ORCA – Online Research @ Cardiff

This is an Open Access document downloaded from ORCA, Cardiff University's institutional repository:https://orca.cardiff.ac.uk/id/eprint/155378/

This is the author's version of a work that was submitted to / accepted for publication.

Citation for final published version:

Zhong, Changting, Li, Gang, Meng, Zeng, Li, Haijiang and He, Wanxin 2023. A self-adaptive quantum equilibrium optimizer with artificial bee colony for feature selection. Computers in Biology and Medicine 153 , 106520. 10.1016/j.compbiomed.2022.106520

Publishers page: http://dx.doi.org/10.1016/j.compbiomed.2022.106520

Please note:
Changes made as a result of publishing processes such as copy-editing, formatting and page numbers may not be reflected in this version. For the definitive version of this publication, please refer to the published source. You are advised to consult the publisher's version if you wish to cite this paper.

This version is being made available in accordance with publisher policies. See http://orca.cf.ac.uk/policies.html for usage policies. Copyright and moral rights for publications made available in ORCA are retained by the copyright holders.

A self-adaptive quantum equilibrium optimizer with artificial bee colony for feature selection

Changtingv Zhong[a,b], Gang Li[a,c]*, Zeng Meng[d], Haijiang Li[e], Wanxin He[a]

[a] Department of Engineering Mechanics, State Key Laboratory of Structural Analyses for Industrial Equipment, Dalian University of Technology, Dalian 116024, China

[b] School of Civil Engineering and Architecture, Hainan University, Haikou 570228, China

[c] Ningbo Institute of Dalian University of Technology, Ningbo 315000, China

[d] School of Civil Engineering, Hefei University of Technology, Hefei 230009, China

[e] BIM for smart Engineering Centre, Cardiff School of Engineering, Cardiff University, Queen's Buildings, Cardiff CF24 3AA, Cardiff, CF24 3AA, Whales, UK

Emails: Changting Zhong, zhongct@hainanu.edu.cn; Gang Li (corresponding author), ligang@dlut.edu.cn; Zeng Meng, mengz@hfut.edu.cn; Haijiang Li, lih@cardiff.ac.uk; Wanxin He, hewanxin@dlut.edu.cn;

Abstract: Feature selection (FS) is a popular data pre-processing technique in machine learning to extract the optimal features to maintain or increase the classification accuracy of the dataset, which is a combinatorial optimization problem, requiring a powerful optimizer to obtain the optimum subset. The equilibrium optimizer (EO) is a recent physical-based metaheuristic algorithm with good performance for various optimization problems, but it may encounter premature or the local convergence in feature selection. This work presents a self-adaptive quantum EO with artificial bee colony for feature selection, named SQEOABC. In the proposed algorithm, the quantum theory and the self-adaptive mechanism are employed into the updating rule of EO to enhance convergence, and the updating mechanism from the artificial bee colony is also incorporated into EO to achieve appropriate FS solutions. In the experiments, 25 benchmark datasets from the UCI repository are investigated to verify SQEOABC, which is compared with several state-of-the-art metaheuristic algorithms and the variants of EO. The statistical results of fitness values and accuracy demonstrate that SQEOABC has better performance than the compared algorithms and the variants of EO. Finally, a real-world FS problem from COVID-19 illustrates the effectiveness and superiority of SQEOABC.

Keywords: Features selection; Metaheuristic; Equilibrium optimizer; Quantum theory; Artificial bee colony

1 Introduction

Nowadays, a tremendous amount of information is generated with the huge development of science and technology. Abundant and irrelevant features are existed in data, making the extraction and classification of data becoming more difficult [1]. Feature selection (FS) as a data-processing technique has been developed rapidly in machine learning, which can find the optimal subset of features and reduce irrelevant features [2]. In essence, FS is a combinatorial optimization problem and has 2^M possible combinations in the dataset with M features [3]. With the increasing size and features of a dataset, the exhaustive method is time-consuming especially for high-dimensional problems.

According to the evaluation criteria of features, FS methods can be classified as filter-based method, wrapper-based method, embedded-based method, and hybrid method [4]. The filter-based method conducts the statistical properties of individual features and evaluates the rank in terms of their relevance, which is efficient in FS problems [5]. The wrapper-based method employs the learning algorithm to estimate the significant features, which has higher accuracy to find the optimal feature subset than the filter-based method, but more computational cost than the filter-based method due to more training times [6]. In the embedded-based method, the feature selection is fused on the learning algorithm, where the classification accuracy relies on the optimal feature and the learning algorithm. The hybrid method combines with the filter model and the wrapper model to handle the feature selection problems. Besides, other feature selection methods have been also attracted attention for researchers, such as rough set theory [7] and fuzzy rough set theory [8].

In the FS problem, the optimization techniques are inevitable and very important to obtain the optimal feature [9]. Metaheuristic algorithms are inspired by nature phenomenon, physical rules or biological behaviors, which are commonly utilized for various optimization problems [10][11]. Most metaheuristic algorithms are population-based algorithms, which can find high quality solution from complex optimization problems with the reasonable computational cost during the iterative process, benefiting from the following advantages: derivative-free, problem-independent, good convergence, ease of implementation [12][13]. They are distinguished by the concepts of inspiration and by the searching mechanism, leading to different performances in various optimization problems. Some of popular and recent metaheuristic algorithms include particle swarm optimization (PSO) [14], covariance matrix adaptation evolution strategy (CMA-ES) [15], teaching-learning-based optimization (TLBO) [16], grey wolf optimizer (GWO) [17], artificial bee colony (ABC) [18][19], Harris hawks optimizer (HHO) [20][21], moth-flame optimization [22], slime mould algorithm [23], salp swarm algorithm [24], snake optimizer [25], white shark optimizer

[26], beluga whale optimization [27], and so on. When metaheuristic algorithms are applied for feature selection, they can provide good performance in classification accuracy. For instance, Ke et al. [28] developed an efficient colony optimization with rough set theory for feature selection. Chen et al. [29] proposed discernibility-matrix-based FS method from rough set theory using PSO to obtain the optimal subset with imbalance data. Hu et al. [30] designed two niching strategies (crowding clustering and speciation clustering) into PSO for FS problems, in order to enhance the classification accuracy and obtain better FS solutions. Ghimatgar et al. [31] developed a graph clustering-based ant colony optimization called GCACO to search for better solutions in FS problems, outperforming the ACO and other compared algorithms in classification accuracy. Too and Mirjalili [32] developed an improved dragonfly algorithm with binary operator and hyper learning for solving FS problems, which was verified by benchmark datasets and a COVID-19 case study. Although there exist a lot of research on metaheuristic-based FS methods, new algorithms are still required due to No Free Lunch (NFL) theory [33], indicating that there doesn't exist any metaheuristic algorithm with good performance for all optimization problems. Thus, it motivates us to enhance more powerful algorithms for FS problems.

In 2020, Faramarzi et al. [34] proposed a physical-based metaheuristic algorithm called equilibrium optimizer (EO), inspired by finding the equilibrium states in a dynamic system, which performs good capacity for various optimization problems, such as photovoltaic model [35], image segmentation [36], job scheduling [37], power distribution network reconfiguration [38], and stock market prediction [39]. The results demonstrated that EO has good convergence in benchmark and engineering optimization problems. However, EO still suffers from local stagnation in optimization. In [40], a fractional-order chaotic EO was developed to solve global optimization problems and apply for PID controllers. Dinkar et al. [41] developed an opposition-based Laplacian EO for image segmentation. Liu et al. [42] proposed an improved EO for global optimization and engineering design, with operators of Levy flight, spiral encirclement from whale optimization algorithm, and adaptive proportional mutation strategy. In [43], an opposition-based learning EO algorithm with nonlinear time parameter, chaos theory, and enhanced updating rules was developed for high-dimensional optimization problems. In [44], an information-utilization strengthened EO was proposed for global optimization, outperforming the original EO in benchmark optimization problems. Moreover, researchers also investigated the performance of EO for feature selection. In [45], an enhanced EO algorithm was developed with operators of adaptive β hill-climbing, improved equilibrium pool and U-shaped transfer function, which was verified by 24 datasets from the UCI repository. Ouadfel and Elaziz [46] developed a ReliefF binary EO using local search strategy for high-dimensional FS problems. Elmanakhly [47] designed an improved EO for FS problems, with the enhanced operators of opposition-based learning, S-shaped transfer function and the local search strategy. Sayed et al. [48] proposed a chaotic EO with different transfer functions to solve FS problems. Minocha and Singh [49] developed a modified binary EO enhanced by AV-shape transfer function and kNN for FS problems in the phishing detection system. In [50], a normalized mutual information-based EO enhanced by chaos was proposed for ES problems, verified by 14 high-dimensional datasets. According to the above research works, the modified strategies can enhance the convergence of EO in solving FS problems. However, the balance between the exploration and exploitation of EO for the FS problem is still challenging, and the updating rule of EO is the key point.

In this work, we propose a self-adaptive quantum EO with artificial bee colony called SQEOABC for feature selection, based on the hybridization of EO and ABC with the operators of quantum theorem for updating rules of EO and self-adaptive mechanism for coefficient of EO. SQEOABC is verified on 25 UCI standard datasets, which is compared with some state-of-the-art metaheuristic algorithms and the variants of EO. Besides, a COVID-19 case is utilized to indicate the effectiveness of SQEOABC in feature selection.

The rest of this paper is organized as follows: Section 2 provides the literature review of metaheuristic-based FS methods. Section 3 provides the brief overview of EO, and Section 4 introduces the enhanced operators and details of the proposed algorithm. The experimental results of FS problems are provided in Section 5. Finally, the conclusions and future works are provided in Section 6.

