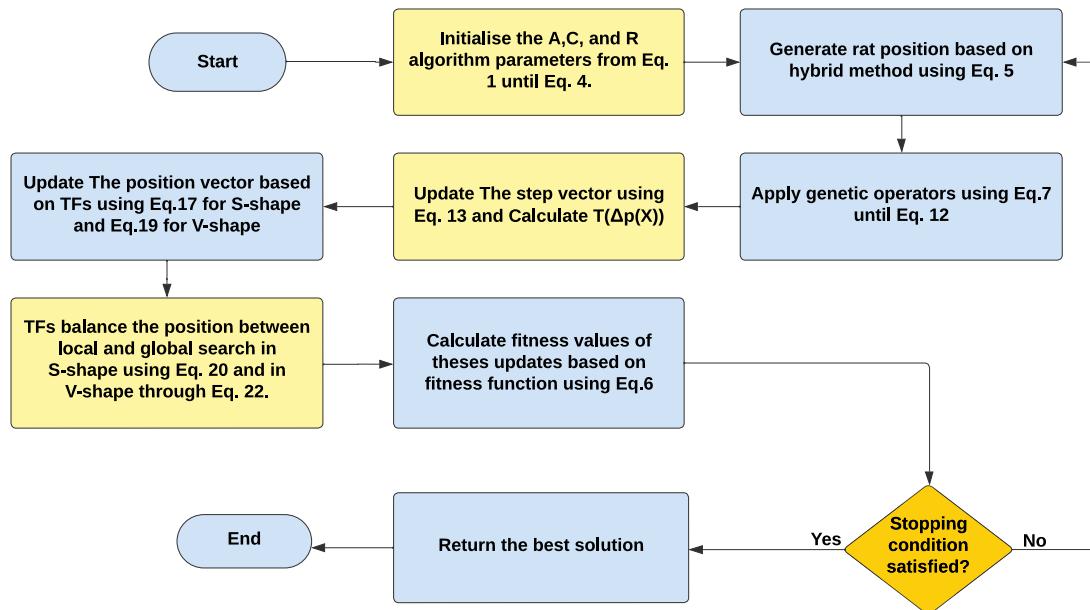


Fig. 1. The proposed bmRSO algorithm.

- Robust generalization: SVMs are known for their ability to generalize well on unseen data. By maximizing the margin between classes, SVMs aim to achieve better separation and minimize overfitting, thus enhancing the classifier's performance on future data instances.

SVM is used increasingly in cheminformatics, so SVM applications in this field predict toxicity-related qualities such as mutagenic toxicity. The SVM step is presented in Algorithm 1.

Algorithm 1 SVM steps

```

Inputs: The splitting data(training and testing) is loaded.
Outputs: An accuracy is calculated.
The tuning parameters are chosen.
while (stopping condition is not met) do
  Train data is built for each data point.
  Testing data is classified
end while
Return accuracy
  
```

4. The proposed bmRSO

This section combines the RSO algorithm with genetic operators and TFs to build a new bmRSO version. The main stages of bmRSO are illustrated in Fig. 1. Specifically, the bmRSO includes three stages: (1) Search agents are controlled, (2) Search agents are produced for the hybrid method, and (3) The new population based on the previous step is updated.

In the update step, all RSO search agents control each position. Here, the genetic operators produce more populations to avoid local optimality. However, they affect RSO positions. Thus, TFs balance local and global search space and the capacity to use the inherent superior convergence. Creating a new hybrid search agent requires creating a new hybrid individual from the current RSO, as shown in Eq. (5).

$$p(X_{\text{hybrid}}) = \alpha \times p(X_i) + (1 - \alpha) \times S_i \quad (5)$$

Where α value between $[0, 1]$ indicates RSO weight, $p(X_i)$ is RSO position, and S_i represents RSO individual position based on the genetic operators and TFs, so this Eq. (5) indicates the effects

of them. The search agent is updated using a greedy selection to select the optimal candidate solution. To improve the quality of solutions and performance of the proposed algorithm.

The reasons that motivate us to present an alternative version of RSO called the bmRSO algorithm will be described in the following subsections.

4.1. Shortcomings of the original RSO

Fast convergence and robust exploitation over the space of optimization problems are the benefits of RSO. However, throughout the test experiments, various issues with its performance surfaced. For example, although the original RSO algorithm has a fast convergence during exploration of the search space, in which the solutions in each iteration move quickly towards the optimal solution found at the current time, this solution to which the algorithm converges may not be the best over the entire search space. Furthermore, rather than the convergence of the algorithm being self-tuned to encourage proposed solutions to explore the remaining regions in the search space, the RSO convergence rate lacks self-tuning, causing the algorithm to have a premature convergence problem. This leads to further issues, such as the algorithm getting stuck in local optima areas and preventing it from identifying the closest/best solutions. When handling complex optimization problems, the original RSO algorithm, like other MAs, cannot adequately explore the whole search space in its current state [55,56].

However, genetic operators control the solution from local to global using Eqs. (6) and (11) to enhance the position and permit the trapping in local search, thereby increasing the variety of algorithms and avoiding local solutions compared between a new and previous one. For more diversity, genetic algorithm operators are used to integrate two mutant vectors, namely y_{Mut} and z_{Mut} , to generate a new child w_{Cross} as described in Eq. (10). Offspring fitness value: y_{Mut} , z_{Mut} and w_{Cross} based on the selection operator Eq. (11) were compared to get the best prey p_x . The search space was boosted by exploring new regions to identify the best candidate solution so several positions are updated by Eqs. (16) and (17). Genetic operators and TF modifications were combined with RSO to produce a new version called bmRSO. The best new solution is updated using Eq. (5) for $p(x_i)$, therefore, TFs balance the position between local and global search

using Eq. (21) to control the next position. The stop condition is the number of maximum iterations that permit evaluating the bmRSO algorithm's performance. Upon completing the bmRSO process, the best solution P_x is returned. Algorithm 2 declares how the best solution is proposed by calculating the fitness function using Eq. (22) for many new populations. The process is stopped if the determined condition is reached. However, neither the algorithm performance nor the optimization problem space is affected.

Additionally, plots of convergence curves in most testing functions in Fig. 4 show a slow divergence of the original RSO during exploitation at most iterations, demonstrating that the approach is trapping in local rather than global regions.

4.2. Architecture of the proposed bmRSO

The primary goal of this research is to design a new RSO version that addresses the issues stated in the previous section and the faults in the original RSO algorithm. Our study did not alter the foundations of the RSO algorithm. However, it boosts the RSO algorithm by increasing the exploration duration and maintaining diversity using the genetic operator strategy, TFs, and linear population reduction. The mathematical description and steps of the bmRSO are formulated in Algorithm 2. Like the original RSO algorithm, the algorithm starts with a randomly selected population of size $2N$. In the modified bmRSO, the population solutions are evolved first, followed by the individual phase from the original RSO. The linear population reduction is used at each iteration to manage population diversity, and the same approach is repeated until the optimization is complete. The following subsections illustrate the two injection operators: genetic and transfer functions.

4.2.1. Genetic operators

Several algorithms use evolutionary operators, especially the two basic algorithms, differential evolution and genetic algorithms. Some examples of such operators are mutation (bit inversion), crossover (single point crossover), and selection.

Mutation: The mutation operation is built using the results of RSO tasks as the solution goal ($P(x)$). A number between 0 and 1 is produced at random for each component. The target agent element ($P(x)$) is considered when the value reaches the mutation rate (ζ). If this value is less than the mutation rate (ζ), the old vector is replaced with a component of the y or z vectors. The mutation operator is determined by applying the following formula:

$$y_{\text{Mut}} = \begin{cases} p_{(x)} & \text{if rand } 1 \geq \zeta \\ y & \text{else} \end{cases} \quad \text{and} \\ z_{\text{Mut}} = \begin{cases} p_{(x)} & \text{if rand } 2 \geq \zeta \\ z & \text{else} \end{cases} \quad (6)$$

$$\text{Where: } \begin{cases} \zeta = \frac{t}{T}; \\ y = |p(x) - p(x)_t^i|; \\ z = y - r_v \end{cases} \quad (7)$$

$$y_{\text{Mut}} = \begin{cases} p_{(x)} & \text{if } \rho_1 \geq \zeta \\ y & \text{else} \end{cases} \quad \text{and} \\ z_{\text{Mut}} = \begin{cases} p_{(x)} & \text{if } \rho_2 \geq \zeta \\ z & \text{else} \end{cases} \quad (8)$$

$$\text{Where: } \begin{cases} \zeta = \frac{t}{T}; \\ y = |p(x) - p(x)_t^i|; \\ z = y - r_v \end{cases} \quad (9)$$

where i th dimension is known by lb^i and ub^i , r_v have D components generated by randomly between $(0, 1)$.

Crossover: The crossover is the combining of two individuals to produce more variety. To generate a new offspring w_{Cross} , a linear combination with random number τ is used.

$$w_{\text{Cross}} = y_{\text{Mut}} + (\tau) * (z_{\text{Mut}} - y_{\text{Mut}}). \quad (10)$$

Compared to other types of operators like linear recombination, this type of operator enables children to inherit more information from their parents.

Selection: The selection type used in RSO is greedy selection based on differential evolution. When evolution functions (mutation and crossover) are accessed, offspring are created. The child and parent performances are then compared to determine the best. Finally, if the parent's performance is good, they can stay in the population. The rule that defines greedy selection is as follows:

$$p_{x+1}^i = \begin{cases} y_{\text{Mut}} & \text{if } \text{fit}(y_{\text{Mut}}) < \text{fit}(P(x)^i) \\ z_{\text{Mut}} & \text{if } \text{fit}(z_{\text{Mut}}) < \text{fit}(P(x)^i) \\ w_{\text{Cross}} & \text{if } \text{fit}(w_{\text{Cross}}) < \text{fit}(P(x)^i) \end{cases} \quad (11)$$

4.2.2. Transfer functions

Transfer functions (TFs) convert the search space from continuous to discrete. Then, the velocity performance is improved based on a suitable TF method. First, several searches employ TF methods, considered independent of the algorithm, and do not affect algorithmic search behavior. Second, adding the TFs does not change the algorithm's computational complexity, as this component is run only once for each solution and in each iteration. Third, local and global searches can be enhanced using TFs. The drawbacks of TFs are their component nature and population algorithm operators; they are the only components that help in exploration and exploitation. RSO needs to choose an appropriate balance between diversification and intensification. During the early phases of the selection process, exploration is more necessary than intensification to check the essential parts of the feature space. However, exploitation becomes increasingly critical in subsequent stages because we need to boost the probability of obtaining better solutions like those found earlier.

TFs are divided into groups based on their forms (S and V-shaped) [57]. The V-shaped TFs are better than the S-shaped TFs because the agents need not be 0 or 1 to solve problems with the bmRSO algorithm. TFs are generally used as input parameters to determine the prediction of changing the element's position to 0 or 1 depending on the value of the i th search agent's step vector (velocity) in the d th dimension of the current iteration (t). Eq. (12) indicates the rat movement direction using a step vector to achieve several exploration and expiation behavior during optimization.

$$\Delta p(X_{i+1}) = (aA_i + cC_i) + p(\vec{X}_i) \quad (12)$$

Where A and C are rat parameters.

According to [58], Eq. (13) is the prediction of transforming continuous positions using the TF.

$$T(v_d^i(t)) = \left| \left(v_d^i(t) \right) / \sqrt{1 + (v_d^i(t))^2} \right| \quad (13)$$

Where $T(v_k^i(t))$ is a result that obtained from Eq. (13) then is applied to convert the i th position vector element to (0 or 1) according Eq. (14)

$$p(X_{t+1}) = \begin{cases} -p(X_t) & r < T(v_k^i(t)) \\ p(X_t) & r \geq T(v_k^i(t)) \end{cases} \quad (14)$$

This equation is described through the random number r between $[0, 1]$. Movement size is determined by the step vector, which indicates the current individual's movement. A decreasing step vector value declares that the agent is near finding the optimal solution and should take fewer steps (exploitation). However,

when the step vector value is high, the agent is not optimal and will require more updates (exploration) [59].

In this method, if the step vector is used to determine the probability of changing positions, TFs strongly influence the balance between global and local search. The probability is determined similarly if the TF remains constant during optimization. Using the effects of the step vector effect on position changes, changing the TF allows exploration and use of the search space. TF Eq. (15) is proposed for the S-shaped family [22].

$$T(v_i^k(t)) = \frac{1}{1 + e^{-v_i^k(t)}} \quad (15)$$

$$p(x_i^k(t+1)) = \begin{cases} 1 & \text{rand} < T(v_i^k(t+1)) \\ 0 & \text{rand} \geq T(v_i^k(t+1)) \end{cases} \quad (16)$$

Where, $v_i^k(t)$ is vector of the step. The i th individuals at the t th iteration in the k th dimension. Each vector element representing the current individual will be updated according to Eq. (16) depending on the probability value $v_i^k(t)$, obtained from Eq. (15).

V-shaped TF named V_{T1} is converted as in Eq. (17):

$$T(\Delta p(X)) = \begin{cases} 1 - \frac{2}{1+e^{-2p(X)}} & p(X) \leq 0 \\ \frac{2}{1+e^{-2p(X)}} - 1 & p(X) > 0 \end{cases} \quad (17)$$

where $p(X)$ is the current position is modified in Eq. (18) according to the probability value calculated from Eq. (17). In [60], a recent comprehensive study indicated the impact of S or V-shaped TFs on the efficiency and ultimate results of binary PSO with TFs.

$$p(X)_{id}^{k+1} = \begin{cases} 0 & r \leq T(\Delta p(x_{t+1})) \text{ and } \Delta p(x_{t+1}) \leq 0 \\ 1 & r \leq T(\Delta p(x_{t+1})) \text{ and } \Delta p(x_{t+1}) \geq 0 \\ p(X)_{id}^k & r > T(\Delta p(x_{t+1})) \end{cases} \quad (18)$$

Where r value between 0 and 1.

Conventional TFs show that the flipping problem of an element is that the d th dimension of the solution dimension is high when the inputs' absolute values are small, and vice versa. Consequently, in the early phases of the optimization process, a small absolute value of the step vector (around zero) is desired to explore the search space best. However, in the later phases of the optimization process, a step vector with a large absolute value is advised for the best exploitation of the search space. Consequently, the optimizer is more likely to search for good solutions near those found during exploration. The inability of a TF to transform step vector values to acceptable without changing their shape during the optimization process, probability values cause all previous problems.

A new TF model is built as indicated in Eqs. (19), (20) and (21). The proposed TFs use time-varying TFs to achieve a good balance between global and local search [60]. To overcome the stability between local and global in an S-shaped family, Eq. (19) is used instead of Eq. (15), and also in the case of the V-shape Eq. (21) is applied as an alternative to Eq. (17).

$$T(v_i^k(t), \tau) = \frac{1}{1 + e^{-\frac{v_i^k(t)}{\tau}}} \quad (19)$$

Where τ represents a TFs variable with an initial value and gradually reduces over iterations as shown in Eq. (20).

$$\tau = \left(1 - \frac{t}{T}\right) \tau_{\max} + \frac{t}{T} \tau_{\min} \quad (20)$$

Where τ_{\min} and τ_{\max} are the min and max values of the control parameter τ , T is the maximum iterations number.

The main idea for the newly suggested TF is its value can be linearly increased as the step vector of search agents increases.

Table 1
Transfer functions equations.

Function name	Function formula
s-shape function	
TV_{S1}	$T(x, \tau) = \frac{1}{1+x^{\frac{-2\tau}{\tau^2}}}$
TV_{S2}	$T(x, \tau) = \frac{1}{1+x^{\frac{-\tau}{2\tau}}}$
TV_{S3}	$T(x, \tau) = \frac{1}{1+x^{\frac{-\tau}{2\tau}}}$
TV_{S4}	$T(x, \tau) = \frac{1}{1+x^{\frac{-\tau}{3\tau}}}$
V-shape	
TV_{v1}	$T(x, \tau) = \begin{cases} 1 - \frac{2}{1+e^{\frac{-2x}{\tau}}} & x \leq 0 \\ \frac{2}{1+e^{\frac{-2x}{\tau}}} - 1 & x > 0 \end{cases}$
TV_{v2}	$T(x, \tau) = \begin{cases} 1 - \frac{2}{1+e^{\frac{-x}{\tau}}} & x \leq 0 \\ \frac{2}{1+e^{\frac{-x}{\tau}}} - 1 & x > 0 \end{cases}$
TV_{v3}	$T(x, \tau) = \begin{cases} 1 - \frac{2}{1+e^{\frac{-x}{2\tau}}} & x \leq 0 \\ \frac{2}{1+e^{\frac{-x}{2\tau}}} - 1 & x > 0 \end{cases}$
TV_{v4}	$T(x, \tau) = \begin{cases} 1 - \frac{2}{1+e^{\frac{-x}{3\tau}}} & x \leq 0 \\ \frac{2}{1+e^{\frac{-x}{3\tau}}} - 1 & x > 0 \end{cases}$

During early phases (when $\tau = \tau_{\max}$), the probability of updating the position's element is higher, which provides higher exploration abilities of the initial population. On the other hand, the changing probability of the position's element becomes very low once $\tau = \tau_{\min}$, which gives a steadier exploitation trend in the latter stages of the run as in [60].