2 Related works

In recent years, a great many of metaheuristic algorithms have been widely proposed for various optimization problems, which can be classified as four parts based on the inspiration from nature [34]: (1) swarm intelligence, (2) evolutionary algorithm, (3) physics-based algorithm, and (4) human-inspired algorithm. In this section, the application of metaheuristic algorithms for FS problem is investigated and classified by the inspiration of metaheuristic algorithms.

Swarm intelligence-based algorithms are inspired from animal behavior in nature. PSO is one of the most well-known metaheuristic algorithms which was proposed by Eberhart and Kennedy [14], to mimic the swarm behaviors of birds or flocks. PSO was also widely used in solving FS problems. Huang and Dun [51] combined PSO with support vector machine (SVM) in feature selection, with continuous and discrete versions, which can find optimal feature subset. In [52], an enhanced PSO with multi-swarm strategy based on SVM classifier was developed to solve FS problems. In [53], a binary PSO with fractional-order velocity, mutation, local and global optimum was developed to incorporate with SVM for FS problems. Xue et al. [54], and Rashno et al. [6] discussed multi-objective PSO for FS problems, respectively. In addition, some development of PSO for FS problems have been investigated, such as the recursive PSO [55], self-adaptive PSO with candidate solution generation strategies (SPS-PSO) [56], and so on. Artificial bee colony (ABC) [18] is a popular swarm-based algorithm mimicking scouting and foraging of honey bees, which is also attracted attention in feature selection. In [57], a binary ABC with similarity coefficient was proposed for FS problems. Furthermore, the enhanced multi-objective ABC based on differential selection and ladder-like sample utilization was also utilized to obtain the optimal features in FS problems [58]. In [59], a self-regulating ABC was developed for FS problems, by embedding the self-regulating strategy into initialization, onlooker bees and scout bees of ABC, and then the random grouping was also incorporated in the proposed algorithm. Dadaneh et al. [60] investigated the performance of ant colony optimization with different classifiers in the unsupervised probabilistic FS problems. Mafarja and Mirjalili [61] developed the whale optimization algorithm (WOA) for feature selection demonstrating that the WOA can achieve better solutions than GA and PSO in most tested datasets. Kundu et al. [62] developed an altruistic WOA for feature selection, which was verified by 8 microarray datasets. Awadallah et al. [63] proposed a binary horse herd optimization algorithm for feature selection, enhanced by binary transfer functions and three crossover operators, which provided competitive solutions of accuracy and efficiency in datasets. Besides, other swarm-based metaheuristic algorithms for FS problems include grey wolf optimizer [64], artificial algae algorithm [65], manta ray foraging optimization [66], butterfly optimization algorithm [67], chimp optimization algorithm [68], rat swarm optimizer [69], shuffled frog leaping algorithm [70], slime mould algorithm [71], and so on.

In the second category, evolutionary algorithms mimic the evolutionary process of biology. Genetic algorithm (GA) as the popular evolutionary algorithm was also applied for feature selection. Yang and Hanavar [72] firstly investigated GA-based FS method, verified by 26 datasets from the UCI repository. Huang and Wang [73] developed the GA-based support vector machines for feature selection, with better accuracy than the grid algorithm. In [74], the nest-GA with wrapper-FS method was developed for FS problems, including high-dimensional cancer Microarray datasets. Carvalho [75] combined GA with a convolutional neural network to find the optimal features in computed tomography (CT) images, demonstrating its good accuracy in feature selection. Meenachi and Ramakrishnan [76] proposed two fuzzy rough set-based feature selection methods based on Tabu search with ACO and GA, respectively, which were verified by 4 cancer medical datasets. In this category, differential evolution (DE) is also popular in solving FS problems. Al-Ani [77] developed an improved DE with a wheel-based search strategy in solving FS problems. In [78], a binary DE with self-learning strategy was developed for multi-objective FS problems. In [3], a self-adaptive weighted DE called SaWDE was proposed to deal with high-dimensional FS problems, verified by UCI datasets with above 1000 dimensions. Other evolutionary algorithms were also developed for FS problems, such as evolution strategy [79] and biogeography-based optimization [80].

In the third category, the physics-based algorithm mimicking physics of law in nature is also popular in the field of feature selection. Debuse [81] proposed a FS method based on information gain and simulated annealing, improving the performance of data mining system and reducing the number of features. Meiri and Zahavi [82] employed simulated annealing to a linear regression model to find the optimal features in marketing applications. Han et al. [83] proposed a modified gravitational search algorithm with piecewise linear chaotic map, embedded into the wrapper model to solve FS problems. Guha et al. [84] developed the binary version of the gravitational search algorithm with clustering population for FS problems, which was verified by 20 datasets from the UCI repository. Al-rawashdeh [85] developed a hybrid water cycle algorithm with simulated annealing to solve FS problems in spam email detection. Other physics-based algorithms for feature selection consist of multi-verse optimizer [86], charged system search [87], black hole [88] and atom search optimization [89].

The last category is human-based algorithms, usually mimicking human beings, having popular algorithms such as TLBO [15] mimicking the teaching and learning behaviors of students, harmony search [90] inspired by the music performance and the sine cosine algorithm [91] inspired by sine and cosine functions. Inbarani et al. [92] developed a hybrid FS method based on the rough set theory with quick reduct technique, where the improved harmony search algorithm is used to find the optimal subset feature. Shukla et al. [93] developed a hybrid TLBO-simulated annealing algorithm with SVM for feature selection, verified by UCI datasets and a case of gene expression data. Pradhan et al. [94] proposed a multi-class SVM based on improved TLBO to solve FS problems, including enzyme subclass classification. Sameer et al. [95] developed a hybrid binary TLBO algorithm with SVM classification for multi-objective feature selection. Gholami et al. [96] proposed a binary global harmony search with KNN classifier to find feature subsets. Hussain et al. [97] developed an improved sine cosine algorithm hybridized with HHO high-dimensional FS problems. Kale et al. [98] developed four versions of boosting sine cosine algorithms for global optimization and feature selection. In [99], a hybrid sine cosine algorithm embedded with simulated annealing and chaos theory was developed for feature selection of Hadith classification. Furthermore, other human-based algorithms applied for FS problems include imperialist competitive algorithm [100] and group search optimizer [101].

3 Equilibrium optimizer (EO)

In 2020, Faramarzi et al. [34] proposed a novel metaheuristic algorithm called equilibrium optimizer (EO), motivating from the mass balance equation in a control volume from a physical principle, attempting to find the equilibrium state of a system. EO is a powerful optimizer when solving optimization problems due to its updating mechanism, including three phases: initialization, equilibrium pool and concentration update. The details of EO are illustrated as follows.

Step 1: Initialization. In this phase, the positions of each particle are regarded as the concentration of the control volume (C), while a set of particles are generated randomly among the boundaries as:

$$C_{i,j} = c_{\min,j} + r\left(c_{\max,j} - c_{\min,j}\right), \quad i = 1,\ldots,n \quad j = 1,\ldots,d \tag{1}$$

where $C_{i,j}$ represents the position in the j-th dimension of the i-th particle, r is a random number between (0, 1), $c_{min,j}$ and $c_{max,j}$ are the boundaries of each particle in j-th dimension, respectively. The fitness values are evaluated and the particles are sorted after the generation of initialized particles to prepare to construct an equilibrium pool.

Step 2: Equilibrium pool and candidates. To find the final equilibrium state of a system in EO, different best-so-far particles are required to enhance the population diversity. Therefore, an equilibrium pool is constructed. After the initialization phase, four sorted particles with best-so-far fitness values are selected as the candidates in the equilibrium pool. Besides, the average position of the above four particles is calculated and saved in the equilibrium pool simultaneously. After each iteration, the five candidates are updated based on the above mechanism. The equilibrium pool is written as:

$$\boldsymbol{C}_{eq,pool} = \left\{\boldsymbol{C}_{eq(1)}, \boldsymbol{C}_{eq(2)}, \boldsymbol{C}_{eq(3)}, \boldsymbol{C}_{eq(4)}, \boldsymbol{C}_{eq(ave)}\right\} \tag{2}$$

where $\boldsymbol{C}_{eq,pool}$ is the equilibrium pool, $\boldsymbol{C}_{eq(i)}$ (i=1,2,3,4) are four candidates with best-so-far fitness values, and $\boldsymbol{C}_{eq(ave)}$ means the average position of four candidates, which can be expressed as:

$$\boldsymbol{C}_{eq(ave)} = \frac{\boldsymbol{C}_{eq(1)} + \boldsymbol{C}_{eq(2)} + \boldsymbol{C}_{eq(3)} + \boldsymbol{C}_{eq(4)}}{4} \tag{3}$$

In each iteration, a candidate is selected randomly from the equilibrium pool as the best particle in the current iteration. It should be noted that, each candidate in the equilibrium pool has same probability to be selected, providing good diversity in population.

Step 3: Concentration update. To update the concentration of particles, two main terms should be mainly considered in the EO algorithm, exponential term ($\boldsymbol{F}$) and generation rate ($\boldsymbol{G}$). The task of term $\boldsymbol{F}$ is to control the balance between exploration and exploitation, defined as:

$$\boldsymbol{F} = a_1 sign(\boldsymbol{r}_1 - 0.5)\left[\exp(-\boldsymbol{r}_2 t_{EO}) - 1\right] \tag{4}$$

where $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ denote the random vectors at the interval (0, 1), a_1 is a constant to control the exploration capacity, t_{EO} represents the coefficient of EO, which is updated during each iteration:

$$t_{EO} = \left(1 - T/M_{iter}\right)^{\left(a_2 T/M_{iter}\right)} \tag{5}$$

where a_2 is a constant controlling the exploitation capacity, T denotes the current iteration, and M_{iter} represents the maximum iteration. If the a_1 is larger, the exploration of EO algorithm will be strengthened. Similarly, if a_2 is larger, the exploitation of EO will be enhanced. In the basic EO algorithm, a_1 and a_2 are set as 2 and 1, respectively [34].

The generation rate ($\boldsymbol{G}$) is another important term for concentration updating in EO algorithm, for it is utilized to transfer exact solution with enhancing exploitation. The mathematical model of generation rate ($\boldsymbol{G}$) is expressed as:

$$\boldsymbol{G} = -\boldsymbol{P}\left(\boldsymbol{C}_{eq} - \boldsymbol{r}_2\boldsymbol{C}_i\right)\boldsymbol{F}, \quad i = 1, \ldots, n \tag{6}$$

$$\boldsymbol{P} = \begin{cases} 0.5 r_{d1} \cdot \boldsymbol{u} & r_{d2} \geq GP \\ \boldsymbol{0} & r_{d2} < GP \end{cases} \tag{7}$$

where $\boldsymbol{C}_{eq}$ is a selected candidate in the equilibrium pool, $\boldsymbol{C}_i$ is the position of the i-th particle, r_{d1} and r_{d2} denote the random numbers between (0, 1), $\boldsymbol{u}$ represents a unit vector, and GP is the generation probability that affects the exploration and exploitation, which is equal to 0.5.

Therefore, the updating rule in EO is established as:

$$\boldsymbol{C}_i^{new} = \boldsymbol{C}_{eq} + \left(\boldsymbol{C}_i - \boldsymbol{C}_{eq}\right)\boldsymbol{F} + \left(1 - \boldsymbol{F}\right)\boldsymbol{G}/\boldsymbol{r}_3 V \tag{8}$$

where $\boldsymbol{C}_i^{new}$ is the updating position of the i-th particle, $\boldsymbol{r}_3$ denotes the random vector at the interval (0, 1), V is equal to 1. After the updating stage of EO, check the boundary of the updated positions and calculate their fitness values, then employ the memory saving mechanism to adopt the better particles in the updated solutions. The pseudo-code of EO is provided in Algorithm 1.

Algorithm 1: The pseudo-code of EO algorithm

Input: Algorithmic parameters (population size, maximum iteration)

Output: The best solution

Set the algorithmic parameters, input information of optimization problems

Initialize particles by Eq. (1), and construct the equilibrium pool

While $T \leq T_{max}$ **Do**

  For each particle ($\boldsymbol{C}_i$) **Do**

    Calculate the fitness value of i-th particle

    If $f(\boldsymbol{C}_i) < f(\boldsymbol{C}_{eq(1)})$

      $\boldsymbol{C}_{eq(1)} = \boldsymbol{C}_i, f(\boldsymbol{C}_{eq(1)}) = f(\boldsymbol{C}_i)$

Elseif $f(\boldsymbol{C}_i)>f(\boldsymbol{C}_{eq(1)})$ and $f(\boldsymbol{C}_i)<f(\boldsymbol{C}_{eq(2)})$

$\boldsymbol{C}_{eq(2)} = \boldsymbol{C}_i, f(\boldsymbol{C}_{eq(2)})=f(\boldsymbol{C}_i)$

Elseif $f(\boldsymbol{C}_i)>f(\boldsymbol{C}_{eq(1)})$ and $f(\boldsymbol{C}_i)>f(\boldsymbol{C}_{eq(2)})$ and $f(\boldsymbol{C}_i)<f(\boldsymbol{C}_{eq(3)})$

$\boldsymbol{C}_{eq(3)} = \boldsymbol{C}_i, f(\boldsymbol{C}_{eq(3)})=f(\boldsymbol{C}_i)$

Elseif $f(\boldsymbol{C}_i)>f(\boldsymbol{C}_{eq(1)})$ and $f(\boldsymbol{C}_i)>f(\boldsymbol{C}_{eq(2)})$ and $f(\boldsymbol{C}_i)>f(\boldsymbol{C}_{eq(3)})$ and $f(\boldsymbol{C}_i)<f(\boldsymbol{C}_{eq(4)})$

$\boldsymbol{C}_{eq(4)} = \boldsymbol{C}_i, f(\boldsymbol{C}_{eq(4)})=f(\boldsymbol{C}_i)$

End If

End For

Calculate the average position of fifth candidate by Eq. (3)

Update the equilibrium pool by Eq. (2)

Implement the memory saving

Evaluate coefficient of EO using Eq. (5)

For each particle ($\boldsymbol{C}_i$) **Do**

Find a candidate in the equilibrium pool as $\boldsymbol{C}_{eq}$

Compute the exponential term ($\boldsymbol{F}$) using Eq. (4)

Compute $\boldsymbol{P}$ using Eq. (7)

Compute the generation rate ($\boldsymbol{G}$) using Eq. (6)

Obtain the concentration of i-th particle using Eq. (8)

Check the boundary of the updated particle

End For

Find current global best solution

T=T+1

End While

4 The proposed algorithm

As mentioned above, EO is an efficient metaheuristic algorithm for optimization problems. However, EO still encounters the difficulty of the local convergence or premature, especially in solving FS problems, due to the imbalance between exploration and exploitation. Therefore, the self-adaptive quantum equilibrium optimizer-artificial bee colony algorithm (SQEOABC) is proposed to overcome the disadvantage of EO in solving FS problems. SQEOABC integrates three efficient strategies: the self-adaptive mechanism, the quantum theory, and the artificial bee colony algorithm. These modifications and the procedure of the proposed algorithm are described as below.

4.1 Self-adaptive mechanism in coefficient

In the metaheuristic algorithms, the quality of solution is highly influenced by the transition from the global exploration to the local exploitation. In the basic EO algorithm, the exponential term $\boldsymbol{F}$ controls the exploration and exploitation, in which the transfer coefficient t_{EO} in Eq. (5) is one of the important point parameter affecting the performance of algorithm. A large value of coefficient t_{EO} benefits for exploration, while a small value of t_{EO} is helpful for exploitation. Besides, the user-defined parameter a_2 in EO may also lead to inefficient solution. However, the existing formulation of the coefficient may fail to achieve good transition from exploration to exploitation. In this work, a self-adaptive coefficient [43] is adopted in t_{EO} to conduct the transition from exploration to exploitation, which can be calculated as:

$$t_{EO} = \frac{M_{iter} - T}{M_{iter}} \times \left[\left(1 - \sin\theta\right) + \cos\theta/2 \right]^{T/M_{iter}} \tag{9}$$

where θ denotes the amplitude of the change with iterations, expressed as $\theta = \pi/2 \times T/M_{iter}$. The proposed self-adaptive strategy in coefficient is more focused on the exploration to enhance the global search capacity, and it does not require user-defined parameter a_2 for tuning, which is beneficial to convergence of the algorithm.

4.2 Quantum equilibrium optimizer (QEO)

With the development of quantum computing, the quantum theorem was widely utilized in the metaheuristic algorithms to enhance the performances [102][103][104][105]. The quantum theorem is established on the Schrödinger equation, which is very important in the physics. In the Schrödinger equation, the trajectory of particle does not depend on the velocity and position, but on the wave function. Therefore, the behaviors of particles in the quantum EO are different from the behaviors in EO. In this research, the quantum equilibrium optimizer (QEO) is developed, and the updating rule of EO is modified by adding an attractor to enhance the diversity of EO, and to improve the probabilities of finding better solution.

We first briefly introduce the basic quantum behaviors in the quantum PSO (QPSO) [102]. The rule of QPSO means that the quantum behavior exists in each particle, which is determined by a wave function. Based on the convergence analysis of PSO [106], it's assumed that a particle moves in D-dimensional space centered at $p_{i,j}$ on the j-th dimension. Therefore, the wave function ψ could be established as:

$$\psi\left(C_{i,j}^{T+1}\right) = \frac{1}{\sqrt{L_{ij}^{T}}} \exp\left(-\left|C_{i,j} - p_{i,j}\right| / L_{ij}^{T}\right) \tag{10}$$

where ψ represents the probability density function, L_{ij}^{T} denotes the standard deviation of the double exponential distribution, where $p_{i,j}$ is the local attractor.

By Monte Carlo method, the position of the j-dimension in the i-th particle is expressed as:

$$C_{i,j}^{T+1} = p_{i,j} \pm \frac{1}{2} L_{ij}^{T} \ln\left(1/r\right) \tag{11}$$

where r is a random number uniformly distributed over (0, 1). L_{ij}^{T} can be calculated as:

$$L_{ij}^{T} = 2\alpha\left|M_j^T - C_{i,j}\right| \tag{12}$$

where M^T denotes the mean best position, calculated as the mean of local positions of particles:

$$M^T = \left(M_1^T, M_2^T, \ldots, M_D^T\right) = \left(\frac{1}{N}\sum_{i=1}^{N} C_{i,1}^T, \frac{1}{N}\sum_{i=1}^{N} C_{i,2}^T, \ldots, \frac{1}{N}\sum_{i=1}^{N} C_{i,D}^T\right) \tag{13}$$

where N represents the population size. Thus, the updating rule of QPSO in Eq. (11) can be rewritten as:

$$C_{i,j}^{T+1} = p_{i,j} \pm \alpha\left|M_j^T - C_{i,j}\right|\ln\left(1/r\right) \tag{14}$$

In this work, we introduce the quantum mechanism into EO from QPSO. The updating rule of QEO is expressed as:

$$C_{i,j}^{T+1} = \begin{cases} C_{eq,j} + \left(C_{i,j} - C_{eq,j}\right)F_j + \left(1 - F_j\right)G_j/r_3V + \alpha\left|M_j^T - C_{i,j}\right| \times \ln\left(1/r_4\right), & GP < 0.5 \\ C_{eq,j} + \left(C_{i,j} - C_{eq,j}\right)F_j + \left(1 - F_j\right)G_j/r_3V - \alpha\left|M_j^T - C_{i,j}\right| \times \ln\left(1/r_4\right), & otherwise \end{cases} \tag{15}$$

where $C_{eq,j}$ is the j-th position of a selected candidate in the equilibrium pool, F_j and G_j are the j-th position of exponential term and generation rate, respectively, r_4 and r_5 are random numbers between (0, 1), and α is the contraction-expansion coefficient, which is changed with the iteration number T:

$$\alpha = \frac{M_{iter} - T}{M_{iter}} \tag{16}$$

The performance of QEO is different from EO, because the Eq. (13) allows particles to search in the whole space at each iteration, while particles in EO only search in a limited space. Besides, in the main updating rule from Eq. (15), the global convergence of EO is enhanced in the optimization process, by introducing the average best position of particles. In QEO, any lagged particle will not be leaved out, performing more intelligent and cooperative group.