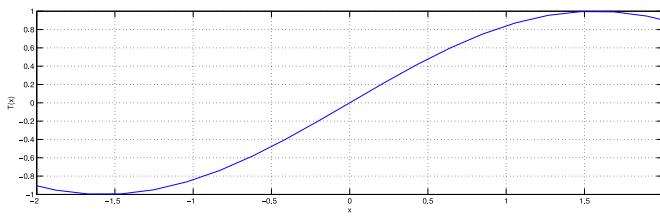
Although some literature indicated that the V-shaped TF has the same problem as the S-shaped TF, they did not study the adaptively updating possibility of this function over several iterations. This study modified the original TF by adding a time-varying TFs parameter as shown in Eq. (21). The proposed TF satisfies the design considerations mentioned in [49].

$$T(\Delta p(x), \tau) = \begin{cases} 1 - \frac{2}{1+e^{\frac{-2p(x)}{\tau}}} & p(x) \leq 0 \\ \frac{2}{1+e^{\frac{-2p(x)}{\tau}}} - 1 & p(x) > 0 \end{cases} \quad (21)$$

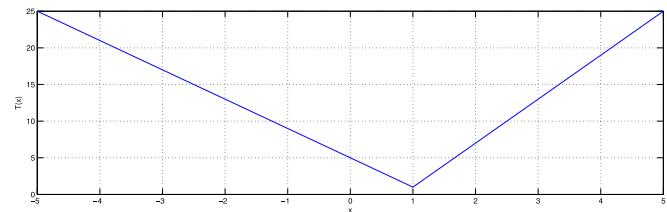
This paper uses eight new TFs modifying the coefficient of $p(x)$. Table 1 discusses the mathematical equations used in our experiment for the TFs shown in Fig. 2.

The developing solution. In this phase, updating solutions is the exploration step for making the solutions in global search, and new solutions are generated through Eq. (5). The solution from exploration to exploitation is generated by Eqs. (1) until (4). After that, genetic operators are applied to enhance the position and permit the trapping in local search using Eqs. (6) until (11), increasing the variety of the algorithm and also avoiding local solutions compared between a new solution and the previous one. For more diversity, genetic algorithm operators are used as this operator tends to integrate both mutant vectors y_{Mut} and z_{Mut} to generate a new child w_{Cross} as described in Eq. (10). Offspring fitness value: y_{Mut} , z_{Mut} and w_{Cross} based on the selection operator as shown in Eq. (11) are compared to get the best prey p_x .

The search space is boosted by exploring more regions to identify the best candidate solution using updating the step vector Eq. (12) that was applied to the TF equations Table 1. Several positions are updated by Eq. (16) in S-shape, and Eq. (18) for V-shape to produce a new version called bmRSO. TFs balance the position between local and global search in S-shape using Eq. (19) and in V-shape through Eq. (21). The stop condition is



(a) S-shape transfer function.



(b) V-shape transfer function.

Fig. 2. The transfer functions.

the maximum number of iterations that permits evaluating the bmRSO algorithm's performance. After finishing the bmRSO process, the best solution P_x is found. The Algorithm 2 declares how the best solution is proposed by calculating the fitness function using Eq. (22) for many new populations. The process is stopped if the determined condition is reached.

The genetic operators (mutation, crossover, and selection) control the effect of group members on each other and help in accelerating the convergence rate as provided from Eq. (6) to Eq. (11) because the genetic operators produce more population diversity to avoid local search, this leads to a relative speed for the rate of convergence over the optimization process. After that, transfer functions update the best position using Eqs. (16), (18) and balance local and global search from Eqs. (19) to (21) for S-shape and V-shape functions.

4.3. Computational complexity

The time complexity of RSO algorithm is $O(\text{Max}_{\text{Iteration}} \times n \times d \times N)$ according to [21]. Where n is the number of iterations, d is the dimension, N is the number of iterative steps until the best result is found, and $\text{Max}_{\text{Iteration}}$ is the maximum number of iterations. Adding genetic operators and TFs does not influence the complexity time according to this study [22], as each solution's TFS is calculated in all iterations, so the complexity of bmRSO is the same as RSO. The SVM classifier requires $O(k^3)$ complexity where k is the amount of data [61]. SVM classifier is hybridized with bmRSO for classification purposes. Therefore, the overall time complexity of the bmRSO-SVM approach is $(\text{Max}_{\text{Iteration}} \times n \times d \times N) \times k^3$.

4.4. Mathematical model for bmRSO

The Algorithm 2 declares how the best solution is proposed by calculating the fitness function. If a special condition is achieved so this process is stopped. The bmRSO-SVM algorithm process steps are detailed in Algorithm 2.

5. Experiments in optimization benchmark problems

The CEC'20 test suite is employed to check the proposed bmRSO performance, so several metrics are used as quantitative metrics (the mean and STD) for the best solutions (minimum value). Qualitative metrics contain search history, average fitness history, and the optimization history obtained by the eight variants of bmRSO and the original RSO algorithm.

5.1. CEC'20 test suite description

Ten test functions are included in the CEC'20 test suite: hybrid, multimodal, unimodal, and composition functions [62]. CEC'20 test suite is illustrated in Table 2, where Fi^* denotes the optimal global value.

Fig. 3 presents a 2D visualization of the CEC'20 test suite to help understand the nature of each problem.

Algorithm 2 The proposed bmRSO.

```

1: Initialization: Initialize the rat population  $p(X)$  using Eq. (23),
    $N, \text{MaxIter}$ .
2: Initialize the parameters of  $A, C, R$ 
3: Creating a new hybrid search agent from the current RSO as
   shown in Eq. (5)
4: Calculate the fitness of all Individuals  $P_r \leftarrow$  the best search
   agent.
5: while (Stopping condition is not met) do
6:   For each search agent do
7:     Apply genetic operator using Eqs. (6) until Eq. (11).
8:     Check if there is any search agent which goes beyond the
       given search space and then adjust it
9:     Calculate step vector using Eq. (12)
10:    Set  $\tau_{\max}$  and  $\tau_{\min}$ 
11:    Determine  $\tau$  by using Eq. (20)
12:    Calculate  $T(\Delta p(X))$  using Table 1.
13:    Two Eqs. (16) and (18) are used to update several position
       based on TFs.
14:    The proposed TF satisfies the design of bmRSO to balance
       the global and local search using Eqs. (19) and (21).
15:    Calculate the fitness of all Individuals.
16:    Update  $P_r$  using feature selection is done by Eq. (24) and
       then the Fitness function is applied by Eq. (22). If there is a
       better solution than the previous optimal solution.
17:   End For.
18:    $t = t + 1$ 
19: end while
20: Return the best Criteria

```

Table 2

Benchmark functions for CEC'20 are detailed.

No.	Function description	Fi^*
Unimodal function		
F1	Shifted and Rotated Bent Cigar Function	100
Multimodal shifted and rotated functions		
F2	Shifted and Rotated Schwefel's Function	1100
F3	Shifted and Rotated Lunacek bi-Rastrigin Function	700
F4	Expanded Rosenbrock's plus Griewangk's Function	1900
Hybrid functions		
F5	Hybrid Function 1 ($N = 3$)	1700
F6	Hybrid Function 2 ($N = 4$)	1600
F7	Hybrid Function 3 ($N = 5$)	2100
Composition functions		
F8	Composition Function 1 ($N = 3$)	2200
F9	Composition Function 2 ($N = 4$)	2400
F10	Composition Function 3 ($N = 5$)	2500

5.2. Statistical results on the CEC'20 test suite

Table 3 presents the mean, STD, and Friedman mean rank of the best value from the proposed algorithm and all other

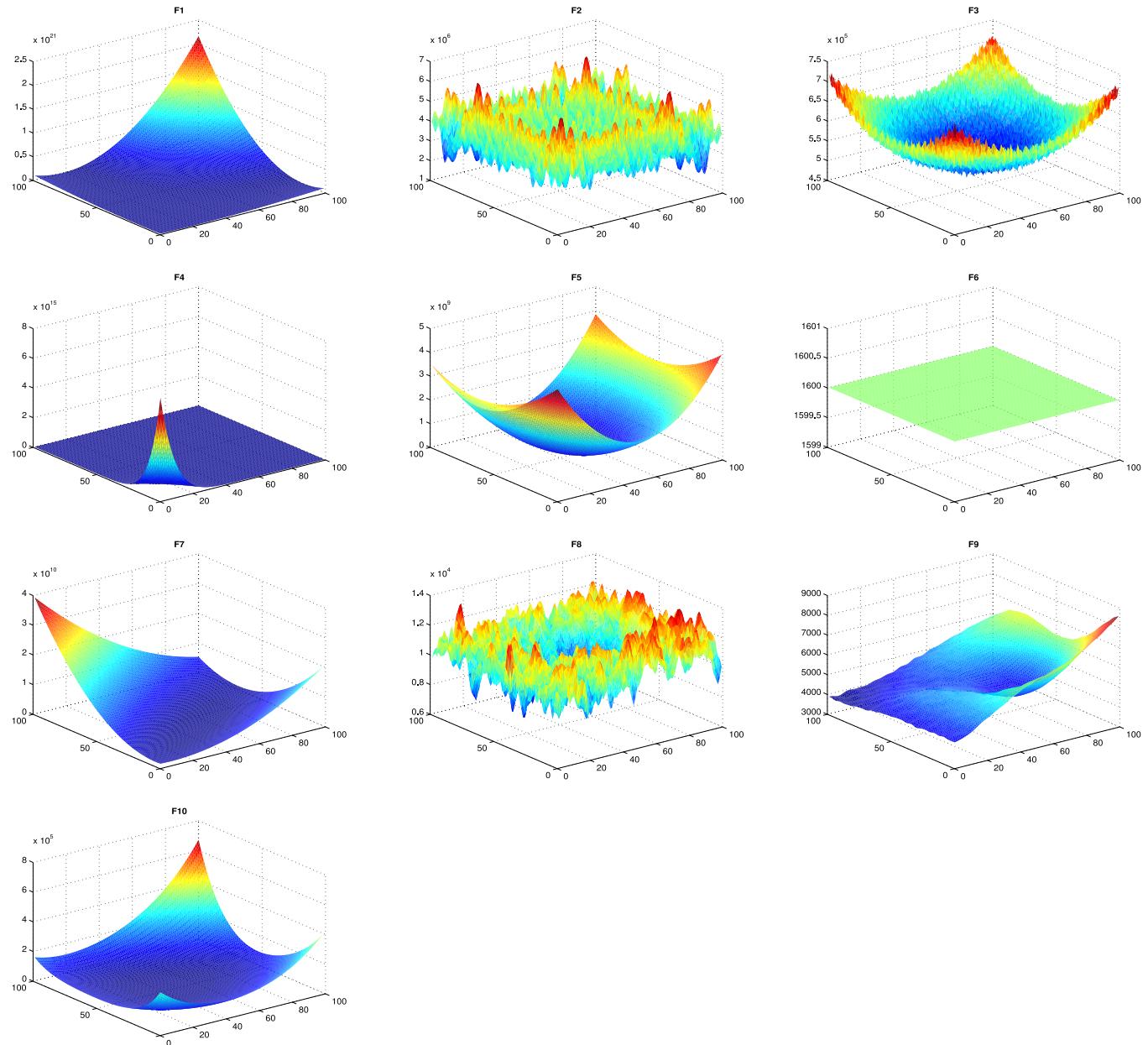


Fig. 3. The 2D visualization of the CEC'20 test suite.

compared methods for each benchmark function with 10 dimensions, the best results (minimum values) are also included. The values of mean, STD, and the Friedman test proved the effectiveness of the suggested algorithms in solving ten of the functions compared to the original algorithm. Moreover, bmRSO1 gained the first place for all the statistical results in mean, STD, and Friedman test terms for most testing functions, but RSO is promising last.

5.3. Convergence behavior analysis

The bmRSO methods and original algorithm performance can be declared with convergence curves and counterparts for the CEC'20 test suite shown in Fig. 4. All of the functions in the proposed algorithm reach a stable point, indicating that fast convergence is the optimal solution. The proposed bmRSO4 is the best for F1, F2, F3, and F4, but bmRSO5 for F5, and also bmRSO2, bmRSO1 for F6, F7 respectively, and bmRSO5 for F8, F9, and

F10. The above experimental results and convergence curves further prove that bmRSO, with the combined effect of genetic operators and transfer functions, significantly improves search capability and accuracy and has advantages over other compared algorithms.

5.4. Box plot behavior analysis

The box plot analysis can indicate the distribution of data, and the class of functions is related to too many local minima. To best understand the distribution of the results, the box plot of results and function for each algorithm is shown in Fig. 5. The minimum and maximum are the lowest and largest data points given by the algorithm, which are the edges of the whiskers. The lower and upper quartiles are delimited by the ends of the rectangles. A narrow box plot indicates the best distribution of the data. Fig. 5 shows the results of a ten-function box plot for $Dim = 10$. The box plots of the proposed bmRSO1 algorithm in F1, F2, F3, F4, F6, F7,

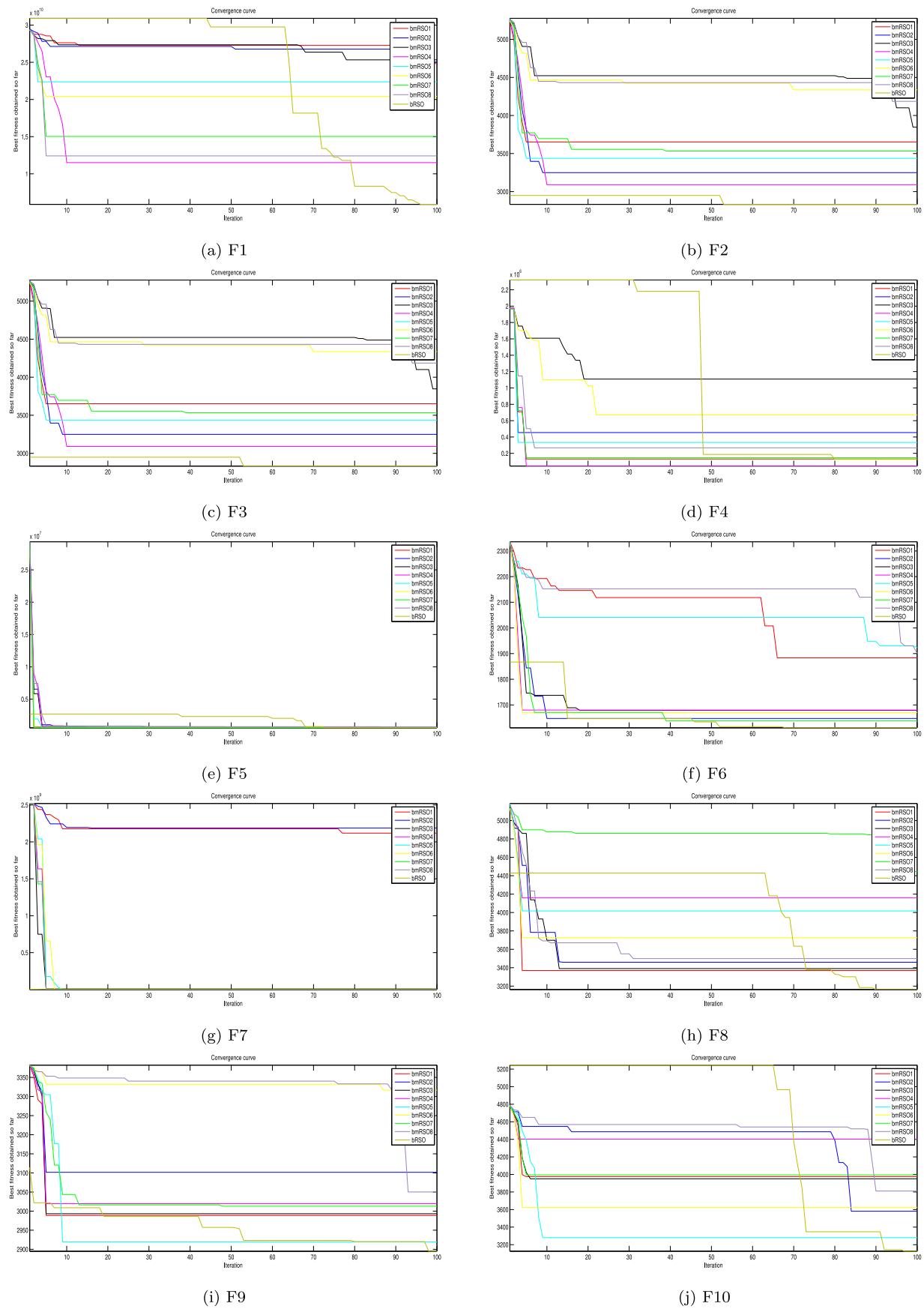


Fig. 4. The convergence curves of the bmRSO variants on the CEC'20 test suite with $Dim = 10$ based on a fitness function.

Table 3The mean and STD of fitness values for 30 runs obtained by the competitor algorithms on the CEC'20 test suite with $Dim = 10$.