4.3 Artificial bee colony (ABC)

To enhance the performance of EO in FS problems, the updating rule of ABC is also employed into the SQEOABC algorithm. ABC was proposed by Karaboga [18], mimicking the scouting and foraging behaviors of honey bees, which was applied for solving various optimization problems [11][19][107]. In the basic ABC algorithm, the population is composed of honey bees, which are divided as employed bees, onlooker bees and scout bees, and the food source is regarded as the solution of optimization problems.

The main task of employed bees is designed to find the food source. The new generated velocity of each employed bee is based on the neighboring area, which is expressed as:

$$V_{i,j} = X_{i,j} + \varphi\left(X_{i,j} - X_{k,j}\right) \tag{17}$$

where $V_{i,j}$ represents the updating velocity in the j-th dimension of the i-th bee, $X_{i,j}$ and $X_{k,j}$ denote the position in the j-th dimension of the i-th bee and the k-th bee, respectively; φ is calculated as:

$$\varphi = 2r - 1 \tag{17}$$

For onlooker bees, the information is shared. The updating rule of the onlooker bee is similar to the employed bee, while the only difference is that the food source of each onlooker bee is selected from the probability in terms of the fitness values calculated as:

$$P_r\left(\boldsymbol{X}_i\right) = \frac{f\left(\boldsymbol{X}_i\right)}{\sum_{i=1}^{N} f\left(\boldsymbol{X}_i\right)} \tag{19}$$

where $f(\boldsymbol{X}_i)$ is the fitness value of the i-th food source. If the random number r generated from (0,1) is smaller than $P_r(\boldsymbol{X}_i)$, the i-th food source is selected; otherwise, repeat the same process until the (i+1)-th food source is selected.

For scouter bees, when the food source is larger than the abandonment limit parameter, the positions are generated with a food source randomly, as shown in Eq. (1), and the older position is updated.

4.4 Procedure of SQEOABC

The proposed SQEOABC is consisted of EO and three strategies, the self-adaptive coefficient, the quantum theory, and the updating rule of ABC. In SQEOABC, the updating rule of EO is enhanced by the quantum EO to enhance the diversity, which improves the probability of finding the global optimum in the search space; The self-adaptive mechanism is used for evaluating the coefficient of EO to increase the exploration capacity, and the updating mechanism from ABC is also developed into the SQEOABC to avoid the local optimum and enhance the global convergence for FS problems. The procedure can be described as follows:

Step 1: Set the parameters and optimization problem, generate population in the initialization phase, and construct the equilibrium pool.

Step 2: Compute the coefficients of quantum EO and ABC, and obtain the mean positions in population.

Step 3: For the quantum EO phase, calculate the exponential term (F) and generate term (G), and obtain the parameters and positions from the quantum theory, then update the position by Eq. (15); for the ABC phase, calculate φ in recruit phase and then update the position by Eq. (17).

Step 4: Check the boundaries of the updated particles and calculate their fitness values, then sort the particles by the fitness values and update the equilibrium pool with four best-so-far particles and their average position.

Step 5: For the onlooker bee phase of ABC, obtain the probability of fitness and calculate φ in the onlooker bee phase, then update the position using Eq. (17).

Step 6: Calculate the fitness values and update the equilibrium pool.

Step 7: If the maximum iteration is satisfied, output the best solution; otherwise, repeat Steps 2−6 until the T is larger than T_{max}.

The pseudo-code and flowchart of SQEOABC algorithm are described in Algorithm 2, and the flowchart of SQEOABC is provided in Figure 1.

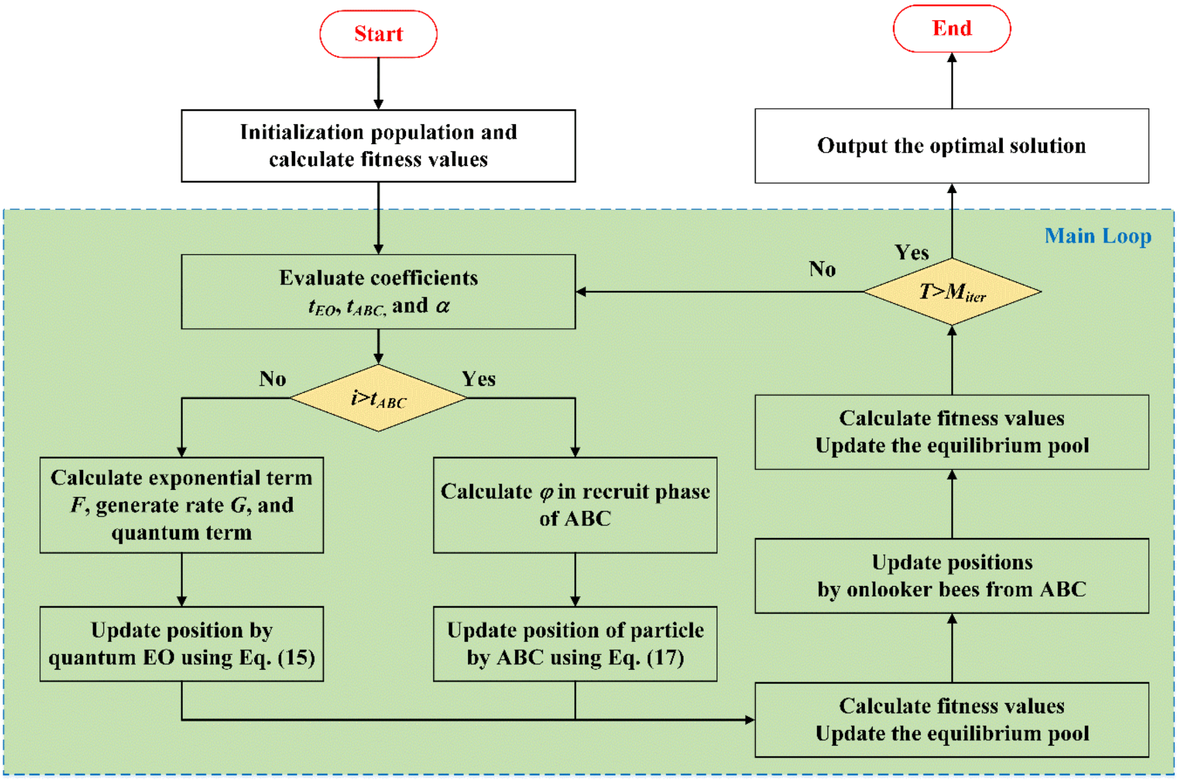


Fig. 1 Flowchart of SQEOABC for feature selection

Algorithm 2: Pseudo-code of SQEOABC

Set parameters of the proposed algorithm (N, M_{iter}, L_A, n_{Alpha}).
Initialize the population of features and construct the equilibrium pool
Set T=1
While $T \leq T_{max}$ **Do**
    Utilize self-adaptive mechanism to evaluate coefficient parameter t_{EO} by Eq. (9)
    Obtain the coefficient $t_{ABC} = t_{EO}$
    Calculate the contraction-expansion coefficient α using Eq. (16) from quantum theory
    Obtain the mean position
    For each particle ($\boldsymbol{C}_i$) **Do**
        If $r>t_{ABC}$
            Find a candidate in the equilibrium pool as $\boldsymbol{C}_{eq}$