Algorithms		F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	Friedman	Rank
RSO	Mean	1.80E+01	1.85E+01	1.89E+01	2.20E+01	3.30E+01	1.80E+05	4.80E+05	4.01E+06	1.65E+10	4.06E+03	7.8	9
	STD	1.84E+01	1.88E+01	2.40E+01	2.30E+01	3.01E+01	1.86E+05	3.86E+05	4.90E+06	1.13E+10	4.40E+01		
bmRSO8	Mean	1.77E+01	1.87E+01	1.83E+01	2.13E+01	3.30E+01	1.79E+05	2.89E+05	3.89E+06	6.36E+08	3.96E+03	2.2	2
	STD	1.80E+01	1.80E+01	2.20E+01	2.20E+01	2.70E+01	1.74E+05	2.94E+05	4.94E+06	1.01E+09	3.33E+01		
bmRSO7	Mean	1.76E+01	1.76E+01	1.80E+01	2.10E+01	3.10E+01	1.68E+05	2.78E+05	3.98E+06	3.65E+08	3.98E+03	4.4	5
	STD	1.78E+01	1.80E+01	2.17E+01	2.17E+01	2.67E+01	1.74E+05	2.84E+05	4.04E+06	7.82E+08	3.79E+01		
bmRSO6	Mean	1.74E+01	1.74E+01	1.78E+01	2.08E+01	3.28E+01	1.70E+05	2.77E+05	3.97E+06	4.54E+08	3.99E+03	5.8	7
	STD	1.69E+01	1.84E+01	2.20E+01	2.35E+01	2.65E+01	1.42E+05	2.92E+05	4.99E+06	9.56E+08	3.61E+01		
bmRSO5	Mean	1.75E+01	1.74E+01	1.75E+01	2.05E+01	3.30E+01	1.73E+05	2.70E+05	3.99E+06	6.65E+08	3.62E+03	7.3	8
	STD	1.72E+01	1.82E+01	2.12E+01	2.32E+01	2.65E+01	1.50E+05	2.59E+05	4.25E+06	1.06E+09	3.44E+01		
bmRSO4	Mean	1.72E+01	1.78E+01	1.70E+01	1.99E+01	3.24E+01	1.65E+05	2.95E+05	3.99E+06	7.52E+06	3.95E+03	5.3	6
	STD	1.68E+01	1.85E+01	2.10E+01	2.30E+01	2.60E+01	1.37E+05	2.47E+05	4.10E+06	1.22E+09	3.38E+01		
bmRSO3	Mean	1.67E+01	1.76E+01	1.68E+01	1.98E+01	3.20E+01	1.69E+05	2.89E+05	3.90E+06	4.43E+08	3.40E+03	3.1	3
	STD	1.71E+01	1.81E+01	2.09E+01	2.25E+01	2.65E+01	1.37E+05	2.97E+05	4.01E+06	9.27E+08	3.50E+01		
bmRSO2	Mean	1.70E+01	1.72E+01	1.67E+01	1.97E+01	3.17E+01	1.67E+05	2.87E+05	3.89E+06	2.24E+08	3.14E+03	3.3	4
	STD	1.73E+01	1.83E+01	2.08E+01	2.22E+01	2.62E+01	1.32E+05	2.82E+05	3.88E+06	7.02E+08	3.30E+01		
bmRSO1	Mean	1.60E+01	1.70E+01	1.65E+01	1.90E+01	3.10E+01	1.63E+05	6.34E+04	7.74E+05	2.20E+08	3.13E+03	1.5	1
	STD	1.65E+01	1.80E+01	2.05E+01	2.15E+01	2.60E+01	1.31E+05	7.54E+04	4.94E+05	6.90E+08	3.29E+01		

F8, F9, and F10 are optimal, but bmRSO8 is the best for F5, which is very narrow compared to other algorithm distributions and has the lowest values. Indeed, the proposed bmRSO methods perform better than the original algorithms for most test functions.

5.5. Qualitative metrics discussion

Monitoring the behavior of the candidate solutions can give more knowledge about algorithm convergence and the search process. The bmRSO1 version qualitative analysis is indicated in Fig. 6. The agent's behaviors are shown in Fig. 6, which illustrates the functions in two dimensions (2D), convergence curves, search history, and average fitness history.

The following points qualitatively analyze the performance of the proposed algorithm:

- *In terms of the domain's topology - functions in 2D views:* The function in 2D space is indicated in the first column of Fig. 6. The functions have different typologies that help the algorithm determine the best performance type or shape of the function.
- *For the search history:* From the first to the last iteration, the agents' search history is displayed in the second column of Fig. 6. Counter lines indicate the search space, with a gradient from blue to red indicating a higher fitness value. According to search history, the recommended bmRSO1 can find the places with the lowest fitness values for specific functions.
- *In terms of average fitness history:* The average fitness history, i.e., the fitness value averaged as a function of the iteration number, is displayed in the third column of Fig. 6. The general behavior of the agents as well as their contribution to the optimization process, are shown by this average. As the historical curves decrease, the population improves. This ongoing improvement displays cooperative-seeking behavior and supports particle law updates' usefulness.

6. Application of bmRSO for feature selection

Feature selection is a binary optimization problem, so the solutions are binary values. A new approach called bmRSO-SVM solves this problem. The solution is a vector of (0 or 1), where 0 is that the feature is not selected and 1 is that the feature is selected. The solution vector length equals the feature number

in the original dataset to obtain optimal accuracy. FS is the pre-step of the classification stage to propose the best feature from the dataset. Furthermore, most datasets are high-dimensional spaces. SVM is more effective in high-dimensional spaces than other classifiers, so the SVM classifier's performance is used to validate the efficacy of dimensional reduction. For the chemical dataset used in our paper, SVM is applied to propose the best chemical compound for a suitable drug. Our major objective is to validate the influence of eight TFs on bmRSO. Furthermore, eight TFs methods are compared with the original RSO. Also, some state of art binary algorithms such as Harris Hawks Optimization (HHO) [14], Grey Wolf Optimization (GWO) [25], Farmland fertility Algorithm (FFA) [26], Artificial Gorilla Troops Optimizer (GTO) [27], African Vultures Optimization Algorithm (AVOA) [28], Runge Kutta Optimizer (RUN) [29], and Slime Mould Algorithm (SMA) [30] are compared with the best bmRSO version.

The advantages of the proposed methodology can be summarized in the following points:

1. More RSO binary versions are produced based on genetic operators and TFs to avoid local search and balance global and local search.
2. The proposed bmRSO-SVM versions are tested on several well-known FS datasets, and the best version results are declared.
3. The optimal bmRSO-SVM version is compared with other state-of-the-art binary algorithms tested on several well-known FS datasets.

6.1. Architecture of the bmRSO-SVM approach

In this paper, a wrapper feature selection method using an SVM classifier [54] is applied to the proposed bmRSO to choose subsets of an attribute, as declared in Fig. 7. During the iterative process, the fitness function has two main targets: to evaluate the accuracy, select the number of features, and confirm their performance, as shown in Eq. (22). The bmRSO-SVM fit is indicated as:

$$fit = \alpha + \beta \frac{|R|}{|C|} - G. \quad (22)$$

where $\beta = \alpha$ and $fit > T$. R is the classification error rate, C is the total feature number in the dataset, and α and β are two parameters that influence classification quality and subset length, respectively. The α parameter is between [0, 1]. G is a group for

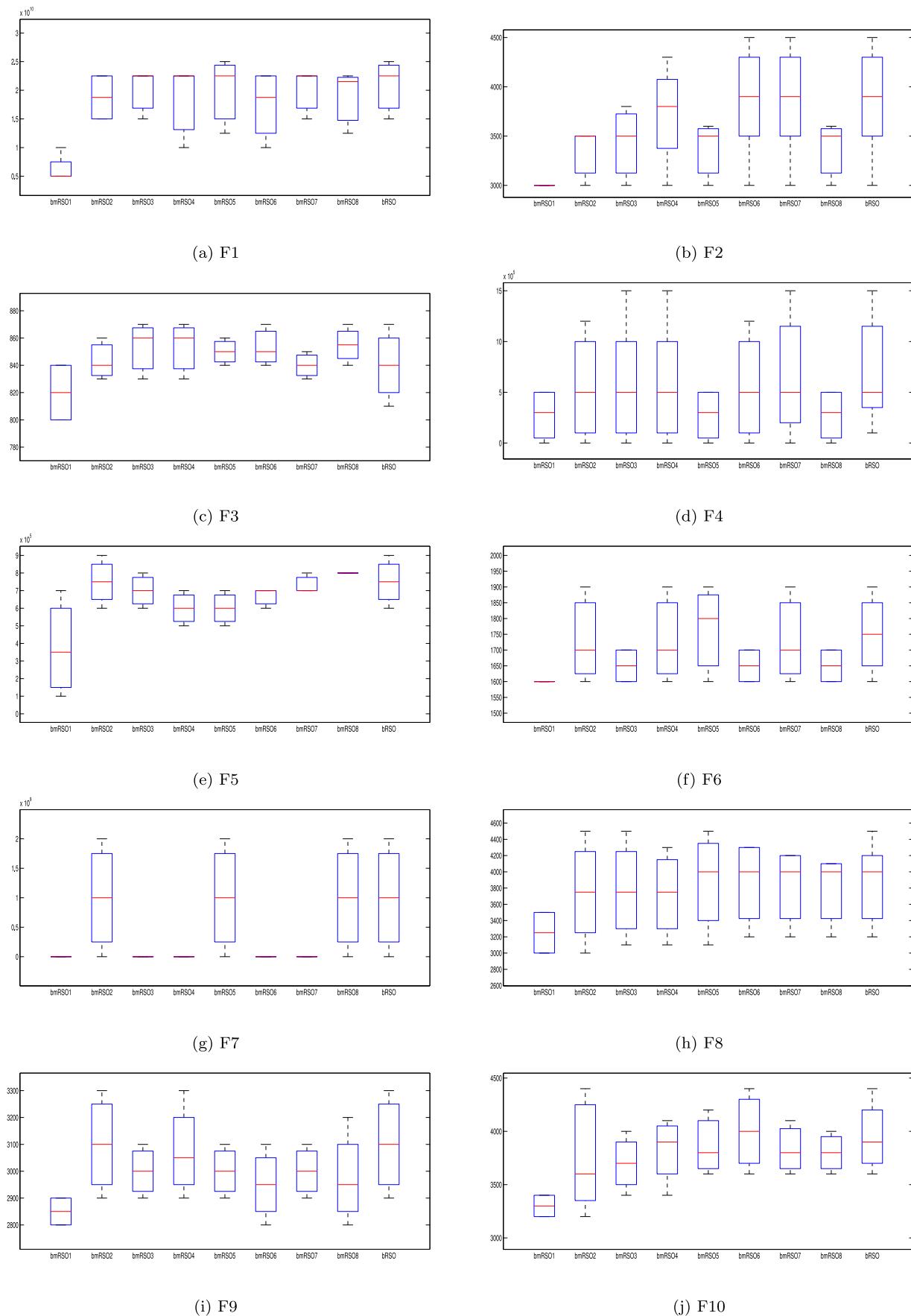


Fig. 5. The box plot of the proposed bmRSO methods compared with the original algorithm on the CEC'20 test suite with $Dim = 10$.

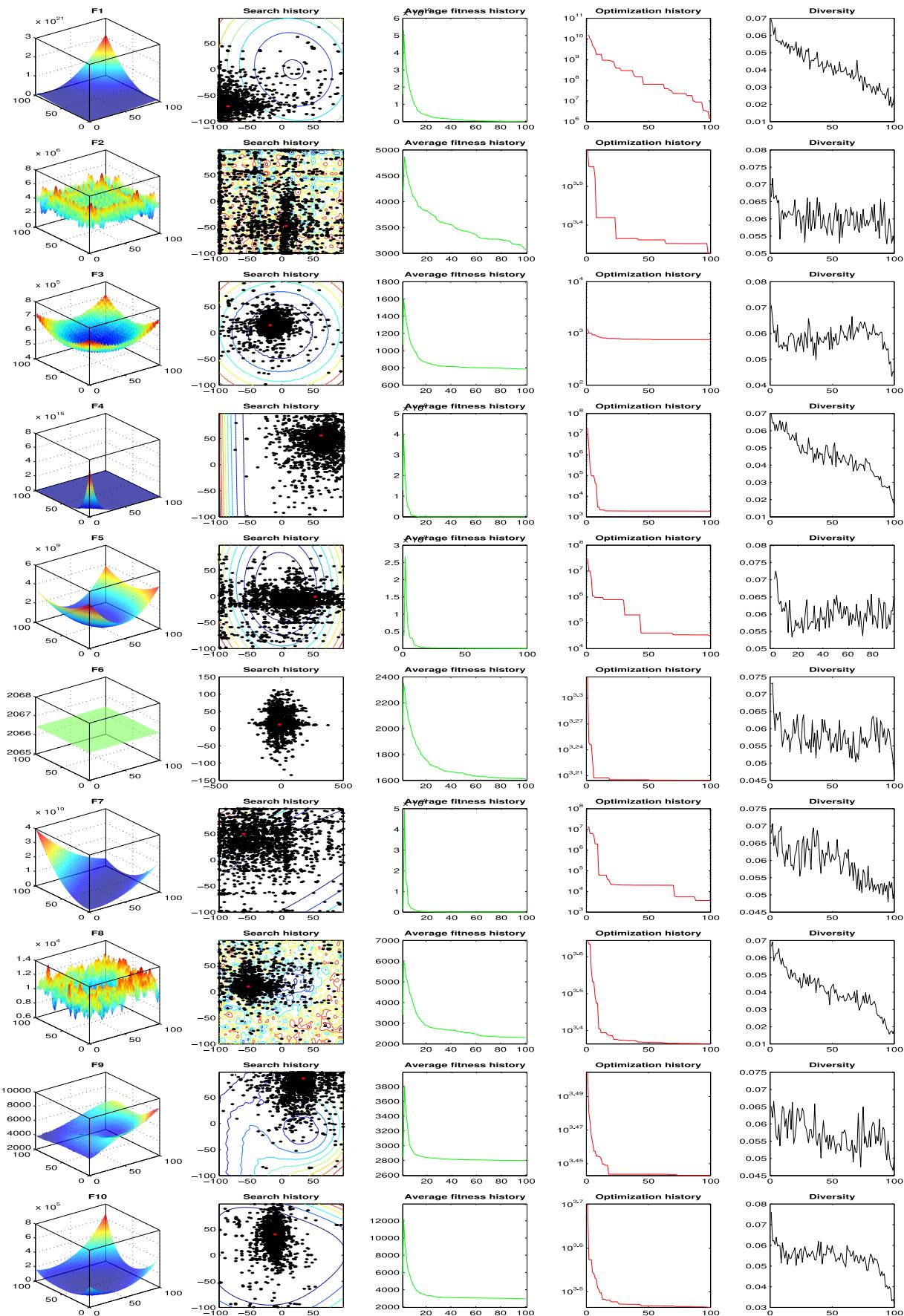


Fig. 6. The qualitative metrics on the CEC'20 test suite are: 2D views of the functions, search history, average fitness history, and optimization history.

Table 4
Datasets description.

Dataset	Features number	Instance number
Monoamine Oxidase	1665	68
QSAR Biodegradation	41	1055
Drug Review	6	215063
Immunotherapy	8	90
QSAR androgen receptor	1024	1687
Anticancer peptides dataset(Breast cancer)	3	950
QSAR oral toxicity	1024	8992
Chemical Composition of Ceramic Samples	19	88

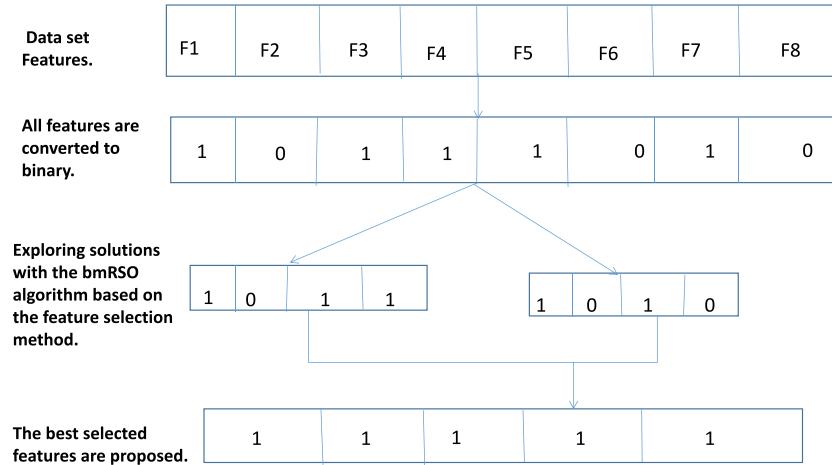


Fig. 7. TFs feature selection.

the classifier, and T is the condition used to compare with the fitness function. The fit is used to check the target of optimization.

The bmRSO-SVM versions start the process of optimization by initializing the population of search agents randomly using a uniform random distribution as follows:

$$p(X_i^{initial}) = LB + \text{rand}_i(UB - LB) \quad i = 1, 2, \dots, n \quad (23)$$

$p(X_i)^{initial}$ is a randomly initialized i th solution vector, UB (upper bound), and LB (lower bound) are respectively, n refers to the search agent, and $\text{rand}_i \in [0, 1]$ is a random value. Before the fitness evaluation process, an intermediate conversion step is required to choose a subset of attributes, so each solution $p(X_i)$ goes through a binary transformation ($p(X_i)_{bin}^i$) using Eq. (24):

$$p(X_i)_{bin}^i = \begin{cases} 1 & \text{if } p(X_i)^i > 0.5 \\ 0 & \text{otherwise.} \end{cases} \quad (24)$$

An example for understanding the selection of features $p(X_i)^i = [0.4, 0.8, 0.33, 0.15, 0.8, 0.2, 0.1, 0.9]$. Eq. (24) is applied to make a binary vector $p(X_i)_{bin}^i = [0, 0, 1, 0, 0, 1, 0, 1, 0, 1]$ where 1 is selected feature and 0 is not selected and must be eliminated, so the binary vector after selected feature $p(X_i)_{bin}^i = [1, 1, 1, 1]$. The fitness function (fit) is calculated by Eq. (22). The SVM from Algorithm 1 is used as a classifier after FS as a preprocessing for the classification. The classification strategy is split into 90% as a training part and the other part as testing. The lower fitness value through all individuals is used to the best prey $p(X_i)$.