Compute the exponential term (***F***) using Eq. (4)
Compute ***P*** using Eq. (7)
Compute the generation rate (***G***) using Eq. (6)
Calculate the standard deviation of the double exponential distribution by Eq. (11)
Update the position by quantum EO by Eq. (15)
Else
Calculate φ in recruit phase of ABC using Eq. (18)
Update the position from recruit bee of ABC by Eq. (17)
End If
End For
For each particle ($\boldsymbol{C}_i$) **Do**
Calculate the fitness value of i-th particle
If $f(\boldsymbol{C}_i)$ is smaller than the previous fitness value of i-th particle
The position and fitness value are updated to C_i and $f(\boldsymbol{C}_i)$; C_{ABC}=C_{ABC}+1
End If
If $f(\boldsymbol{C}_i)<f(\boldsymbol{C}_{eq(1)})$
$\boldsymbol{C}_{eq(1)} = \boldsymbol{C}_i$, $f(\boldsymbol{C}_{eq(1)})=f(\boldsymbol{C}_i)$
Elseif $f(\boldsymbol{C}_i)>f(\boldsymbol{C}_{eq(1)})$ and $f(\boldsymbol{C}_i)<f(\boldsymbol{C}_{eq(2)})$
$\boldsymbol{C}_{eq(2)} = \boldsymbol{C}_i$, $f(\boldsymbol{C}_{eq(2)})=f(\boldsymbol{C}_i)$
Elseif $f(\boldsymbol{C}_i)>f(\boldsymbol{C}_{eq(1)})$ and $f(\boldsymbol{C}_i)>f(\boldsymbol{C}_{eq(2)})$ and $f(\boldsymbol{C}_i)<f(\boldsymbol{C}_{eq(3)})$
$\boldsymbol{C}_{eq(3)} = \boldsymbol{C}_i$, $f(\boldsymbol{C}_{eq(3)})=f(\boldsymbol{C}_i)$
Elseif $f(\boldsymbol{C}_i)>f(\boldsymbol{C}_{eq(1)})$ and $f(\boldsymbol{C}_i)>f(\boldsymbol{C}_{eq(2)})$ and $f(\boldsymbol{C}_i)>f(\boldsymbol{C}_{eq(3)})$ and $f(\boldsymbol{C}_i)<f(\boldsymbol{C}_{eq(4)})$
$\boldsymbol{C}_{eq(4)} = \boldsymbol{C}_i$, $f(\boldsymbol{C}_{eq(4)})=f(\boldsymbol{C}_i)$
End If
End For
Obtain the average position of four candidates using Eq. (3)
Update the equilibrium pool by Eq. (2)
For each particle ($\boldsymbol{C}_i$) **Do**
If $i<n_{\mathrm{Alpha}}$
Obtain the probability by Eq. (17) and select particle k using roulette wheel selection
Calculate φ in the onlooker bees
Update the position by Eq. (17)
Else
Restart babysitters by Eq. (1)
End If
Check boundary of particles
Calculate the fitness values and sort the particles
Find the best four candidates and update the equilibrium pool
End For
T=T+1
End While
Output final global solution

4.5 SQEOABC for FS problems

In this work, the wrapper-based FS method is utilized to evaluate the fitness of solution, with the main task of decreasing the number of selected features, to improve the classification accuracy of dataset. Learning algorithm is required in the wrapper-based FS method. The k-nearest neighbor (kNN, k=5) is adopted in the FS method to measure the classification error, due to simplicity and low computational burden [65]. When solving FS problems, the proposed SQEOABC algorithm is employed to final the optimal solution and evaluate the accuracy of kNN classifier. Thus, the fitness function is defined as:

$$\text{Fit} = \lambda \times \gamma_S + (1-\lambda)\frac{|Y|}{|M|} \tag{20}$$

where γ_S denotes the rate of classification error, $|Y|$ is the number of selected features, $|M|$ is the total number of features, λ means the weight of classification error rate and subjects to (0, 1).

5 Experimental results and discussion

In this section, the proposed SQEOABC is verified by a series of benchmark datasets with different types and features, and compared with several state-of-the-art metaheuristic algorithms. Besides, several variants of EO are also compared with SQEOABC. The performance analysis is based on the statistical results evaluated from all algorithms. Furthermore, a real-world feature selection problem of COVID-19 is also investigated to demonstrate the effectiveness of the proposed algorithm.

5.1 Experimental setups

The performance of SQEOABC-FS algorithm is verified by the experiment of 25 datasets from UCI machine learning repository (http://archive.ics.uci.edu). The 25 datasets are selected from various fields, including Breast Cancer, BreastEW, CongressEW, Exactly, Exactly2, HeartEW, Ionosphere, KrVsKpEW, Lymphography, M-of-n, PenglungEW, Sonar, SpectEW, Tic-tac-toe, Vote, WaveformEW, Wine, Zoo, CNAE, Connectionist Bench Data, Lung Cancer, Optical Recognition of Handwritten, QSAR biodegradation, SPECTFHeart and UJIIndoorLoc. Table 1 summaries the details of these datasets, including number of attributes, samples, classes, and the source fields. Each dataset has two parts: training data and testing data, and the fitness values of accuracy are obtained by the KNN classifier. The details of datasets can also be found at [3][46][98].

To measure the performance of a metaheuristic algorithm in solving FS problems, two metrics are included: mean accuracy, and mean fitness value.

Table 1 Description of datasets

No.	Dataset	Attributes	Samples	Classes	Domain
1	Breast Cancer	9	569	2	Biology
2	BreastEW	30	569	2	Biology
3	CongressEW	16	435	2	Politics
4	Exactly	13	1000	2	Biology
5	Exactly2	13	1000	2	Biology
6	HeartEW	13	270	2	Biology
7	Ionosphere	34	351	2	Electromagnetic
8	KrVsKpEW	36	3196	2	Game
9	Lymphography	18	148	2	Biology
10	M-of-n	13	1000	2	Biology
11	PenglungEW	325	73	2	Biology
12	Sonar	60	208	2	Biology
13	SpectEW	22	267	2	Biology
14	Tic-tac-toe	9	958	2	Game
15	Vote	16	300	2	Politics
16	WaveformEW	40	5000	3	Physics
17	Wine	13	178	3	Chemistry
18	Zoo	16	101	7	Artificial
19	CNAE	856	1080	9	Business
20	Connectionist Bench Data	60	208	2	Physical
21	Lung Cancer	56	32	3	Biology
22	Optical Recognition of Handwritten	64	3823	10	Computer
23	QSAR biodegradation	41	1055	2	Chemistry
24	SPECTF Heart	44	267	2	Life
25	UJIndoor Loc	529	21048	2	Computer

(1) **Mean accuracy:** The mean accuracy (μ_{Acc}) denotes the classification accuracy of classifier when obtaining the selected features from the dataset, which can be described as the average value of results evaluated by several independent runs of the algorithm:

$$\mu_{Acc} = \frac{1}{M}\sum_{k=1}^{M} Acc^k \tag{21}$$

where M means the number of independent runs, and Acc^k is the obtained accuracy in the k-th run. The larger the accuracy is, the better the algorithm performs.

(2) **Mean fitness value:** The mean fitness value (μ_{Fit}) represents the average value of the obtained fitness values among several independent runs of an algorithm:

$$\mu_{Fit} = \frac{1}{M}\sum_{k=1}^{M} Fit^k \tag{22}$$

where Fit^k is the obtained fitness value in the k-th run. The smaller the fitness value is, the better the algorithm performs.

The proposed SQEOABC is compared with 11 different metaheuristic algorithms, including PSO [14], evolution strategy with covariance matrix adaptation (CMAES) [15], grey wolf optimizer (GWO) [17], salp swarm algorithm (SSA) [24], Harris hawks optimization [20], slime mould algorithm (SMA) [23], white shark optimizer (WSO) [26], teaching-learning-slime mould algorithm (TLSMA) [108], directionally driven self-regulating PSO (DDSRPSO) [109], elite evolutionary strategy-HHO (EESHHO) [110] and LSHADE-cnEpSin [111]. For each algorithm, the population size is 20, and the maximum iteration is 100. Each algorithm is performed to solve the FS problem with 30 independent runs, and the statistical results of two performance metrics are obtained (the best results are outlined in boldface). Table 2 provides the parameters of each metaheuristic algorithm in solving FS problems. All algorithms are run in the MATLAB 2018b, under Windows 11, with Intel(R) Core (TM) i9-10900k CPU @ 3.70 GHz and 128 GB RAM.

Table 2 Parameter settings of algorithms

Algorithm	Parameter setting
Common Values	Population size N = 20, Maximum iteration T_{max} = 100
PSO	c_1=2, c_2=2, w=0.7298
GWO	Interval a = [2 0]
CMAES	α=2
HHO	Probability of escaping 0.5
SSA	Leader position update probability p=0.5
SMA	z=0.03
WSO	f_{min}=0.07, f_{max}=0.75, τ=4.125, a_0=6.25, a_1=100, a_2=0.005
TLSMA	z=0.03
DDSRPSO	Probability of attack p_R= 0.5
EESHHO	Interval of E_0 =[−1, 1]
LSHADE-cnEpSin	μF=0.5, μCR=0.5, H=5, $freq$=0.5, ps=0.5, pc=0.4, N_{size}=[18D, 4]
SQEOABC	a_1=2, GP=0.5

5.2 Comparison of SQEOABC with other metaheuristic algorithms

This section provides the statistical results of SQEOABC and 11 compared metaheuristic algorithms. Table 3 summarizes the statistical results of fitness values obtained by SQEOABC and compared algorithms, including mean, STD and rank values, where the best results among all algorithms are in boldface. According to the results, SQEOABC ranks the first in 23 out of 25 datasets (92%), which outperforms the compared algorithms. For comparison, EESHHO achieves the first rank in 3 datasets (12%), TLSMA in 2 datasets (8%), GWO, HHO and SMA in 1 dataset (4%). Based on the Friedman mean rank and final rank, SQEOABC obtains the first rank with

Freidman mean rank value 1.08, followed by GWO (3.6), LSHADE-cnEpSin (3.6), EESHHO (4.16), HHO (5.32), and other compared algorithms. Obviously, SQEOABC outperforms the compared metaheuristic algorithms in terms of the fitness values.

Table 4 also gives the statistical results of accuracy values, including mean, STD and rank values. From the results, SQEOABC achieves the best in 24 out of 25 datasets (96%), while EESHHO ranks the first in 4 datasets (4%), HHO and LSHADE-cnEpSin in 3 datasets (12%), GWO in 2 datasets (8%). It should be noted that most compared algorithms achieve the best accuracy solutions in the UJIIndoorLoc dataset, which is consistent to the results of [3]. This dataset has 523 attributes, where 520 attributes are equal to the same values (100) except for a few outliers, so most metaheuristic algorithms are easy to obtain the optimal features. The results of Friedman mean rank and final rank values indicate that SQEOABC obtains the first rank with Freidman mean rank value 1.16, followed by LSHADE-cnEpSin (3.16), GWO (3.68), EESHHO (3.88), HHO (5.0), and other compared algorithms. Thus, SQEOABC can provide good performance in the classification of accuracy, outperforming the compared metaheuristic algorithms.

Table 3 The fitness values evaluated by SQEOABC and compared algorithms

No.	Datasets	Algo	PSO	CMAES	GWO	HHO	SSA	SMA	WSO	TLSMA	DDSRPSO	EESHHO	LSHADE-cnEpSin	SQEOABC
1	BreastCancer	Mean	0.03239	0.03192	0.03075	0.02986	0.03268	0.03942	0.03124	0.03501	0.03161	0.03057	0.03061	**0.02956**
		STD	0.00366	0.00328	0.00169	0.00050	0.00293	0.00568	0.00223	0.00413	0.00222	0.00144	0.00153	**0.00031**
		Rank	9	8	5	2	10	12	6	11	7	3	4	**1**
2	BreastEW	Mean	0.07116	0.07099	0.06763	0.06613	0.07144	0.07637	0.07008	0.06977	0.06969	0.06869	0.06786	**0.06387**
		STD	0.00272	0.00644	0.00336	0.00379	0.00276	0.00359	0.00257	0.00344	0.00227	0.00295	0.00299	**0.00351**
		Rank	10	9	3	2	11	12	8	7	6	5	4	**1**
3	CongressEW	Mean	0.04199	0.03934	0.03824	0.03716	0.04422	0.04566	0.03965	0.04145	0.03906	0.03857	0.03773	**0.03478**
		STD	0.00506	0.00426	0.00275	0.00341	0.00389	0.00186	0.00412	0.00377	0.00458	0.00311	0.00391	**0.00262**
		Rank	10	7	4	2	11	12	8	9	6	5	3	**1**
4	Exactly	Mean	0.05525	0.05143	0.02475	0.07633	0.06715	0.27138	0.02402	0.16720	0.02831	**0.00462**	0.00533	**0.00462**
		STD	0.10170	0.10411	0.07663	0.12960	0.08713	0.05409	0.05569	0.14278	0.04938	**1.76E-18**	0.00271	**1.76E-18**
		Rank	7	6	4	9	8	11	3	10	5	**1**	2	**1**
5	Exactly2	Mean	0.27986	0.28270	0.27258	**0.23441**	0.27905	**0.23441**	0.27761	**0.23441**	0.27868	**0.23441**	0.27547	0.25691
		STD	0.01516	0.01451	0.01619	**2.82E-17**	0.01756	**2.82E-17**	0.01309	**2.82E-17**	0.00572	**2.82E-17**	0.01195	0.02137
		Rank	8	9	3	**1**	7	**1**	5	**1**	6	**1**	4	2
6	HeartEW	Mean	0.20961	0.20934	0.20384	0.20368	0.21753	0.23485	0.20773	0.22399	0.20993	0.20764	0.20451	**0.19521**
		STD	0.01025	0.00840	0.00953	0.00698	0.01773	0.00793	0.00824	0.01327	0.00931	0.00782	0.00773	**0.00404**
		Rank	8	7	3	2	10	12	6	11	9	5	4	**1**
7	Ionosphere	Mean	0.09882	0.08848	0.06978	0.08496	0.11781	0.11722	0.10186	0.07598	0.09917	0.07600	0.08407	**0.05577**
		STD	0.01159	0.01518	0.01206	0.01059	0.00824	0.01376	0.01224	0.01554	0.00948	0.01560	0.01367	**0.00695**
		Rank	8	7	2	6	12	11	10	3	9	4	5	**1**
8	KrVsKpEW	Mean	0.03837	0.02663	0.03149	0.03288	0.04721	0.08745	0.03702	0.03390	0.02993	0.02717	0.02679	**0.02265**
		STD	0.01048	0.00529	0.00930	0.00588	0.00983	0.02045	0.00762	0.00822	0.00544	0.00639	0.00308	**0.00068**
		Rank	10	2	6	7	11	12	9	8	5	4	3	**1**
9	Lymphography	Mean	0.15034	0.14999	0.13241	0.13873	0.14986	0.18545	0.13786	0.14754	0.13750	0.14091	0.12948	**0.11816**

		STD	0.01835	0.01853	0.01682	0.01812	0.01954	0.01890	0.01234	0.01804	0.01880	0.01790	0.01496	**0.00721**
		Rank	11	10	3	6	9	12	5	8	4	7	2	**1**
10	M-of-n	Mean	0.01986	0.00507	**0.00462**	0.00464	0.03704	0.13137	0.00735	0.00556	0.00817	**0.00462**	0.00467	**0.00462**
		STD	0.04118	0.00209	**1.76E-18**	0.00014	0.04621	0.04530	0.00671	0.00461	0.01306	**1.76E-18**	0.00020	**1.76E-18**
		Rank	8	4	**1**	2	9	10	6	5	7	**1**	3	**1**
11	PenglungEW	Mean	0.09530	0.04432	0.01625	0.07320	0.10712	0.12509	0.09529	0.03549	0.08708	0.03733	0.06930	**0.00620**
		STD	0.02395	0.02773	0.01354	0.02132	0.01226	0.01780	0.01496	0.01877	0.01956	0.02384	0.02065	**0.01085**
		Rank	10	5	2	7	11	12	9	3	8	4	6	**1**
12	Sonar	Mean	0.13476	0.11636	0.10602	0.13532	0.15968	0.19645	0.13602	0.11695	0.12974	0.12416	0.11445	**0.09543**
		STD	0.01697	0.01430	0.01686	0.01815	0.02117	0.01373	0.01581	0.01776	0.01553	0.01669	0.01487	**0.01376**
		Rank	8	4	2	9	11	12	10	5	7	6	3	**1**
13	SpectEW	Mean	0.14551	0.14069	0.13705	0.13995	0.15842	0.17591	0.14216	0.15030	0.14151	0.13878	0.13356	**0.12820**
		STD	0.01247	0.01309	0.00788	0.00697	0.00978	0.01070	0.00968	0.01673	0.01206	0.00870	0.00692	**0.00320**
		Rank	9	6	3	5	11	12	8	10	7	4	2	**1**
14	Tic-tac-toe	Mean	0.22693	0.22435	0.22416	0.22292	0.23381	0.24935	0.22419	0.22514	0.22479	0.22548	0.22262	**0.22161**
		STD	0.00785	0.00329	0.00558	0.00152	0.01284	0.01110	0.00404	0.00707	0.00779	0.00548	0.00145	**5.65E-17**
		Rank	10	6	4	3	11	12	5	8	7	9	2	**1**
15	Vote	Mean	0.03174	0.03222	0.02817	0.02724	0.03377	0.03682	0.03096	0.03173	0.03094	0.02920	0.02785	**0.02298**
		STD	0.00494	0.00384	0.00432	0.00467	0.00617	0.00356	0.00462	0.00497	0.00432	0.00454	0.00463	**0.00208**
		Rank	9	10	4	2	11	12	7	8	6	5	3	**1**
16	WaveformEW	Mean	0.21615	0.20223	0.20283	0.21685	0.22812	0.25015	0.21612	0.21608	0.21149	0.20945	0.20524	**0.19647**
		STD	0.00732	0.00469	0.00560	0.00654	0.00882	0.00399	0.00721	0.00686	0.00654	0.00696	0.00521	**0.00176**
		Rank	9	2	3	10	11	12	8	7	6	5	4	**1**
17	Wine	Mean	0.03288	0.03389	0.03109	0.02883	0.03929	0.06926	0.03180	0.04063	0.03258	0.02888	0.03016	**0.02686**
		STD	0.01055	0.01133	0.00493	0.00479	0.00903	0.01763	0.00702	0.01196	0.00771	0.00563	0.00607	**1.41E-17**
		Rank	8	9	5	2	10	12	6	11	7	3	4	**1**
18	Zoo	Mean	0.04752	0.04654	0.04406	0.04301	0.04478	0.06008	0.04320	0.04366	0.04389	0.04295	0.04287	**0.04257**
		STD	0.00804	0.00895	0.00480	0.00052	0.00472	0.01526	0.00046	0.00326	0.00335	0.00031	0.00036	**3.53E-17**