Moreover, the feature selection detailed steps are exhibited in Fig. 7. This figure explains how dataset features can be converted to binary steps and then the role of the proposed bmRSO in selected features using the feature selection method. Finally, the best features are selected.

6.2. Applying bmRSO for FS

This part indicates how the proposed methods can give the best results in chemical data. Firstly, data preprocessing is necessary to extract several descriptors (features), and after that, statistical experimental results are displayed.

6.2.1. Data description

Different datasets are applied to our proposed FS model to check its performance in this experiment. All datasets are available from the machine learning repository (UCI), but MOA is taken from cheminformatic.org (see Table 4).

6.2.2. Pre-possessing data

As shown in Fig. 8, preprocessing steps are applied to some chemical data. The main stages are discussed: (1) First, the information about proteins is transformed into a chemical structure; (2) descriptors are calculated; and (3) the chemical structure is converted into a mathematical form. The following are the phases:

1. Protein information is transformed into Simplified Molecular-Input Line-Entry System Style (SMILES) using open Babel software [63]. Firstly, protein information is represented in a chemical format called monoamine oxidase (MAO), which is needed to transform into SMILES using Babel software. Features are attributes with values used to create instances.
2. Several chemical features are performed to implement several 2D and 3D datasets in the QSAR model, and descriptors are calculated using E-Dragon software. The descriptors are divided into rotary and physicochemical links (weight, distance between atoms, and atom type).

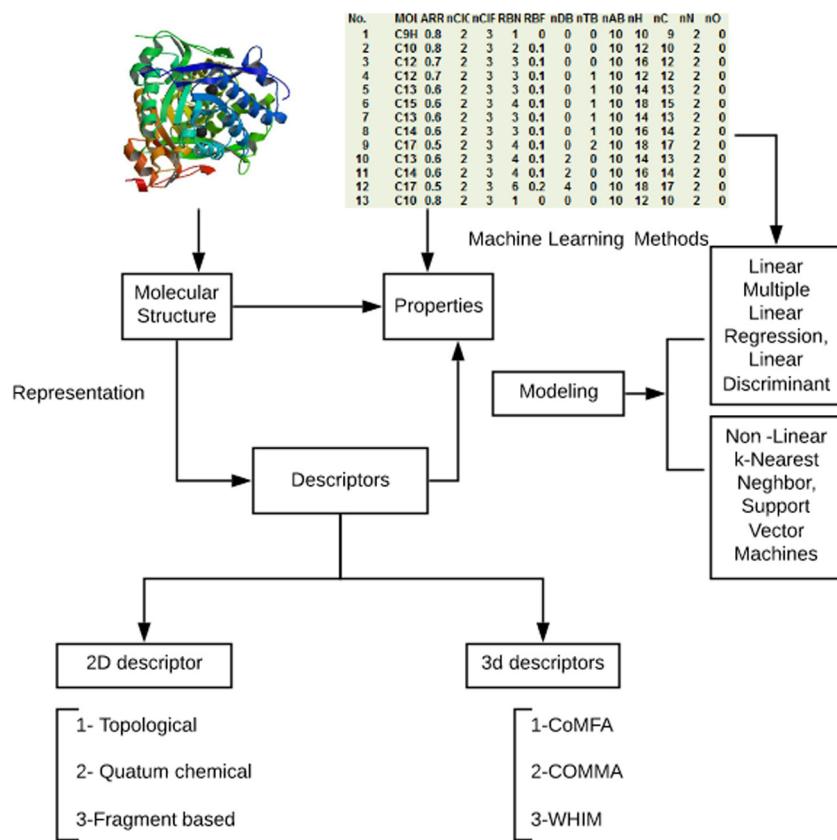


Fig. 8. Mapping from molecules to features.

3. The QSAR model is the mathematical relationship between chemical design and biological activity. The features can also identify the instances, as shown in Fig. 9. QSAR is used to indicate the major properties of chemical compounds [64].

6.2.3. FS statistical results

This subsection presents the statistical criteria for the best value from the proposed bmRSO-SVM methods compared with the original RSO for each dataset. Results in several statistical terms, as shown in Tables 5 and 6, revealed the preference of the proposed methods in solving the FS problems compared to the original RSO. The maximum value represents the best, the minimum is the worst, STD is the smallest value, CPU time indicates the execution time for each algorithm, and the average number of features that retrieve the selected feature count is depicted in Tables 5 and 6.

6.2.4. Convergence behavior analysis

The proposed binary versions of the bmRSO-SVM convergence curves compared to the original RSO are presented in Fig. 10 over eight chemical datasets and have a fixed point for all datasets. Furthermore, the proposed bmRSO1-SVM version has achieved optimal solutions, which are the best for most datasets. The bmRSO1-SVM algorithm is the best for solving FS problems and achieving high accuracy compared to other methods, but the original RSO is the last, as shown in Fig. 10. The convergence curves show that bmRSO1 significantly improves search capability accuracy and outperforms other compared algorithms when combined with genetic operators and TFS.

6.2.5. Box plot behavior analysis

The box plot is used to evaluate the performance of several datasets as a non-parametric measure. However, in descriptive statistics, a box plot represents the groups of numerical data as a graph through their quartiles. The maximum or minimum are the largest or lowest data points the algorithm reaches. In these experiments, the box plots for bmRSO-SVM methods over the eight datasets are presented in Fig. 11. The proposed bmRSO1-SVM box plot is very narrow compared to another method for the largest dataset, thus having the most significant values.

6.3. Applying bmRSO for toxicity data FDA

Toxicity data from the Food and Drug Administration (FDA) can be defined as a drug harmful to the complete organism as an organ like the liver. The toxicity risk influences the medications' drug biotransformation in the liver as shown in Table 7. The FDA dataset [7] contains 5909 FDA-approved medications with a variety of therapeutic effects ranging from hypertension management to cancer treatment and nutritional supplementation. Furthermore, the DataWarrior software is used to extract 31 molecular characteristics from medicines [65].

FDA dataset, experimental: The toxicity dataset benchmark consists of four datasets (tumorigenicity, mutagenicity, irritating, and reproductive). The experimental results indicated that the proposed bmRSO1-SVM is the best method compared to others. It shows its performance according to results as in Table 8, the convergence curve as in Fig. 12, and Fig. 13 for the box plot.

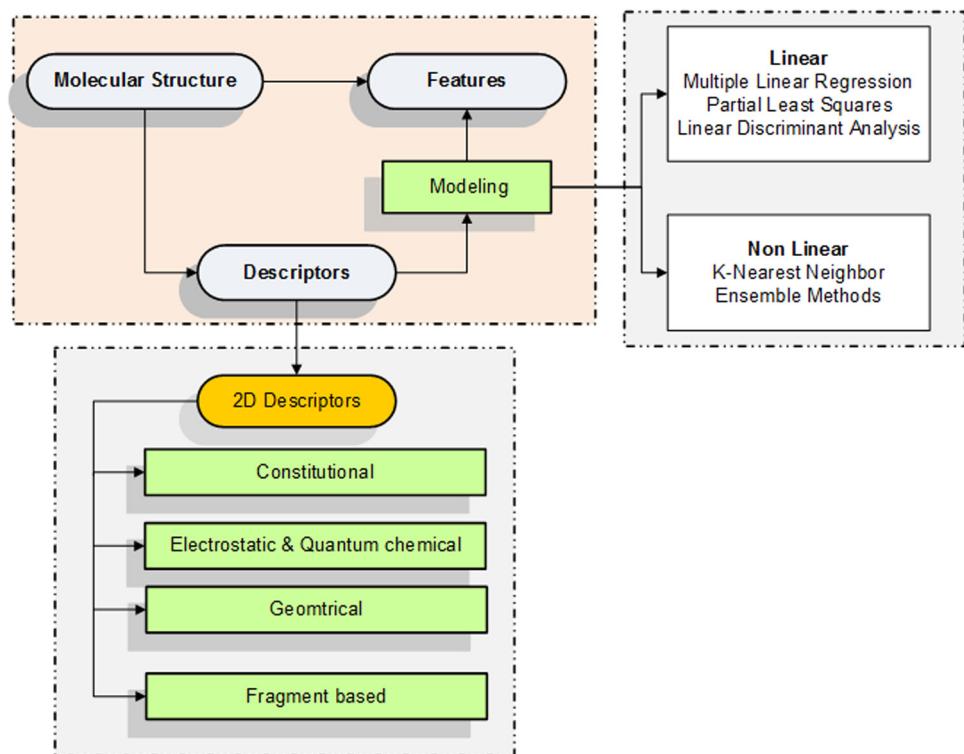


Fig. 9. Flowchart of the QSAR model.

Table 5

Statistical results of the several algorithms using the SVM with a stopping criterion based on maximum iteration.

Algorithm	Mean	STD	Best	Worst	CPU time	Average selection number
dataset No. 1: Monoamine Oxidase (MAO)						
bmRSO1	9.60E+01	1.03E-01	98.201	96.190	0.570	20.001
bmRSO2	9.21E+01	1.13E-01	96.010	93.070	0.570	21.005
bmRSO3	9.50E+01	1.27E-01	94.100	91.160	0.579	20.002
bmRSO4	9.06E+01	1.09E-01	93.555	90.100	0.574	20.005
bmRSO5	9.18E+01	1.86E-01	92.760	90.700	0.577	21.100
bmRSO6	9.19E+01	1.37E-01	92.030	91.105	0.578	21.200
bmRSO7	9.36E+01	1.40E-01	91.150	90.104	0.580	21.500
bmRSO8	9.06E+01	1.25E-01	90.100	89.780	0.580	22.002
RSO	8.30E+01	1.40E-01	90.050	89.140	0.565	31.500
dataset No. 2: Drug Review						
bmRSO1	9.40E+01	1.03E-01	95.100	92.100	0.100	2.001
bmRSO2	9.33E+01	1.15E-01	93.010	91.100	0.100	3.100
bmRSO3	9.30E+01	1.12E-01	91.001	90.101	0.107	2.500
bmRSO4	9.26E+01	1.17E-01	90.010	89.102	0.107	3.500
bmRSO5	9.22E+01	1.06E-01	94.190	91.401	0.101	3.200
bmRSO6	9.20E+01	1.07E-01	93.001	92.115	0.100	3.800
bmRSO7	8.41E+01	1.20E-01	88.150	87.130	0.100	3.200
bmRSO8	8.20E+01	1.30E-01	86.910	85.120	0.170	3.100
RSO	8.10E+01	3.30E-00	84.900	81.184	0.080	4
dataset No. 3: QSAR Oral						
bmRSO1	9.70E+01	1.08E-01	96.100	93.120	0.420	24.300
bmRSO2	9.65E+01	1.10E-01	92.211	90.110	0.420	24.700
bmRSO3	9.60E+01	1.12E-01	91.101	90.001	0.430	24.500
bmRSO4	9.55E+01	1.19E-01	92.011	90.100	0.430	24.800
bmRSO5	9.40E+01	1.20E-01	91.100	90.400	0.433	25.001
bmRSO6	9.35E+01	1.27E-01	90.100	89.115	0.430	30.100
bmRSO7	8.49E+01	1.40E-01	87.100	86.100	0.450	27.006
bmRSO8	8.40E+01	1.50E-01	87.710	86.020	0.455	27.700
RSO	8.10E+01	1.30E-01	83.100	80.104	0.400	30.001

(continued on next page)

6.4. Comparison with other metaheuristics

In this part, several binary metaheuristic versions such as Harris Hawks Optimization (HHO) [14], Grey Wolf Optimization

(GWO) [25], Farmland Fertility Algorithm (FFA) [26], Artificial Gorilla Troops Optimizer (GTO) [27], African Vultures Optimization Algorithm (AVOA) [28], Runge Kutta Optimizer (RUN) [29], and Slime Mould Algorithm (SMA) [30] are applied in comparison

Table 5 (continued).

Algorithm	Mean	STD	Best	Worst	CPU time	Average selection number
dataset No. 4: QSAR Androgen						
bmRSO1	9.80E+01	1.06E-01	96.120	95.120	0.320	24.500
bmRSO2	9.70E+01	1.10E-01	94.310	93.810	0.320	24.600
bmRSO3	9.65E+01	1.22E-01	95.120	93.221	0.330	24.500
bmRSO4	9.55E+01	1.25E-01	95.011	94.100	0.335	24.700
bmRSO5	9.40E+01	1.20E-01	93.100	91.400	0.337	25.100
bmRSO6	9.45E+01	1.30E-01	92.120	90.105	0.338	30.100
bmRSO7	9.49E+01	1.40E-01	91.120	90.020	0.340	27.005
bmRSO8	8.45E+01	1.55-E01	89.720	88.022	0.342	27.800
RSO	8.50E+01	1.35-E01	87.100	85.144	0.315	30.300

Table 6

Statistical results of the several algorithms using the SVM with stopping criterion based on maximum iteration.

Algorithm	Mean	STD	Best	Worst	CPU time	Average selection number
dataset No. 5: Immunotherapy						
bmRSO1	9.20E+01	1.02E-01	92.100	90.120	0.170	2.001
bmRSO2	9.20E+01	1.20E-01	91.810	90.010	0.170	4.187
bmRSO3	8.55E+01	1.22E-01	89.120	88.021	0.172	5.325
bmRSO4	8.65E+01	1.35E-01	87.311	86.124	0.174	5.100
bmRSO5	9.15E+01	1.20E-01	90.100	89.420	0.170	3.231
bmRSO6	8.55E+01	1.40E-01	88.020	87.215	0.170	6.300
bmRSO7	8.59E+01	1.45E-01	88.120	87.920	0.176	5.200
bmRSO8	8.45E+01	1.55-E01	85.120	83.021	0.170	4.100
RSO	8.60E+01	1.65-E01	82.101	81.141	0.160	6.002
dataset No. 6: Chemical Composition						
bmRSO1	9.25E+01	1.01E-01	93.100	91.100	0.150	6.923
bmRSO2	9.25E+01	1.01E-01	91.210	90.710	0.150	8.001
bmRSO3	8.51E+01	1.22E-01	88.220	87.021	0.155	9.120
bmRSO4	8.45E+01	1.35E-01	89.301	88.024	0.155	8.210
bmRSO5	9.15E+01	1.20E-01	90.120	89.550	0.150	9.126
bmRSO6	9.10E+01	1.20E-01	91.120	90.200	0.157	7.560
bmRSO7	8.59E+01	1.45E-01	88.120	87.920	0.158	10.832
bmRSO8	8.60E+01	1.50-E01	87.021	86.320	0.160	11.567
RSO	8.70E+01	1.65-E01	83.211	80.640	0.140	12.982
dataset No. 7: Anticancer Peptides cancer(Breast cancer)						
bmRSO1	9.35E+01	1.10E-01	94.120	92.120	0.100	0.500
bmRSO2	9.30E+01	1.11E-01	92.240	91.810	0.100	0.800
bmRSO3	8.27E+01	1.25E-01	88.521	87.120	0.120	1.001
bmRSO4	9.25E+01	1.30E-01	91.021	90.120	0.120	0.900
bmRSO5	9.20E+01	1.35E-01	91.220	90.610	0.125	0.821
bmRSO6	9.15E+01	1.40E-01	91.120	90.100	0.130	2.000
bmRSO7	8.47E+01	1.55E-01	86.100	86.120	0.135	2.000
bmRSO8	8.50E+01	1.60-E01	89.020	88.020	0.138	1.920
RSO	8.10E+01	1.65-E01	83.211	80.640	0.080	2.000
dataset No. 8: QSAR Biodegradation						
bmRSO1	9.35E+01	1.10E-01	92.120	92.000	0.240	19.781
bmRSO2	9.30E+01	1.15E-01	91.540	91.010	0.240	20.467
bmRSO3	8.40E+01	1.25E-01	88.120	87.520	0.245	21.385
bmRSO4	8.45E+01	1.20E-01	89.320	88.120	0.248	22.567
bmRSO5	9.20E+01	1.18E-01	90.521	90.110	0.249	23.590
bmRSO6	9.45E+01	1.27E-01	89.020	88.500	0.250	24.967
bmRSO7	8.42E+01	1.25E-01	87.120	85.220	0.250	29.001
bmRSO8	8.47E+01	1.22-E01	88.100	87.522	0.250	27.201
RSO	8.20E+01	1.30-E01	85.011	83.510	0.235	30.200

with the best bmRSO version to evaluate the performance of the proposed method over the same dataset previously used. The experimental results indicate that bmRSO1-SVM proposes the best solution, according to Tables 9 to 11. bmRSO1-SVM attains the best convergence speed, according to Figs. 14 and 15. According to Figs. 16 and 17, bmRSO1-SVM achieves the best box plot result compared with other methods.