		Rank	11	10	8	4	9	12	5	6	7	3	2	**1**
19	CNAE	Mean	0.15513	0.09969	0.14405	0.16052	0.21264	0.18286	0.18234	0.14052	0.12785	0.11270	0.12059	**0.08319**
		STD	0.01838	0.01688	0.01595	0.01377	0.02517	**5.65E-17**	0.02607	0.01333	0.01332	0.01339	0.01782	0.00714
		Rank	8	2	7	9	12	11	10	6	5	3	4	**1**
20	Connectionist Bench Data	Mean	0.09947	0.07440	0.06905	0.09930	0.12499	0.15363	0.09904	0.08566	0.09567	0.08998	0.07195	**0.05305**
		STD	0.02288	0.01353	0.01311	0.01534	0.01480	0.01856	0.01792	0.01782	0.01947	0.01918	0.01626	**0.01062**
		Rank	10	4	2	9	11	12	8	5	7	6	3	**1**
21	Lung Cancer	Mean	0.15868	0.13384	0.10633	0.10993	0.19894	0.23795	0.15045	0.18365	0.14662	0.11503	0.12223	**0.05524**
		STD	0.06231	0.04535	0.05822	0.04929	0.05320	0.05503	0.04965	0.05509	0.05577	0.06213	0.04915	**0.04496**
		Rank	9	6	2	3	11	12	8	10	7	4	5	**1**
22	Optical Recognition of Handwritten	Mean	0.03451	0.02514	0.02931	0.03295	0.04462	0.04111	0.03514	0.03041	0.03071	0.02634	0.02543	**0.02123**
		STD	0.00586	0.00327	0.00586	0.00352	0.00823	**7.06E-18**	0.00509	0.00307	0.00330	0.00391	0.00396	0.00217
		Rank	9	2	5	8	12	11	10	6	7	4	3	**1**
23	QSAR biodegradation	Mean	0.15062	0.14184	0.13978	0.15140	0.16505	0.18651	0.15243	0.15001	0.14716	0.14304	0.14161	**0.13157**
		STD	0.00761	0.00848	0.00841	0.00690	0.01014	0.01023	0.00758	0.00916	0.00782	0.01015	0.00652	**0.00616**
		Rank	8	4	2	9	11	12	10	7	6	5	3	**1**
24	SPECT Heart	Mean	0.05734	0.04699	0.03009	0.06630	0.10180	0.15936	0.07315	0.05230	0.07031	0.04319	0.04157	**0.01474**
		STD	0.02598	0.02633	0.01936	0.02362	0.03051	0.03005	0.02179	0.02379	0.03006	0.02472	0.01932	**0.01695**
		Rank	7	5	2	8	11	12	10	6	9	4	3	**1**
25	UJIIndoorLoc	Mean	0.00364	0.00260	0.00055	0.00092	0.00447	0.00048	0.00431	**0.00002**	0.00338	0.00021	0.00342	0.00005
		STD	0.00025	0.00022	0.00016	0.00104	0.00014	0.00141	0.00018	**3.45E-21**	0.00046	0.00051	0.00016	0.00008
		Rank	10	7	5	6	12	4	11	**1**	8	3	9	2
Friedman mean rank			8.96	6.04	3.6	5.32	10.52	11	7.64	6.88	6.72	4.16	3.6	**1.08**
Final rank			9	5	2	4	10	11	8	7	6	3	2	**1**

Table 4 The accuracy values evaluated by SQEOABC and state-of-the-art algorithms

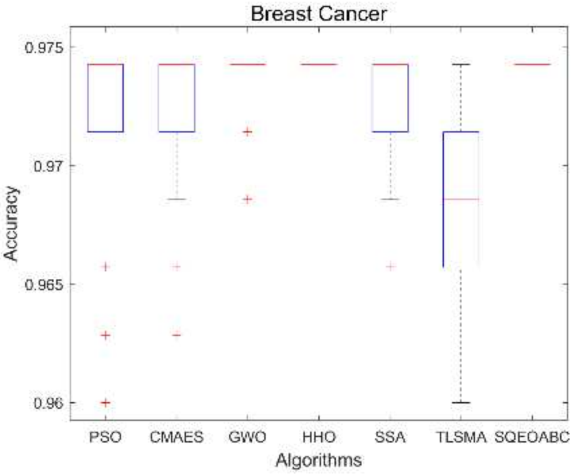
No.	Datasets	Algo	PSO	CMAES	GWO	HHO	SSA	SMA	WSO	TLSMA	DDSRPSO	EESHHO	LSHADE-cnEpSin	SQEOABC
1	BreastCancer	Mean	0.97210	0.97257	0.97362	0.97429	0.97238	0.96486	0.96762	0.96848	0.97343	0.97390	0.97400	**0.97429**
		STD	0.00349	0.00341	0.00144	0.00000	0.00274	0.00601	0.00413	0.00400	0.00153	0.00124	0.00087	**3.39E-16**
		Rank	9	7	5	2	8	12	11	10	6	4	3	**1**
2	BreastEW	Mean	0.93087	0.93063	0.93357	0.93627	0.93181	0.92653	0.93122	0.93110	0.93239	0.93239	0.93357	**0.93721**
		STD	0.00285	0.00635	0.00343	0.00396	0.00285	0.00411	0.00201	0.00366	0.00234	0.00312	0.00303	**0.00359**
		Rank	10	11	4	2	7	12	8	9	5	6	3	**1**
3	CongressEW	Mean	0.96083	0.96298	0.96375	0.96528	0.95868	0.95499	0.95745	0.95975	0.96375	0.96390	0.96467	**0.96790**
		STD	0.00510	0.00506	0.00337	0.00396	0.00428	0.00262	0.00537	0.00452	0.00465	0.00344	0.00442	**0.00331**
		Rank	8	7	6	2	10	12	11	9	5	4	3	**1**
4	Exactly	Mean	0.94880	0.95253	0.97953	0.92687	0.93753	0.72987	0.90753	0.83453	0.97667	1.00000	0.99933	**1.00000**
		STD	0.10395	0.10611	0.07790	0.13239	0.08851	0.05708	0.09907	0.14590	0.04910	0.00000	0.00254	**0.00000**
		Rank	7	6	4	9	8	12	10	11	5	2	3	**1**
5	Exactly2	Mean	0.72387	0.72040	0.73013	0.76400	0.72347	0.76400	0.71367	0.76400	0.72593	**0.76400**	0.72807	0.74467
		STD	0.01448	0.01413	0.01437	0.00000	0.01654	0.00000	0.01734	0.00000	0.00564	**0.00000**	0.01072	0.01840
		Rank	9	11	6	4	10	3	12	2	8	**1**	7	5
6	HeartEW	Mean	0.79185	0.79210	0.79728	0.79753	0.78395	0.76420	0.77210	0.77605	0.79160	0.79358	0.79679	**0.80593**
		STD	0.01017	0.00824	0.00982	0.00710	0.01784	0.00872	0.01758	0.01438	0.00927	0.00796	0.00770	**0.00408**
		Rank	7	6	3	2	9	12	11	10	8	5	4	**1**
7	Ionosphere	Mean	0.90379	0.91383	0.93125	0.91610	0.88504	0.88277	0.88504	0.92443	0.90360	0.92462	0.91780	**0.94508**
		STD	0.01126	0.01455	0.01188	0.01083	0.00799	0.01388	0.01209	0.01580	0.00889	0.01572	0.01341	**0.00689**
		Rank	8	7	2	6	11	12	10	4	9	3	5	**1**
8	KrVsKpEW	Mean	0.96648	0.97812	0.97174	0.97222	0.95788	0.91756	0.96195	0.97111	0.97572	0.97797	0.97837	**0.98166**
		STD	0.01084	0.00561	0.00944	0.00605	0.01008	0.01974	0.01035	0.00796	0.00545	0.00647	0.00295	**0.00112**
		Rank	9	3	7	6	11	12	10	8	5	4	2	**1**
9	Lymphography	Mean	0.85270	0.85315	0.87027	0.86396	0.85315	0.81622	0.83874	0.85450	0.86577	0.86171	0.87342	**0.88468**

		STD	0.01856	0.01869	0.01760	0.01876	0.01999	0.01931	0.01699	0.01901	0.01909	0.01833	0.01526	**0.00772**
		Rank	10	9	3	5	8	12	11	7	4	6	2	**1**
10	M-of-n	Mean	0.98493	0.99967	**1.00000**	**1.00000**	0.96833	0.87473	0.96147	0.99920	0.99667	**1.00000**	**1.00000**	**1.00000**
		STD	0.04131	0.00183	**0.00000**	**0.00000**	0.04648	0.04535	0.05575	0.00438	0.01273	**0.00000**	**0.00000**	**0.00000**
		Rank	5	2	**1**	**1**	6	8	7	3	4	**1**	**1**	**1**
11	PenglungEW	Mean	0.90721	0.95766	0.98468	0.92793	0.89640	0.87477	0.89910	0.96486	0.91532	0.96306	0.93333	**0.99459**
		STD	0.02426	0.02811	0.01362	0.02168	0.01246	0.01807	0.01576	0.01898	0.01974	0.02405	0.02098	**0.01100**
		Rank	9	5	2	7	11	12	10	3	8	4	6	**1**
12	Sonar	Mean	0.86795	0.88590	0.89519	0.86699	0.84327	0.80513	0.84840	0.88397	0.87308	0.87692	0.88814	**0.90641**
		STD	0.01693	0.01445	0.01700	0.01822	0.02145	0.01404	0.01725	0.01802	0.01565	0.01683	0.01483	**0.01358**
		Rank	8	4	2	9	11	12	10	5	7	6	3	**1**
13	SpectEW	Mean	0.85697	0.86219	0.86468	0.86219	0.84453	0.82711	0.84353	0.85149	0.86119	0.86318	0.86891	**0.87413**
		STD	0.01256	0.01296	0.00825	0.00726	0.01020	0.01110	0.01262	0.01701	0.01218	0.00905	0.00698	**0.00379**
		Rank	8	5	3	6	10	12	11	9	7	4	2	**1**
14	Tic-tac-toe	Mean	0.77711	0.77933	0.77975	0.78107	0.77056	0.75651	0.75581	0.77905	0.77933	0.77815	0.78149	**0.78288**
		STD	0.00761	0.00343	0.00572	0.00210	0.01274	0.00988	0.01795	0.00707	0.00804	0.00577	0.00200	**0.00000**
		Rank	9	6	4	3	10	11	12	7	5	8	2	**1**
15	Vote	Mean	0.97089	0.97044	0.97422	0.97511	0.96911	0.96422	0.96244	0.96978	0.97178	0.97311	0.97467	**0.97933**
		STD	0.00510	0.00417	0.00454	0.00493	0.00567	0.00446	0.00689	0.00546	0.00453	0.00479	0.00476	**0.00203**
		Rank	7	8	4	2	10	11	12	9	6	5	3	**1**
16	WaveformEW	Mean	0.78719	0.80109	0.79927	0.78739	0.77544	0.75627	0.77829	0.78864	0.79301	0.79480	0.79808	**0.80665**
		STD	0.00750	0.00457	0.00574	0.00645	0.00916	0.00307	0.00837	0.00614	0.00593	0.00646	0.00527	**0.00163**
		Rank	9	2	3	8	11	12	10	7	6	5	4	**1**
17	Wine	Mean	0.97116	0.97004	0.97266	0.97528	0.96479	0.93333	0.95655	0.96217	0.97191	0.97528	0.97378	**0.97753**
		STD	0.01091	0.01193	0.00566	0.00544	0.00921	0.01793	0.01836	0.01269	0.00821	0.00619	0.00681	**3.39E-16**
		Rank	7	8	5	3	9	12	11	10	6	2	4	**1**
18	Zoo	Mean	0.95621	0.95752	0.95948	**0.96078**	0.95948	0.94510	0.95686	0.96013	0.96013	**0.96078**	**0.96078**	**0.96078**
		STD	8.43E-03	0.00904	0.00497	**0**	0.00497	0.01579	0.01080	0.00358	0.00358	**0**	**0**	**0**