7. Overall discussion

First, the proposed eight variants of bmRSO were compared with the original RSO algorithms tested on the CEC'20 test suite. Then, eight chemical and four toxicity datasets were then used to evaluate the proposed bmRSO-SVM versions' performance. For

the CEC'20 test suite, quantitative and qualitative metrics were used to evaluate the performance of bmRSO variants. The proposed bmRSO1 achieved the best value for statistical results for most testing functions compared with other versions. The mean values of bmRSO1 are 1.60E+01, 1.70E+01, 1.65E+01, 1.90E+01, 3.10E+01, 1.63E+05, 6.34E+04, 7.74E+05, 2.20E+08, and 3.13E+03. Although RSO values are 1.80E+01, 1.85E+01, 1.89E+01, 2.20E+01, 3.30E+01, 1.80E+05, 4.80E+05, 4.01E+06, 1.65E+10, 4.06E+03, and the bmRSO standard deviation values are 1.65E+1, 1.80E+1, 2.05E+1, 2.15E+1, 2.60E+1, 1.31E+05, 7.54E+04, 4.94E+05, 6.90E+08, and 3.29E+01, RSO values are 1.84E+01, 1.88E+01, 2.40E+01, 2.30E+01, 3.01E+01, 1.86E+05, 3.86E+05, 4.90E+06, 1.13E+10, 4.40E+01 for testing functions, respectively, the Friedman test for the bmRSO1 value is 1.5, which is the first rank

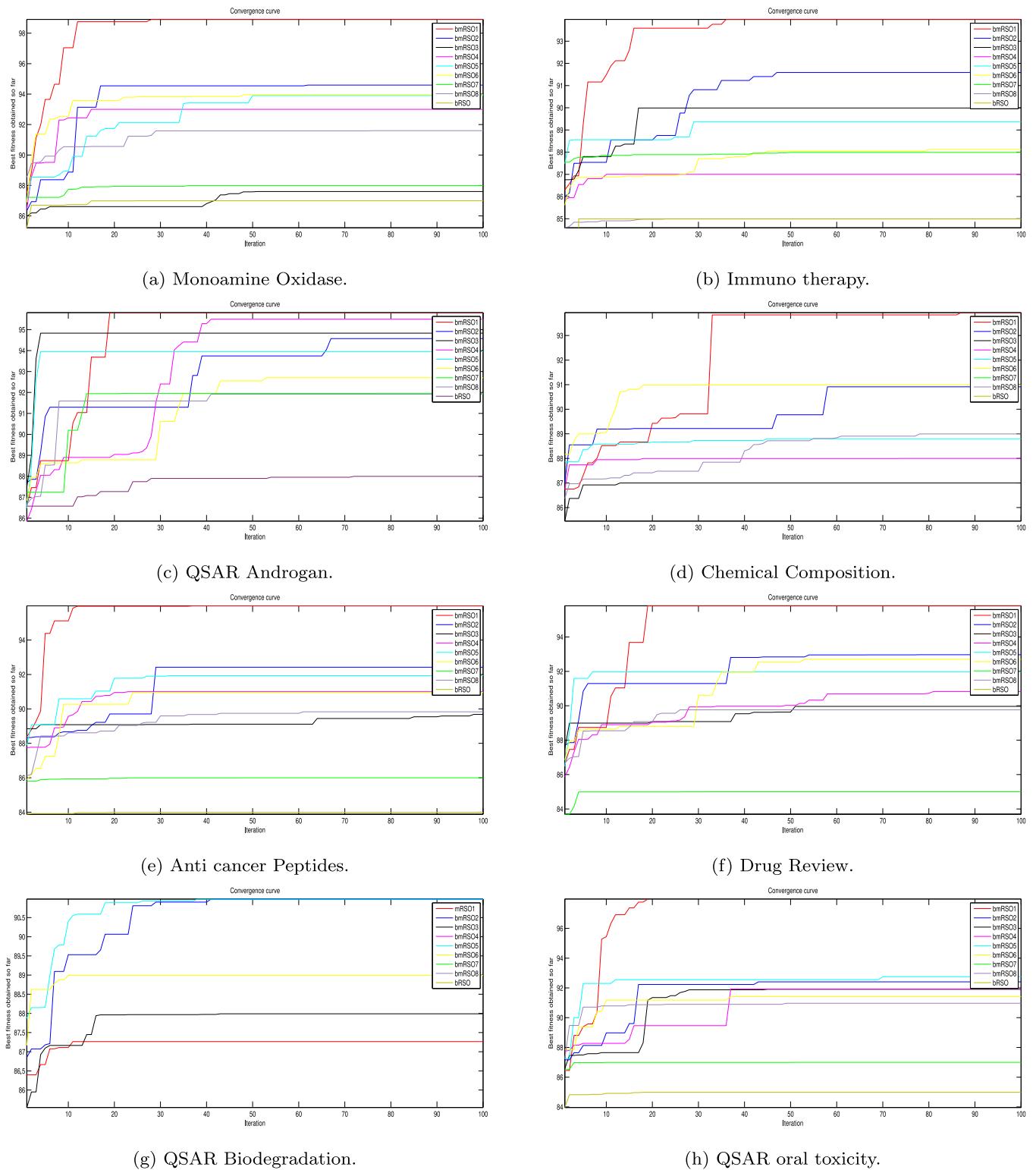
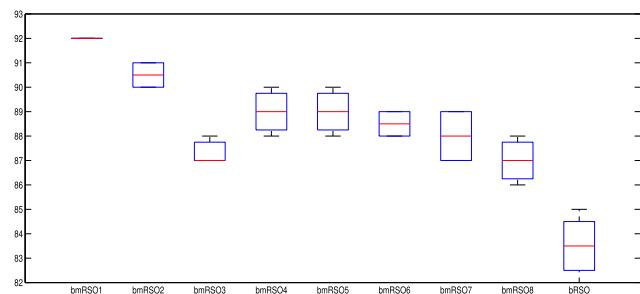


Fig. 10. The convergence curves obtained from the proposed bmRSO-SVM versions and the original bRSO-SVM algorithm over eight datasets.

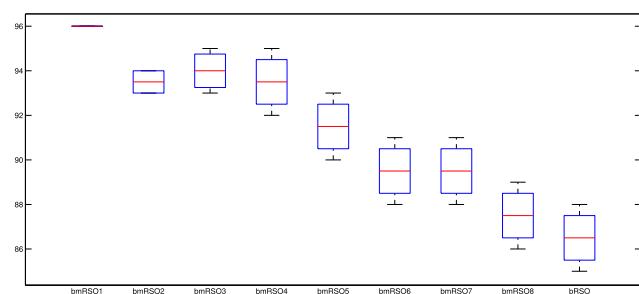
compared with other versions. At the same time, RSO is a 7.8 value and the last rank, so bmRSO1 is optimal compared with other versions, as shown in Table 3.

Also, the best results are the minimum convergence curve and box plot, as shown in Figs. 4 and 5. The bmRSO1 convergence curve is optimal for F7, but bmRSO1 is optimal for a box plot. Fig. 3 shows the parameter space used for 2D visualization of the

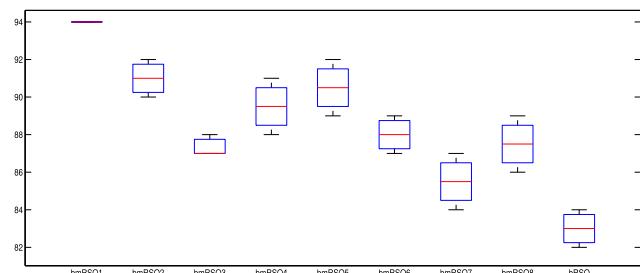
CEC'20 test suite to indicate the differences and nature of several problems. The qualitative metrics indicate the best summary of the algorithm performance for several problems to confirm the high performance of the proposed bmRSO1 algorithm, as shown in Fig. 6.



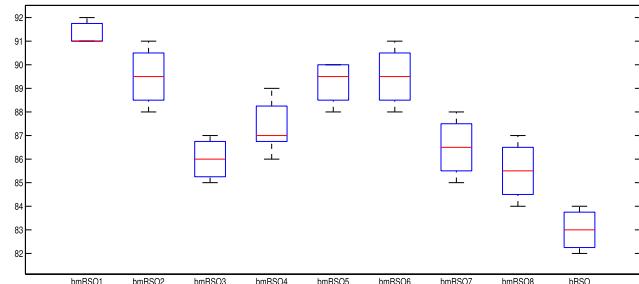
(a) QSAR Biodegradation



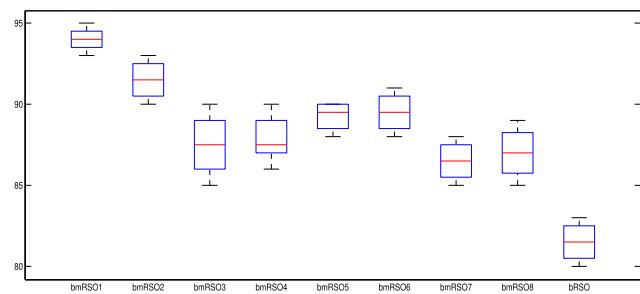
(b) QSAR Androgen



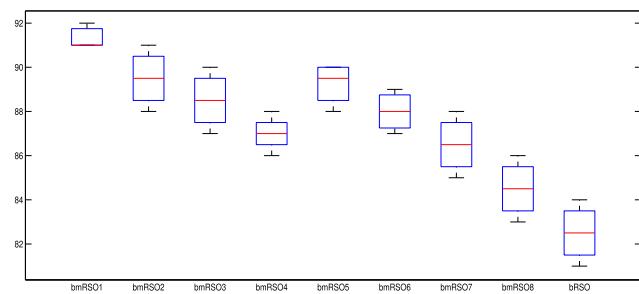
(c) Anticancer peptide



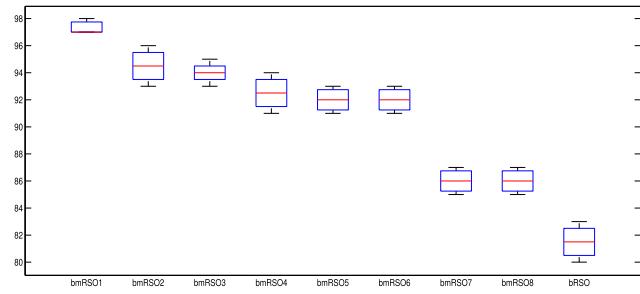
(d) Chemical Composition



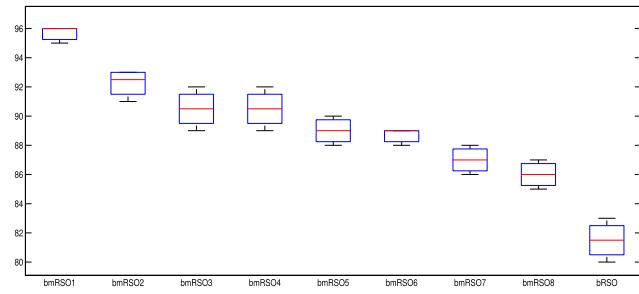
(e) Drug Review



(f) Immuno therapy



(g) Monoamine Oxidase



(h) QSAR Oral toxicity

Fig. 11. The box plot comparison obtained from the counterpart algorithms using SVM is based on the stopping criteria applied over eight datasets.

For FS, bmRSO-SVM versions are evaluated over several chemical datasets. The proposed bmRSO1-SVM version maximizes accuracy and reduces the number of features compared with other versions. In the MOA dataset, the bmRSO1-SVM version reaches 98.201% accuracy, and the average number of features reaches 20.001 compared with other versions of bmRSO-SVM and other binary algorithms. However, in the reprodictive dataset with

lower results, the best result is bmRSO1-SVM which has 82.30% accuracy and 12.400 average feature numbers compared with other versions. The bmRSO1-SVM has achieved the best statistical results as shown in Tables 5, 6, and 8 over all datasets. The convergence curves were used to evaluate the bmRSO-SVM approach over eight chemical datasets and four toxicity datasets, as illustrated in Figs. 10 and 12. The convergence represents the number

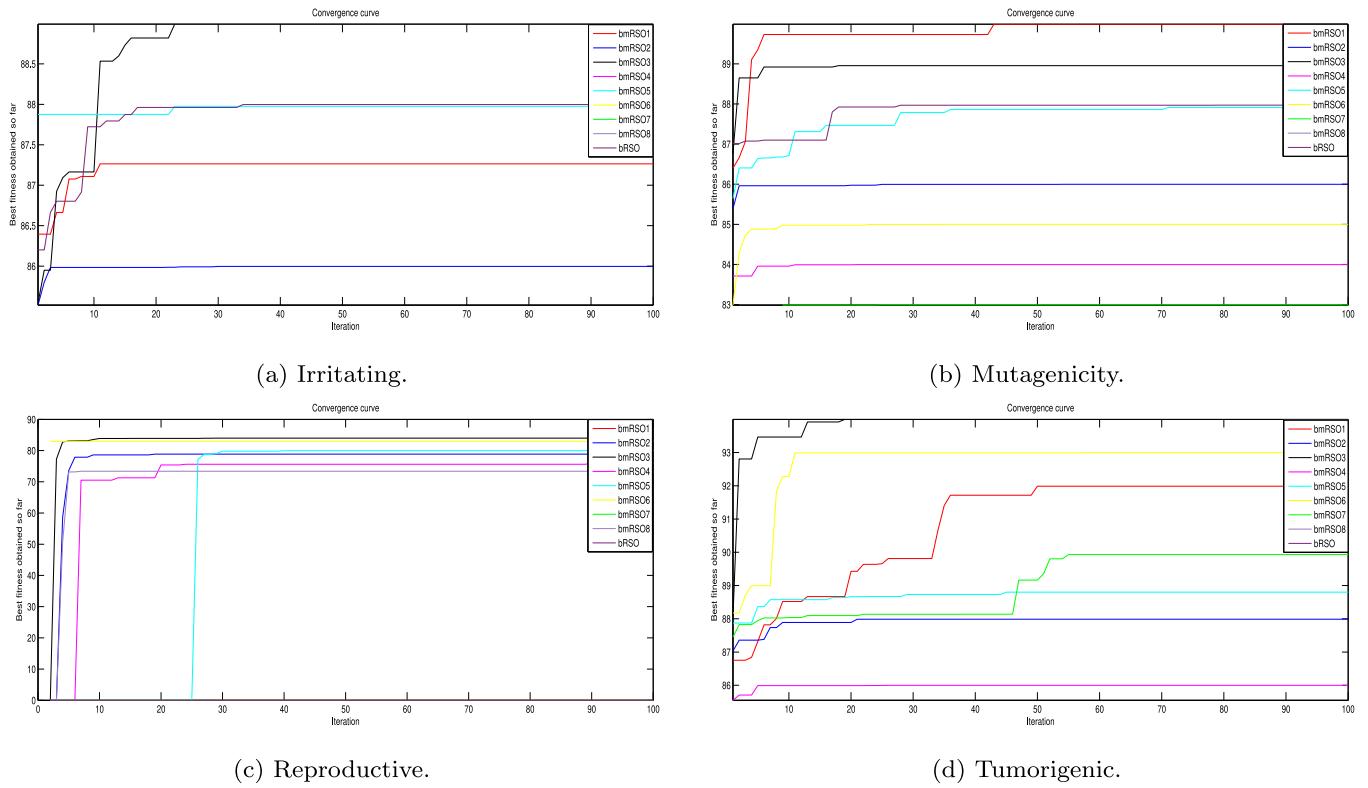


Fig. 12. FDA toxicity convergence curves obtained from the proposed bmRSO-SVM methods and the original algorithms over four toxicity datasets.

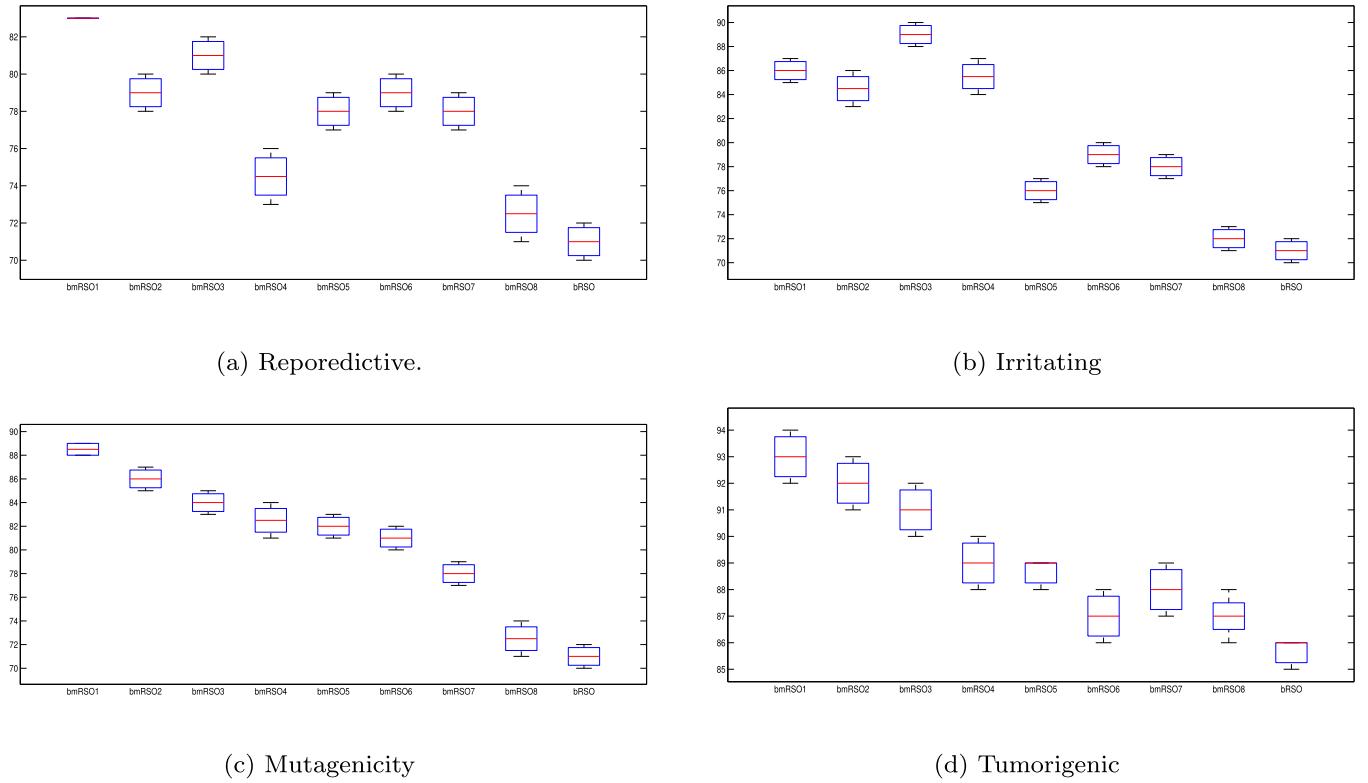


Fig. 13. FDA toxicity box plot comparison obtained from the counterparts using SVM on the different stop criteria applied to four toxicity datasets.

of iterations, and the fitness function is a relationship. It declares the best-performing algorithm that was compared among approaches. The box plot analysis indicates that the bmRSO1-SVM approach has performed better than other methods, as shown in

Figs. 11 and 13. Some binary MAs versions, such as Harris Hawks Optimization (HHO) [14], Grey Wolf Optimization (GWO) [25], Farmland Fertility Algorithm (FFA) [26], Artificial Gorilla Troops Optimizer (GTO) [27], African Vultures Optimization Algorithm

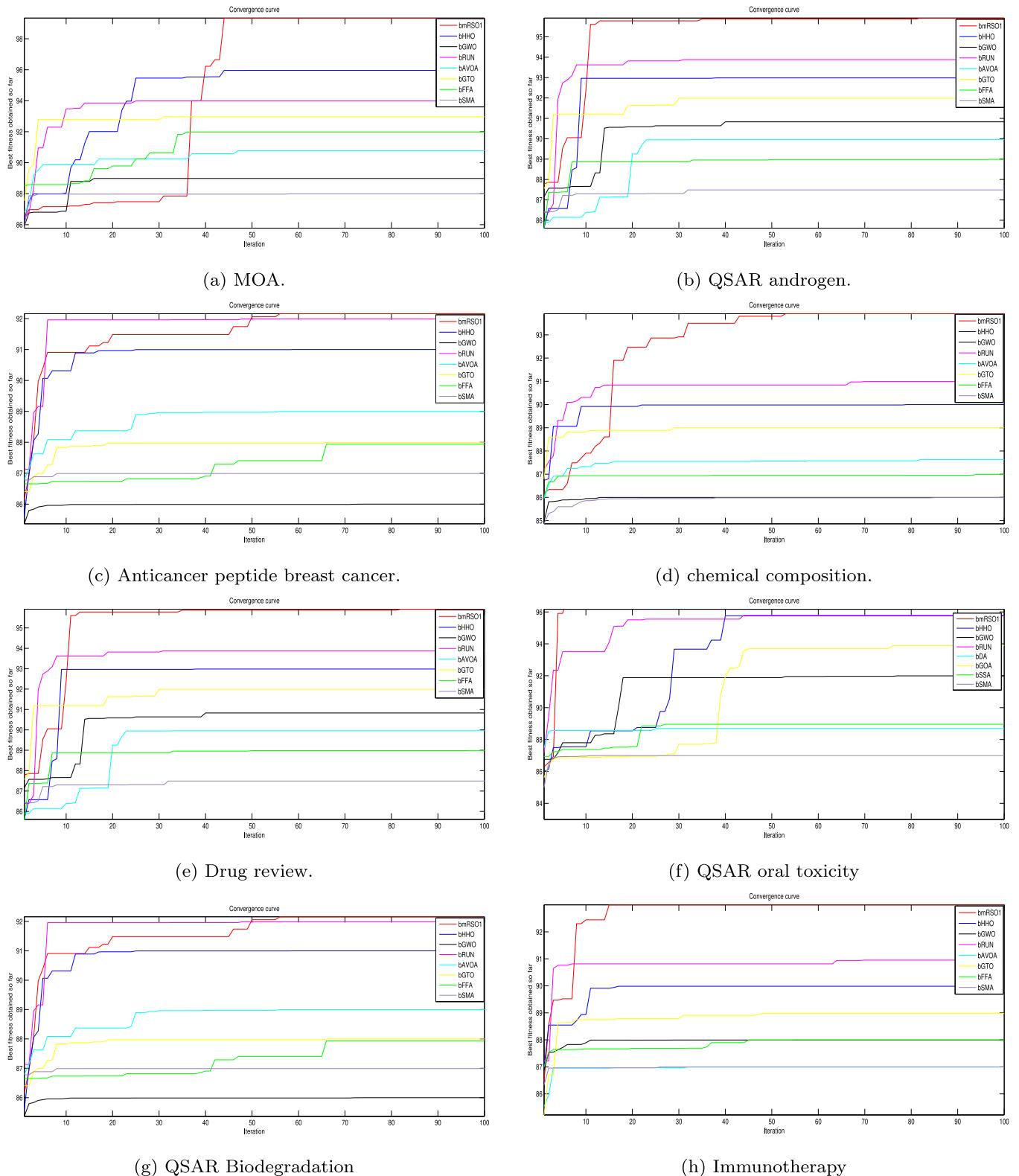


Fig. 14. The convergence curves obtained from the proposed bmRSO1-SVM versions and other algorithms over eight datasets.

(AVOA) [28], Runge Kutta Optimizer (RUN) [29], and Slime Mould Algorithm (SMA) [30], are applied to compare with the best bmRSO1-SVM version to declare the performance of the proposed method over the same data. According to Tables 9 to 11, bmRSO1-SVM proposes the best statistical results. Figs. 12 and 14, show

the best convergence curve for bmRSO1-SVM. The bmRSO1-SVM box plot is optimal, according to Figs. 16 and 17.

The experimental results showed that adding TFs does not influence the computational complexity of the modified method because this component runs only once for each solution and in each iteration. Thus our enhancement method is better than

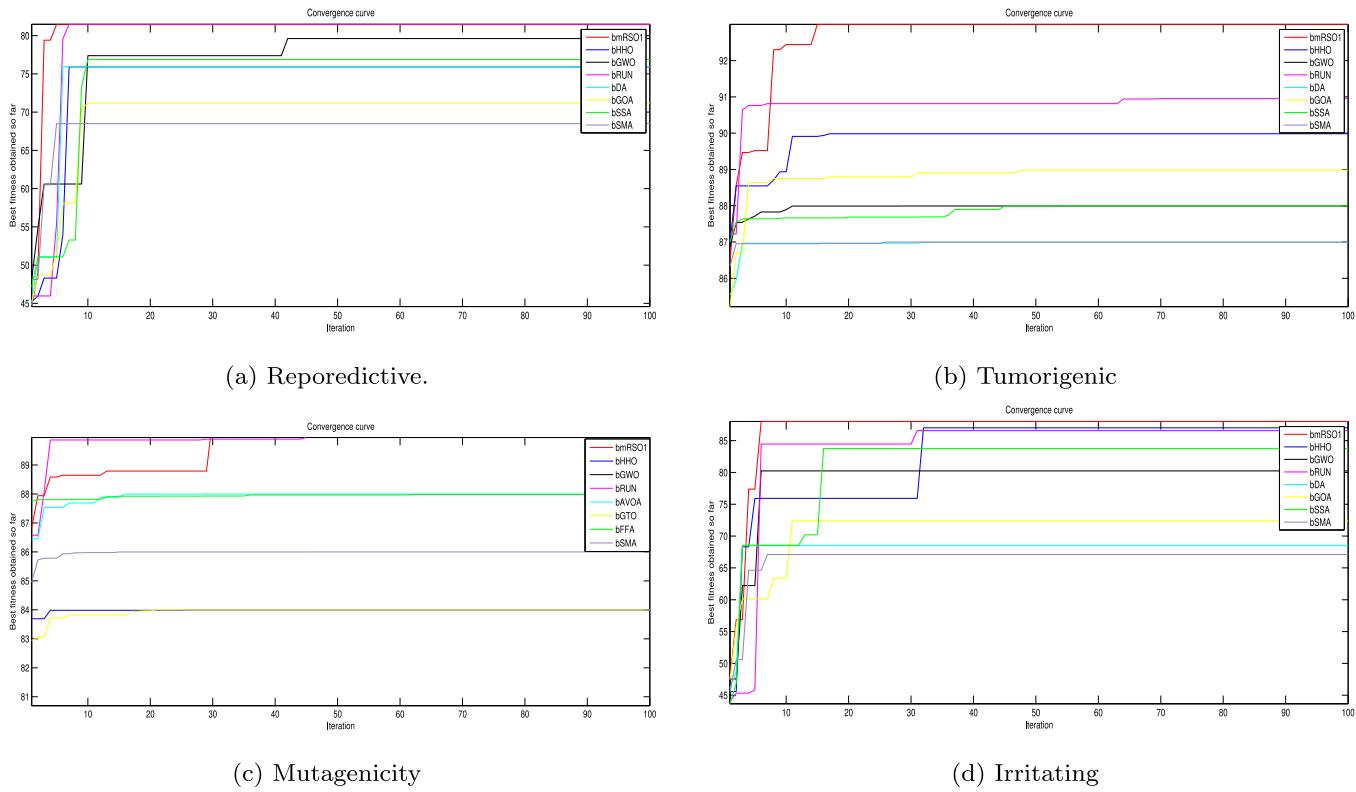


Fig. 15. FDA toxicity convergence curve comparison obtained from the counterparts using SVM on the different stop criteria applied to four toxicity datasets.

Table 7
Toxicity description.

Feature NO	Name
1	Count of Molecular Weight
2	Weight of several molecular
3	Weight with Absolute
4	cLogP (Octanol/Water, partition coefficient)
5	cLogS (Aqueous solubility)
6	H-Acceptors (Hydrogen bond Acceptor)
7	H-Donors (Hydrogen bond donor)
8	Total Surface Area
9	Polar Surface Area
10	Druglikeness
11	Molecular Shape Index
12	Molecular Flexibility
13	Molecular Complexity
14	Non Hydrogen Atoms
15	Non-Carbon/Hydrogen Atoms
16	Metal Atoms
17	Electron Negative Atoms
18	Stereo Centers
19	Rotatable Bonds
20	cRings
21	Aromatic Rings
22	Aromatic Atoms
23	sp ³ -Atoms
24	Symmetric atoms
25	Amides (acid amide)
26	Amines
27	AlkylAmines
28	Aromatic Amines
29	Aromatic Nitrogen
30	Basic Nitrogen
31	Acidic Oxygen

another approaches as indicated by the CPU time criteria in Tables 5, 6, 8, 9, 10 and 11. The analysis indicates that the proposed bmRSO1-SVM method has achieved better results than

its counterparts. The bmRSO2-SVM is the second rank, whereas the RSO is the last rank. For precise analysis and under the same parameter setting, the search agent count was set to 30 for all experiments with a different dimension number according to the dataset dimensions. Compared to other binary MAs methods, bmRSO1-SVM produces the best results; bRUN and bHHO introduce the best results across all datasets in the second stage, but SMA comes in last.

7.1. Advantages and limitations of the proposed bmRSO method

The advantages and drawbacks of the proposed method called bmRSO are detailed in this part. It contains some suggestions for ways to enhance this methodology as well. The experimental results show that MAs are effective optimization techniques. Still, they can also have drawbacks like premature convergence, an unbalanced exploration-exploitation ratio, and a tendency to get stuck in an optimal local region. We researched current methods and features to create a more effective algorithm to improve classification performance and an effective FS approach. The RSO is a relatively new MA shown in the literature to be effective at resolving practical optimization problems. The RSO algorithm's fundamental idea, easy formula, and small parameters make it simple to implement. In [21], it performed better than several algorithms. In particular, RSO has been tested for optimization problems, including on thirty-eight benchmark test functions, which are divided into four main categories: unimodal, multimodal, fixed-dimension multimodal, and CEC-15 special session functions. Although the original RSO has shown well-competitive performance with other state-of-the-art algorithms, it still suffers from some drawbacks. While the original RSO has demonstrated performance that is competitive with other state-of-the-art algorithms, it still has major drawbacks, including:

- Slow convergence.

Table 8

Statistical results of the several algorithms using the SVM with a stopping criterion based on maximum iteration.

Algorithm	Mean	STD	Best	Worst	CPU time	Average selection number
dataset No. 1: Reproductive						
bmRSO1	8.40E+01	2.10E-01	82.300	81.400	0.200	12.400
bmRSO2	8.35E+01	2.40E-01	80.220	79.710	0.210	15.001
bmRSO3	8.30E+01	2.35E-01	81.820	80.250	0.220	12.800
bmRSO4	7.60E+01	2.30E-01	77.701	76.120	0.210	16.230
bmRSO5	7.65E+01	2.33E-01	78.020	77.150	0.241	17.081
bmRSO6	7.75E+01	2.35E-01	79.020	77.221	0.235	15.201
bmRSO7	7.77E+01	2.35E-01	79.040	78.421	0.255	15.561
bmRSO8	7.50E+01	2.45-E01	73.220	71.520	0.214	17.254
RSO	7.33E+01	2.60-E01	71.610	70.940	0.184	18.101
dataset No. 2: Irritating						
bmRSO1	8.45E+01	2.35E-01	87.020	86.400	0.140	13.500
bmRSO2	8.20E+01	2.38E-01	86.540	85.510	0.150	14.980
bmRSO3	8.30E+01	2.36E-01	89.321	88.100	0.155	12.400
bmRSO4	8.27E+01	2.39E-01	87.420	85.421	0.160	14.561
bmRSO5	7.60E+01	3.15E-01	77.520	76.110	0.155	17.981
bmRSO6	7.65E+01	3.10E-01	79.320	78.121	0.155	15.002
bmRSO7	7.67E+01	3.12E-01	79.300	78.100	0.150	18.397
bmRSO8	7.55E+01	3.20-E01	73.320	72.120	0.155	19.890
RSO	7.40E+01	3.30-E01	71.011	70.640	0.123	20.100
dataset No. 3: Mutagenicity						
bmRSO1	8.35E+01	1.10E-01	89.120	87.000	0.140	12.900
bmRSO2	8.30E+01	1.15E-01	86.540	85.110	0.140	14.900
bmRSO3	8.30E+01	1.25E-01	88.520	85.120	0.145	15.800
bmRSO4	8.25E+01	1.20E-01	84.220	83.020	0.147	17.801
bmRSO5	8.30E+01	1.18E-01	87.520	86.410	0.144	16.120
bmRSO6	8.32E+01	1.27E-01	85.020	83.501	0.147	16.520
bmRSO7	8.25E+01	1.22E-01	83.120	81.520	0.148	17.006
bmRSO8	8.30E+01	1.15-E01	87.900	84.520	0.150	16.186
RSO	8.10E+01	1.30-E01	82.011	81.010	0.130	18.321
dataset No. 4: Tumorigenic						
bmRSO1	9.20E+01	1.02E-01	93.420	91.000	0.140	11.800
bmRSO2	8.90E+01	1.05E-01	88.440	86.510	0.144	14.001
bmRSO3	8.60E+01	1.15E-01	86.120	84.146	0.150	15.002
bmRSO4	8.65E+01	1.25E-01	86.520	84.120	0.148	16.601
bmRSO5	8.95E+01	1.08E-01	89.620	88.610	0.146	14.581
bmRSO6	9.15E+01	1.03E-01	92.320	91.601	0.149	12.100
bmRSO7	8.92E+01	1.06E-01	89.520	85.320	0.145	14.760
bmRSO8	8.90E+01	1.27-E01	88.900	86.520	0.149	14.003
RSO	8.60E+01	1.25-E01	85.011	83.110	0.131	17.761

Table 9

Statistical results of the several algorithms using the SVM with a stopping criterion based on maximum iteration.

Algorithm	Mean	STD	Best	Worst	CPU time	Average selection number
dataset No. 1: MonoAmine Oxidase (MAO)						
bmRSO1	9.60E+01	1.03E-01	98.201	96.190	0.570	20.001
bHHO	9.60E+01	1.17E-01	96.010	93.070	0.581	22.201
bGWO	9.39E+01	1.42E-01	89.120	88.060	0.590	23.311
bFFA	9.56E+01	1.19E-01	92.150	90.201	0.595	24.100
bGTO	9.50E+01	1.20E-01	93.120	90.201	0.596	23.001
bAVOA	9.56E+01	1.27E-01	92.120	91.005	0.671	25.500
bRUN	9.52E+01	1.10E-01	95.100	94.204	0.580	21.100
bSMA	8.90E+01	1.50E-01	88.130	87.100	0.610	25.600
dataset No. 2: Drug Review						
bmRSO1	9.40E+01	1.08E-01	95.100	92.100	0.110	2.001
bHHO	9.00E+01	1.27E-01	89.120	88.140	0.120	2.100
bGWO	8.89E+01	1.12E-01	86.470	85.100	0.130	2.700
bFFA	9.10E+01	1.17E-01	87.110	86.110	0.150	3.100
bGTO	9.00E+01	1.16E-01	89.100	88.500	0.140	3.700
bAVOA	8.90E+01	1.27E-01	88.200	87.005	0.160	3.500
bRUN	9.25E+01	1.10E-01	91.250	90.100	0.105	2.600
bSMA	8.70E+01	1.37-E01	86.470	85.020	0.170	2.900

(continued on next page)

- Trapping of a local optimum.
- Unbalanced between exploration and exploitation search.

In this work, we divide the steps of the exploration and exploitation phases and improve the RSO using two methodologies. To evolve both local and global search during the optimization

Table 9 (continued).

Algorithm	Mean	STD	Best	Worst	CPU time	Average selection number
dataset No. 3: QSAR Oral						
bmRSO1	9.70E+01	1.08E-01	96.100	93.120	0.420	24.300
bHHO	9.55E+01	1.19E-01	94.700	92.120	0.415	24.700
bGWO	9.45E+01	1.25E-01	90.120	89.001	0.425	24.900
bFFA	9.50E+01	1.21E-01	92.210	91.120	0.430	26.100
bGTO	8.85E+01	1.30E-01	89.100	88.100	0.441	26.900
bAVOA	9.40E+01	1.37E-01	90.100	89.115	0.450	27.100
bRUN	9.63E+01	1.20E-01	94.210	93.120	0.423	25.100
bSMA	8.80E+01	1.40-E01	87.910	86.920	0.445	27.200
dataset No. 4: QSAR Androgen						
bmRSO1	9.80E+01	1.06E-01	96.120	95.120	0.320	24.500
bHHO	9.15E+01	1.50E-01	92.510	90.110	0.310	24.700
bGWO	9.20E+01	1.61E-01	90.920	88.120	0.325	25.100
bFFA	9.55E+01	1.45E-01	92.210	93.120	0.340	25.300
bGTO	9.60E+01	1.30E-01	91.200	90.220	0.350	26.100
bAVOA	9.50E+01	1.50E-01	90.120	89.105	0.360	26.900
bRUN	9.70E+01	1.20E-01	94.420	93.520	0.310	24.800
bSMA	8.50E+01	1.85-E01	87.320	85.120	0.360	28.500

Table 10

Statistical results of the compared algorithms using the SVM with a stopping criterion based on maximum iteration.

Algorithm	Mean	STD	Best	Worst	CPU time	Average selection number
dataset No. 5: Immunotherapy						
bmRSO1	9.10E+01	1.02E-01	92.100	90.120	0.170	2.001
bHHO	9.05E+01	1.10E-01	91.020	89.120	0.160	3.001
bGWO	8.97E+01	1.20E-01	86.410	85.120	0.175	3.300
bFFA	8.90E+01	1.25E-01	88.310	87.120	0.180	4.100
bGTO	8.90E+01	1.30E-01	88.200	87.220	0.190	4.500
bAVOA	8.90E+01	1.35E-01	89.100	86.110	0.200	6.100
bRUN	9.05E+01	1.10E-01	91.520	90.120	0.172	3.400
bSMA	8.55E+01	1.50-E01	87.021	82.150	0.190	6.300
dataset No. 6: Chemical Composition						
bmRSO1	9.25E+01	1.01E-01	93.100	91.100	0.150	6.923
bHHO	8.80E+01	1.30E-01	90.400	88.510	0.151	7.120
bGWO	8.75E+01	1.20E-01	86.500	85.120	0.140	7.900
bFFA	8.90E+01	1.31E-01	87.500	86.220	0.170	8.100
bGTO	8.70E+01	1.40E-01	88.120	87.100	0.160	8.900
bAVOA	8.82E+01	1.32E-01	88.020	86.100	0.180	9.200
bRUN	9.20E+01	1.15E-01	92.100	90.100	0.153	7.130
bSMA	8.50E+01	1.50-E01	86.920	85.100	0.190	10.100
dataset No. 7: Anticancer Peptides cancer(Breast cancer)						
bmRSO1	9.35E+01	1.10E-01	94.120	92.120	0.100	0.500
bHHO	9.20E+01	1.24E-01	91.140	90.410	0.105	0.700
bGWO	9.27E+01	1.30E-01	90.820	90.610	0.110	0.800
bFFA	8.90E+01	1.40E-01	87.220	86.020	0.120	0.900
bGTO	8.80E+01	1.45E-01	89.120	87.100	0.110	1.200
bAVOA	8.90E+01	1.40-E01	87.100	86.100	0.130	1.500
bRUN	9.47E+01	1.15E-01	91.100	90.100	0.102	0.800
bSMA	8.75E+01	1.50E-01	86.100	85.120	0.140	1.900
dataset No. 8: QSAR Biodegradation						
bmRSO1	9.35E+01	1.10E-01	92.120	92.000	0.240	19.781
bHHO	9.20E+01	1.14E-01	90.221	90.100	0.238	20.100
bGWO	9.20E+01	1.14E-01	90.240	90.100	0.245	20.500
bFFA	8.85E+01	1.20E-01	89.510	88.100	0.249	24.001
bGTO	8.70E+01	1.25E-01	88.100	87.710	0.253	27.080
bAVOA	8.72E+01	1.25E-01	87.120	85.220	0.255	29.100
bRUN	9.30E+01	1.12E-01	91.100	90.120	0.243	20.150
bSMA	8.70E+01	1.25-E01	88.100	87.522	0.260	31.001

process, prevent being trapped in local optima, achieve a balance between them, and speed up RSO convergence, multiple exploration and exploitation strategies are included. For the chemical and toxicity dataset, we used it as an FS for evaluating the suggested approach based on the SVM classifier. The comparative results show that the suggested bmRSO1-SVM is an effective and popular FS method.

The following points list the advantages of the proposed bmRSO versions:

- Improve the convergence speed of RSO.

- Enhances the exploration ability of RSO.
- Improving the exploration and exploitation steps using the advantages of the genetic operator to avoid the local search.
- Also, using the TFs emphasizes the importance of the best solution and moves towards the global best solution, which can effectively balance exploration and exploitation.
- Reduces the possibility of RSO falling into the local optima.
- The proposed bmRSO-SVM versions help in solving dataset problems and give competitive results compared to their counterpart algorithms. The best bmRSO1-SVM version has

Table 11

Statistical results of the compared algorithms using the SVM with a stopping criterion based on maximum iteration.

Algorithm	Mean	STD	Best	Worst	CPU time	Average selection number
dataset No. 1: Reproductive						
bmRSO1	8.40E+01	2.10E-01	82.300	81.400	0.200	12.400
bHHO	8.30E+01	2.45E-01	80.120	79.100	0.210	12.600
bGWO	7.90E+01	2.50E-01	79.140	78.120	0.220	13.100
bFFA	7.60E+01	2.25E-01	77.100	76.100	0.230	14.500
bGTO	7.85E+01	2.58E-01	78.120	77.100	0.240	14.650
bAVOA	7.90E+01	2.50E-01	79.120	77.120	0.260	15.200
bRUN	8.35E+01	2.30E-01	81.120	79.129	0.205	12.700
bSMA	7.50E+01	2.70-E01	73.100	71.100	0.270	19.100
dataset No. 2: Irritating						
bmRSO1	8.45E+01	2.35E-01	87.100	86.120	0.140	13.500
bHHO	8.10E+01	2.40E-01	86.100	85.210	0.130	13.900
bGWO	8.20E+01	2.43E-01	87.020	85.022	0.143	14.100
bFFA	7.80E+01	3.02E-01	79.320	78.121	0.150	15.100
bGTO	7.70E+01	3.15E-01	77.520	76.110	0.155	16.200
bAVOA	7.87E+01	3.10E-01	79.300	78.100	0.160	18.200
bRUN	8.20E+01	2.41E-01	86.320	85.100	0.135	13.600
bSMA	7.60E+01	3.25-E01	73.320	72.120	0.165	21.100
dataset No. 3: Mutagenicity						
bmRSO1	8.35E+01	1.10E-01	89.120	87.000	0.140	12.900
bHHO	8.25E+01	1.17E-01	86.100	85.210	0.132	13.100
bGWO	8.30E+01	1.15-E01	87.700	84.108	0.143	13.300
bFFA	8.30E+01	1.12E-01	87.620	86.100	0.145	14.100
bGTO	8.29E+01	1.20E-01	85.010	83.200	0.150	14.500
bAVOA	8.20E+01	1.23E-01	84.310	83.100	0.152	15.920
bRUN	8.34E+01	1.10E-01	88.100	85.220	0.140	13.100
bSMA	8.20E+01	1.29E-01	83.100	81.170	0.160	20.100
dataset No. 4: Tumorigenic						
bmRSO1	9.20E+01	1.02E-01	93.420	91.000	0.140	11.800
bHHO	8.90E+01	1.05E-01	89.220	88.100	0.137	11.900
bGWO	8.80E+01	1.07-E01	88.900	86.520	0.142	12.100
bFFA	8.92E+01	1.04E-01	89.100	85.000	0.145	12.300
bGTO	8.84E+01	1.07E-01	88.140	86.510	0.150	12.600
bAVOA	8.75E+01	1.30E-01	86.100	84.224	0.150	14.900
bRUN	9.15E+01	1.03E-01	92.220	91.600	0.143	11.900
bSMA	8.70E+01	1.45E-01	86.100	84.120	0.162	18.910

a significant performance advantage in chemical and toxicity data compared to various widely used algorithms that represent an application in the field of improving the prediction of chemical data.

The experiments show that, when compared to other current optimization strategies, bmRSO1 produces results that are more accurate. We use the CEC'2020 test suite, which maintains extremely complex problems, for a thorough evaluation of the proposed technique. In order to evaluate the efficacy of the proposed bmRSO-SVM versions in practical settings, we also applied them to chemical and toxicity data problems. The comparison results show that the bmRSO1-SVM is an effective and attractive solution for resolving dataset problems and several optimization problems. The proposed bmRSO has advantages, but it also has certain drawbacks, which are detailed below:

- Hybridizing the genetic operator and adding TFs parameters would have a significant effect. Adding genetic operators may increase computational costs. The two main weaknesses in TFs are the population algorithm operators and the fact that they are the only components that contribute to explosion and exploitation.
- Referencing the No-Free-Lunch (NFL) theorem, one can support the idea that no better optimization technique can solve every optimization problem. The authors conclude that although the bmRSO1 algorithm outperforms many other current and well-known algorithms, it follows the same rule as the other MAs approaches.

To improve the limitations of the proposed algorithm, we can apply the following:

- Hybridize the RSO algorithm with genetic operators (mutation, crossover, and selection) for more diversity to enhance performance and avoid the local search.
- Then, add TFs that are used to convert the search space from continuous to discrete. Based on an appropriate TF approach, velocity performance is improved. First, a number of searches use TF techniques, which are thought to be algorithm-independent and have no impact on algorithmic search behavior. Second, because this component is only run once for each solution and in each iteration, adding the TFs has no effect on the algorithm's computational complexity.
- TFs can improve local and global searches and control exploitation and exploration searches.
- The proposed bmRSO1-SVM version helps in solving the chemical and toxicity dataset problems and gives competitive results compared to their counterpart versions, and the best bmRSO1-SVM version has a significant performance advantage in the dataset compared to various widely used algorithms that represent an application in the field of improving the prediction of chemical and toxicity data.

8. Conclusions and future work

This paper proposes an efficient alternative binary version of Rat Swarm Optimization (RSO) called bmRSO. The bmRSO versions are produced from the combination of genetic operators and transfer functions (TFs) as a search strategy to overcome the drawbacks of the original RSO. Then the bmRSO versions are hybrid with the Support Vector Machine (SVM) as a classifier process (called bmRSO-SVM) for the classification purpose.

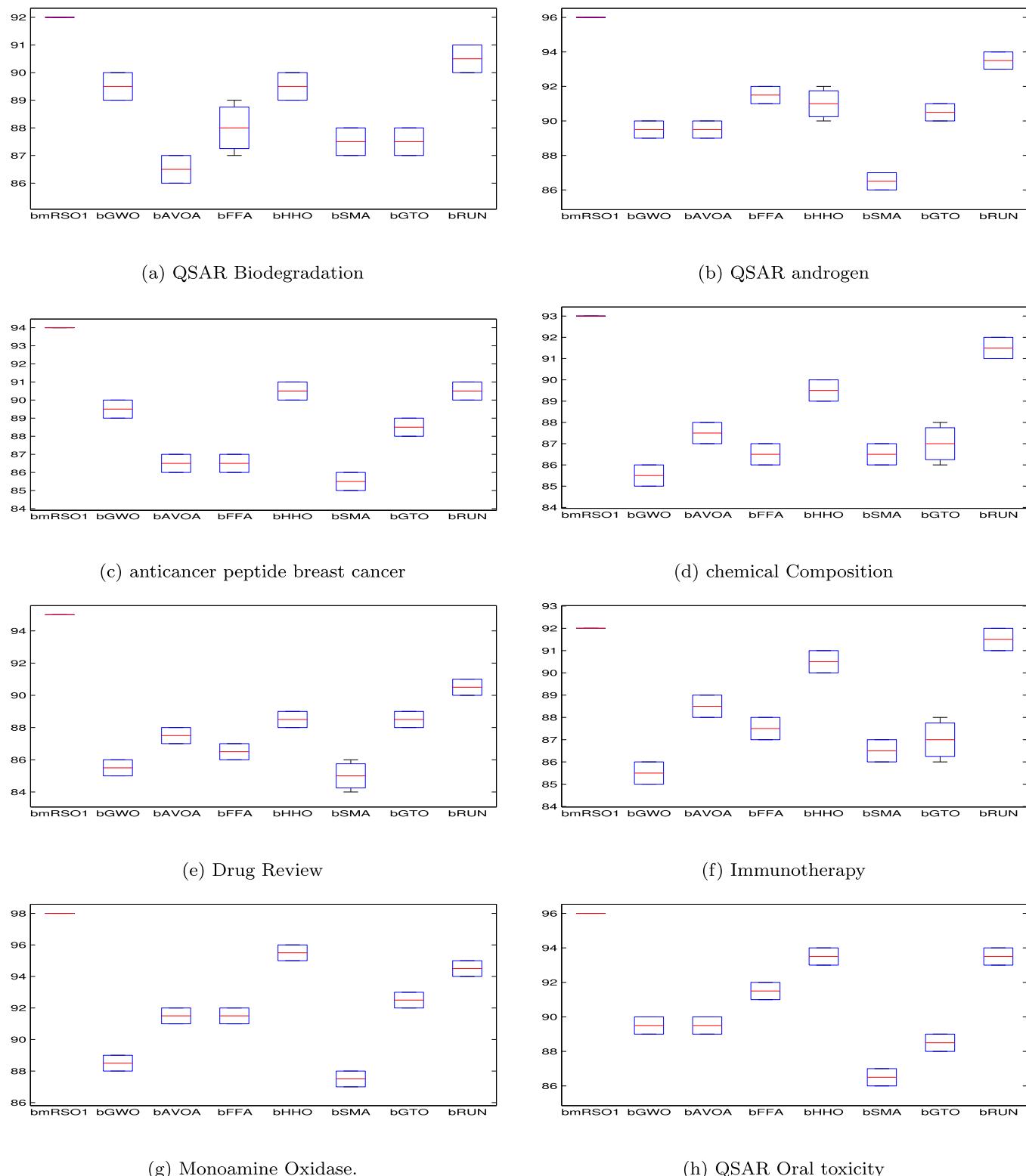


Fig. 16. The box plot comparison obtained from the counterpart algorithms using SVM is based on the stopping criteria applied over eight datasets.

The fitness function employed permits minimizing the feature number that consequently increases the accuracy of the classification process. The performance of the proposed bmRSO versions is evaluated over the CEC20 benchmark, and some chemical datasets, especially toxic data, are evaluated in this paper. Additionally, bmRSO-SVM versions are used as a tool in drug development. Eight chemical datasets and four FDA toxicity data were

used. The bmRSO1-SVM attained the optimal maximum accuracy and reduced the number of features in all datasets representing our objective function according to the statistical results, convergence curves, and box plots. The MOA dataset reaches the highest results for bmRSO1-SVM; so it reached high accuracy results of 98.201% and selected 20.001 features and the reproductive data, which has lower results, the optimal is bmRSO1-SVM which

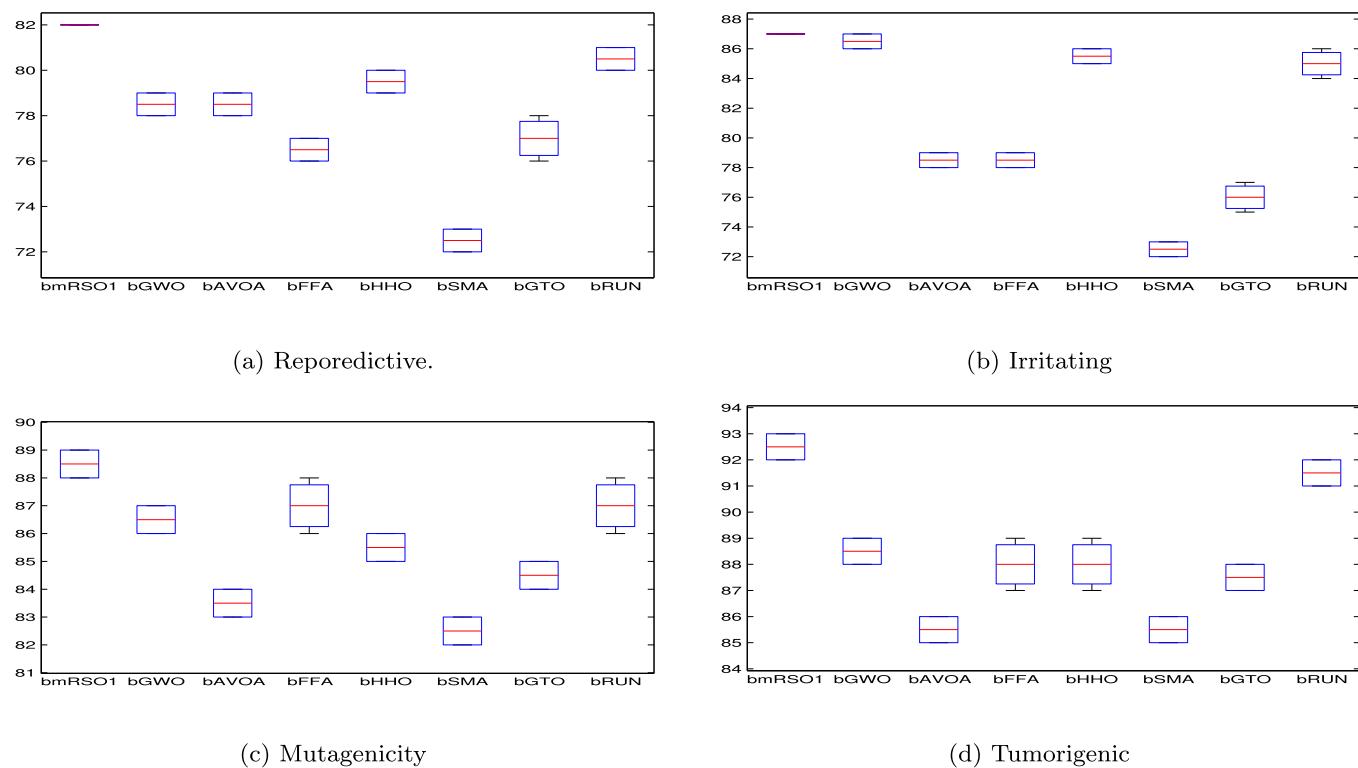


Fig. 17. FDA toxicity box plot comparisons obtained from the counterparts using SVM on the different stop criteria applied to four toxicity datasets.

has 82.30% accuracy and return 12,400 features. The statistical and visualization results revealed the superiority of the bmRSO1, which has higher convergence speeds without increasing additional processing costs across all datasets compared with other versions. Also, the bmRSO1-SVM achieves the best results compared with some binary MAs algorithms such as (HHO, GWO, FFA, AVOA, RUN, and SMA) on the datasets that are used in the paper. In future work, the proposed bmRSO can be applied to multi-objective problems and image processing problems as a developing approach for many drug development problems.

Compliance with ethical standards

This article does not contain any studies with human participants or animals performed by any of the authors.

CRediT authorship contribution statement

Essam H. Houssein: Supervising, Software, Methodology, Conceptualization, Formal analysis, Investigation, Visualization, Writing – review & editing. **Mosa E. Hosney:** Software, Resources, Writing – original draft. **Diego Oliva:** Methodology, Conceptualization, Formal analysis, Writing – review & editing. **Eman M.G. Younis:** Methodology, Conceptualization, Formal analysis, Investigation, Data curation, Writing – review & editing. **Abdelmgeid A. Ali:** Supervision. **Waleed M. Mohamed:** Methodology, Conceptualization, Formal analysis, Investigation, Data Curation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article

Acknowledgment

Funding information is not available. All authors read and approved the final paper.

References

- [1] J. Gasteiger, *Chemoinformatics: Achievements and challenges, a personal view*, *Molecules* 21 (2) (2016) 151.
- [2] T. Katsila, G.A. Spyroulias, G.P. Patrinos, M.-T. Matsoukas, Computational approaches in target identification and drug discovery, *Comput. Struct. Biotechnol. J.* 14 (2016) 177–184.
- [3] S. Forli, R. Huey, M.E. Pique, M.F. Sanner, D.S. Goodsell, A.J. Olson, Computational protein-ligand docking and virtual drug screening with the autodock suite, *Nat. Protoc.* 11 (5) (2016) 905.
- [4] A.U. Khan, et al., Descriptors and their selection methods in QSAR analysis: Paradigm for drug design, *Drug Discov. Today* 21 (8) (2016) 1291–1302.
- [5] A. Bhattacharya, R.T. Goswami, K. Mukherjee, A feature selection technique based on rough set and improvised pso algorithm (PSORS-FS) for permission based detection of android malwares, *Int. J. Mach. Learn. Cybern.* 10 (2019) 1893–1907.
- [6] A. Tharwat, Y.S. Moemen, A.E. Hassanien, Classification of toxicity effects of biotransformed hepatic drugs using whale optimized support vector machines, *J. Biomed. Inform.* 68 (2017) 132–149.
- [7] M. Abd Elaziz, Y.S. Moemen, A.E. Hassanien, S. Xiong, Toxicity risks evaluation of unknown fda biotransformed drugs based on a multi-objective feature selection approach, *Appl. Soft Comput.* 97 (2020) 105509.
- [8] T. Sander, J. Freyss, M. von Korff, C. Rufener, Datawarrior: An open-source program for chemistry aware data visualization and analysis, *J. Chem. Inf. Model.* 55 (2) (2015) 460–473.
- [9] E.H. Houssein, E. Çelik, M.A. Mahdy, R.M. Ghoniem, Self-adaptive equilibrium optimizer for solving global, combinatorial, engineering, and multi-objective problems, *Exp. Syst. Appl.* 195 (2022) 116552.
- [10] E.H. Houssein, A. Sayed, Dynamic candidate solution boosted beluga whale optimization algorithm for biomedical classification, *Mathematics* 11 (3) (2023) 707.

- [11] E.H. Houssein, A.G. Gad, Y.M. Wazery, P.N. Suganthan, Task scheduling in cloud computing based on meta-heuristics: Review, taxonomy, open challenges, and future trends, *Swarm Evol. Comput.* 62 (2021) 100841.
- [12] K. Rajwar, K. Deep, S. Das, An exhaustive review of the metaheuristic algorithms for search and optimization: Taxonomy, applications, and open challenges, *Artif. Intell. Rev.* (2023) 1–71.
- [13] V. Beiranvand, W. Hare, Y. Lucet, Best practices for comparing optimization algorithms, *Opt. Eng.* 18 (2017) 815–848.
- [14] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, *Future Gener. Comput. Syst.* 97 (2019) 849–872.
- [15] B. Crawford, R. Soto, G. Astorga, J. García, C. Castro, F. Paredes, Putting continuous metaheuristics to work in binary search spaces, *Complexity* (2017).
- [16] C.-W. Chen, Y.-H. Tsai, F.-R. Chang, W.-C. Lin, Ensemble feature selection in medical datasets: Combining filter, wrapper, and embedded feature selection results, *Expert Syst.* 37 (5) (2020) e12553.
- [17] Y.B. Wah, N. Ibrahim, H.A. Hamid, S. Abdul-Rahman, S. Fong, Feature selection methods: Case of filter and wrapper approaches for maximising classification accuracy, *Pertanika J. Sci. Technol.* 26 (1).
- [18] G. Chandrashekhar, F. Sahin, A survey on feature selection methods, *Comput. Electr. Eng.* 40 (1) (2014) 16–28.
- [19] K. Balakrishnan, R. Dhanalakshmi, U. Mahadeo Khaire, Analysing stable feature selection through an augmented marine predator algorithm based on opposition-based learning, *Expert Syst.* 39 (1) (2022) e12816.
- [20] I.M. EL-Hasnony, M. Elhoseny, Z. Tarek, A hybrid feature selection model based on butterfly optimization algorithm: COVID-19 as a case study, *Expert Syst.* 39 (3) (2022) e12786.
- [21] G. Dhiman, M. Garg, A. Nagar, V. Kumar, M. Dehghani, A novel algorithm for global optimization: Rat swarm optimizer, *J. Ambient Intell. Humaniz. Comput.* 12 (8) (2021) 8457–8482.
- [22] M. Mafarja, I. Aljarah, A.A. Heidari, H. Faris, P. Fournier-Viger, X. Li, S. Mirjalili, Binary dragonfly optimization for feature selection using time-varying transfer functions, *Knowl.-Based Syst.* 161 (2018) 185–204.
- [23] P. Agrawal, T. Ganesh, D. Oliva, A.W. Mohamed, S-shaped and v-shaped gaining-sharing knowledge-based algorithm for feature selection, *Appl. Intell.* 52 (1) (2022) 81–112.
- [24] Y. Guo, Z. Zhang, F. Tang, Feature selection with kernelized multi-class support vector machine, *Pattern Recognit.* 117 (2021) 107988.
- [25] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey Wolf Optimizer, *Adv. Eng. Softw.* 69 (2014) 46–61.
- [26] H. Shayanfar, F.S. Gharehchopogh, Farmland fertility: A new metaheuristic algorithm for solving continuous optimization problems, *Appl. Soft Comput.* 71 (2018) 728–746.
- [27] B. Abdollahzadeh, F. Soleimanian Gharehchopogh, S. Mirjalili, Artificial gorilla troops optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems, *Int. J. Intell. Syst.* 36 (10) (2021) 5887–5958.
- [28] B. Abdollahzadeh, F.S. Gharehchopogh, S. Mirjalili, African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems, *Comput. Ind. Eng.* 158 (2021) 107408.
- [29] I. Ahmadianfar, A.A. Heidari, A.H. Gandomi, X. Chu, H. Chen, Run beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method, *Expert Syst. Appl.* 181 (2021) 115079.
- [30] S. Li, H. Chen, M. Wang, A.A. Heidari, S. Mirjalili, Slime mould algorithm: A new method for stochastic optimization, *Future Gener. Comput. Syst.*
- [31] M. Hassan Baig, K. Ahmad, S. Roy, J. Mohammad Ashraf, M. Adil, M. Haris Siddiqui, S. Khan, M. Amjad Kamal, I. Provažník, I. Choi, Computer aided drug design: Success and limitations, *Curr. Pharmaceutical Des.* 22 (5) (2016) 572–581.
- [32] A.A. Jamali, R. Ferdousi, S. Razzaghi, J. Li, R. Safdari, E. Ebrahimie, Drug-miner: Comparative analysis of machine learning algorithms for prediction of potential druggable proteins, *Drug Discov. Today* 21 (5) (2016) 718–724.
- [33] S. Yuan, H.S. Chan, S. Filipek, H. Vogel, Pymol and inkscape bridge the data and the data visualization, *Structure* 24 (12) (2016) 2041–2042.
- [34] F. Gräß er, S. Kallumadi, H. Malberg, S. Zaunseder, Aspect-based sentiment analysis of drug reviews applying cross-domain and cross-data learning, in: *Proceedings of the 2018 International Conference on Digital Health*, 2018, pp. 121–125.
- [35] M.J. Martínez, J.S. Dussaut, I. Ponzoni, Bioclustering as strategy for improving feature selection in consensus QSAR modeling, *Electron. Notes Discrete Math.* 69 (2018) 117–124.
- [36] G.B. Goh, N.O. Hodas, A. Vishnu, Deep learning for computational chemistry, *J. Comput. Chem.* 38 (16) (2017) 1291–1307.
- [37] M.V. Sabando, I. Ponzoni, A.J. Soto, Neural-based approaches to overcome feature selection and applicability domain in drug-related property prediction, *Appl. Soft Comput.* 85 (2019) 105777.
- [38] A.O. Balogun, S. Basri, S.J. Abdulkadir, A.S. Hashim, Performance analysis of feature selection methods in software defect prediction: A search method approach, *Appl. Sci.* 9 (13) (2019) 2764.
- [39] M.M. Mafarja, S. Mirjalili, Hybrid whale optimization algorithm with simulated annealing for feature selection, *Neurocomputing* 260 (2017) 302–312.
- [40] S.S. Shekhawat, H. Sharma, S. Kumar, A. Nayyar, B. Qureshi, BSSA: Binary salp swarm algorithm with hybrid data transformation for feature selection, *Ieee Access* 9 (2021) 14867–14882.
- [41] H.M. Zawbaa, E. Emara, C. Grosan, V. Snasel, Large-dimensionality small-instance set feature selection: A hybrid bio-inspired heuristic approach, *Swarm Evol. Comput.* 42 (2018) 29–42.
- [42] J. Cai, J. Luo, S. Wang, S. Yang, Feature selection in machine learning: A new perspective, *Neurocomputing* 300 (2018) 70–79.
- [43] F. Kılıç, Y. Kaya, S. Yıldırım, A novel multi population based particle swarm optimization for feature selection, *Knowl.-Based Syst.* 219 (2021) 106894.
- [44] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, *Knowl.-Based Syst.* 89 (2015) 228–249.
- [45] O. Abedinia, N. Amjad, A. Ghasemi, A new metaheuristic algorithm based on shark smell optimization, *Complexity* 21 (5) (2016) 97–116.
- [46] P. Mehta, B.S. Yıldız, S.M. Sait, A.R. Yıldız, Hunger games search algorithm for global optimization of engineering design problems, *Mater. Test.* 64 (4) (2022) 524–532.
- [47] E.H. Houssein, E. Saber, A.A. Ali, Y.M. Wazery, Centroid mutation-based search and rescue optimization algorithm for feature selection and classification, *Expert Syst. Appl.* 191 (2022) 116235.
- [48] M. Abd Elaziz, D. Yousri, Automatic selection of heavy-tailed distributions-based synergy henry gas solubility and harris hawk optimizer for feature selection: Case study drug design and discovery, *Artif. Intell. Rev.* 54 (6) (2021) 4685–4730.
- [49] S. Mirjalili, A. Lewis, S-shaped versus v-shaped transfer functions for binary particle swarm optimization, *Swarm Evol. Comput.* 9 (2013) 1–14.
- [50] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, *IEEE Trans. Evol. Comput.* 1 (1) (1997) 67–82.
- [51] K.M. Ng, R. Gani, Chemical product design: Advances in and proposed directions for research and teaching, *Comput. Chem. Eng.* 126 (2019) 147–156.
- [52] H. Rao, X. Shi, A.K. Rodrigue, J. Feng, Y. Xia, M. Elhoseny, X. Yuan, L. Gu, Feature selection based on artificial bee colony and gradient boosting decision tree, *Appl. Soft Comput.* 74 (2019) 634–642.
- [53] Z. Algamal, M. Qasim, H. Ali, A qsar classification model for neuraminidase inhibitors of influenza a viruses (H1N1) based on weighted penalized support vector machine, *SAR QSAR Environ. Res.* 28 (5) (2017) 415–426.
- [54] R. Rodríguez-Pelárez, M. Vogt, J. Bajorath, Support vector machine classification and regression prioritize different structural features for binary compound activity and potency value prediction, *ACS Omega* 2 (10) (2017) 6371–6379.
- [55] E. Wang, P. Yan, J. Liu, A hybrid chatter detection method based on WPD, SSA, and SVM-PSO, *Shock Vib.* 2020 (2020) 1–13.
- [56] Y. Zhou, Y. Wang, K. Wang, L. Kang, F. Peng, L. Wang, J. Pang, Hybrid genetic algorithm method for efficient and robust evaluation of remaining useful life of supercapacitors, *Appl. Energy* 260 (2020) 114169.
- [57] S. Saremi, S. Mirjalili, A. Lewis, How important is a transfer function in discrete heuristic algorithms, *Neural Comput. Appl.* 26 (3) (2015) 625–640.
- [58] S. Mirjalili, Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, *Neural Comput. Appl.* 27 (4) (2016) 1053–1073.
- [59] S. Mirjalili, G.-G. Wang, Ld.S. Coelho, Binary optimization using hybrid particle swarm optimization and gravitational search algorithm, *Neural Comput. Appl.* 25 (6) (2014) 1423–1435.
- [60] M.J. Islam, X. Li, Y. Mei, A time-varying transfer function for balancing the exploration and exploitation ability of a binary PSO, *Appl. Soft Comput.* 59 (2017) 182–196.
- [61] A. Abdiansah, R. Wardoyo, Time complexity analysis of support vector machines (SVM) in LIBSVM, *Int. J. Comput. Appl.* 128 (3) (2015) 28–34.
- [62] A.W. Mohamed, A.A. Hadi, A.K. Mohamed, N.H. Awad, Evaluating the performance of adaptive gainingsharing knowledge based algorithm on CEC 2020 benchmark problems, in: *2020 IEEE Congress on Evolutionary Computation*, CEC, IEEE, 2020, pp. 1–8.
- [63] J.L. Andersen, C. Flamm, D. Merkle, P.F. Stadler, A software package for chemically inspired graph transformation, in: *International Conference on Graph Transformation*, Springer, 2016, pp. 73–88.
- [64] I.L. Ruiz, M.A. Gómez-Nieto, Advantages of relative versus absolute data for the development of quantitative structure–activity relationship classification models, *J. Chem. Inf. Model.* 57 (11) (2017) 2776–2788.
- [65] T.A. Shehabelddeen, M. Abd Elaziz, A.H. Elsheikh, O.F. Hassan, Y. Yin, X. Ji, X. Shen, J. Zhou, A novel method for predicting tensile strength of friction stir welded AA6061 aluminium alloy joints based on hybrid random vector functional link and Henry gas solubility optimization, *IEEE Access* 8 (2020) 79896–79907.