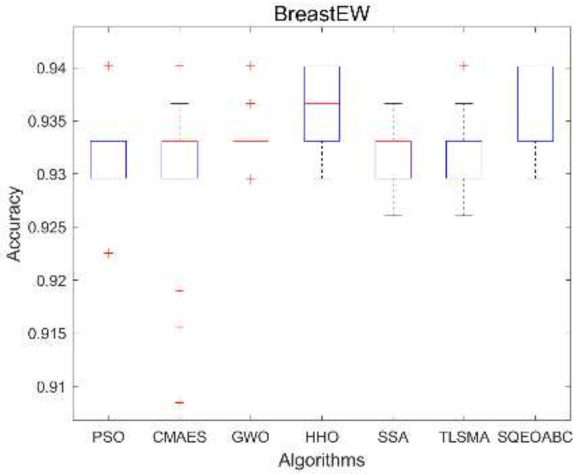
		Rank	8	6	5	1	4	9	7	3	2	1	1	1
19	CNAE	Mean	0.84797	0.90406	0.85670	0.84550	0.79030	0.82540	0.80423	0.86631	0.87910	0.89330	0.88298	**0.92205**
		STD	0.01859	0.01712	0.01613	0.01433	0.02540	0.00000	0.03183	0.01314	0.01307	0.01308	0.01806	**0.00694**
		Rank	8	2	7	9	12	10	11	6	5	3	4	**1**
20	Connectionist Bench Data	Mean	0.90320	0.92785	0.93242	0.90274	0.87808	0.84749	0.88584	0.91553	0.90731	0.91096	0.93059	**0.94886**
		STD	0.02302	0.01390	0.01343	0.01541	0.01498	0.01825	0.02138	0.01802	0.01927	0.01929	0.01646	**0.01075**
		Rank	8	4	2	9	11	12	10	5	7	6	3	**1**
21	Lung Cancer	Mean	0.84242	0.86667	0.89394	0.89091	0.80303	0.76061	0.81212	0.81515	0.85455	0.88485	0.87879	**0.94545**
		STD	0.06286	0.04613	0.05888	0.05008	0.05383	0.05590	0.04735	0.05590	0.05650	0.06286	0.04970	**0.04530**
		Rank	8	6	2	3	11	12	10	9	7	4	5	**1**
22	Optical Recognition of Handwritten	Mean	0.97029	0.97933	0.97457	0.97390	0.96048	0.96857	0.96390	0.97667	0.97638	0.97933	0.97952	**0.98381**
		STD	0.00603	0.00333	0.00602	0.00342	0.00843	0.00000	0.00694	0.00271	0.00290	0.00365	0.00405	**0.00217**
		Rank	9	4	7	8	12	10	11	5	6	3	2	**1**
23	QSAR biodegradation	Mean	0.85216	0.86072	0.86189	0.85108	0.83811	0.81568	0.84405	0.85144	0.85595	0.85856	0.86090	**0.87054**
		STD	0.00752	0.00839	0.00844	0.00686	0.01012	0.01049	0.00934	0.00954	0.00775	0.01037	0.00661	**0.00604**
		Rank	7	4	2	9	11	12	10	8	6	5	3	**1**
24	SPECT Heart	Mean	0.94524	0.95476	0.97143	0.93571	0.90119	0.84048	0.90595	0.94881	0.93214	0.95833	0.96071	**0.98690**
		STD	0.02608	0.02642	0.01967	0.02373	0.03066	0.03073	0.02566	0.02425	0.03017	0.02496	0.01956	**0.01750**
		Rank	7	5	2	8	11	12	10	6	9	4	3	**1**
25	UJIIndoorLoc	Mean	**1.00000**	**1.00000**	**1.00000**	**1.00000**	**1.00000**	0.99990	**1.00000**	**1.00000**	**1.00000**	**1.00000**	**1.00000**	**1.00000**
		STD	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	0.00056	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**	**0.00000**
		Rank	**1**	**1**	**1**	**1**	**1**	2	**1**	**1**	**1**	**1**	**1**	**1**
Friedman mean rank			7.8	5.56	3.68	5	9.32	10.72	9.88	6.64	5.88	3.88	3.16	**1.16**
Final rank			9	6	3	5	10	12	11	8	7	4	2	**1**

To investigate the data distribution of quartiles from the algorithms, the boxplot analysis is implemented. When the median value is larger or the spread value is smaller, the metaheuristic algorithm has better performance. Fig. 2 summarizes the boxplots of fitness accuracy for SQEOABC and its competitors, including PSO, CMAES, GWO, HHO, SSA and TLSMA. X-axis represents the algorithms, and y-axis denotes the accuracy values. It is observed that SQEOABC has higher boxplots than that of the competitors with the highest median in the tested datasets, such as Breast Cancer, BreastEW, Exactly, Ionosphere, KrVsKpEW, Lymphography, SpectEW, WaveformEW, and so on. In comparison, the compared algorithms also provide competitive results in some datasets, such as Breast Cancer (GWO, HHO), Exactly (GWO), Zoo (PSO, CMAES, GWO, HHO, SSA, TLSMA). In all, SQEOABC outperforms the compared algorithms in the majority of the datasets.

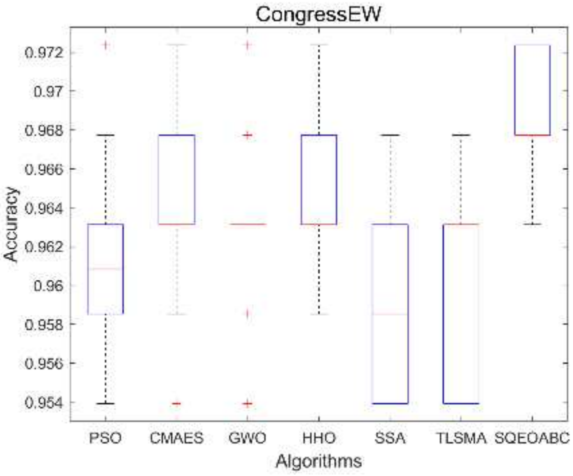
Fig. 3 summarizes the convergence curves of accuracy values obtained by SQEOABC and state-of-the-art algorithms. X-axis and Y-axis denote the values of accuracy and iteration, respectively. Based on the convergence behaviors, SQEOABC has fast convergence in most of the datasets such as BreastEW, CongressEW, HeartEW, Ionosphere, Lymphography, SpectEW, etc., which outperforms competitors including PSO, CMAES, GWO, HHO, SSA, SMA, WSO, TLSMA, DDSRPSO, EESHHO, and LSHADE-cnEpSin. In comparison, several metaheuristic algorithms also provide competitive convergence in some datasets, such as Breast Cancer, KrVsKpEW, but the compared algorithms cannot obtain better convergence accuracy curves in the majority of datasets. The convergence behaviors of accuracy demonstrate the effectiveness of SQEOABC.



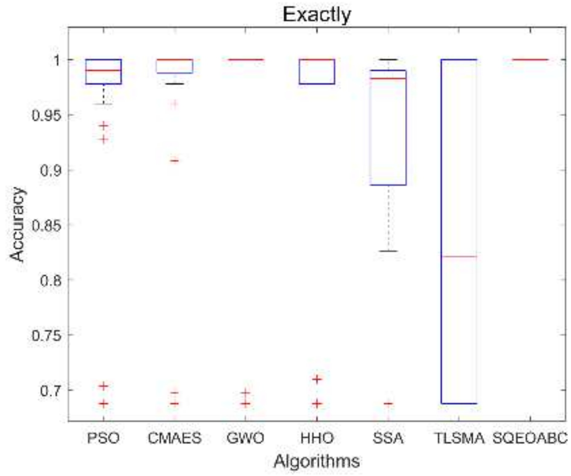
(a) Breast Cancer



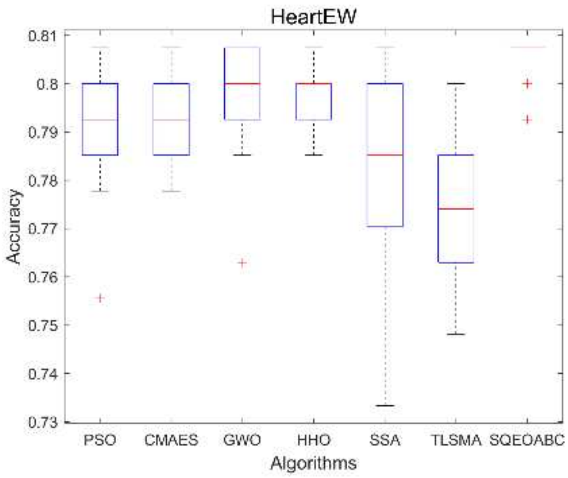
(b) BreastEW



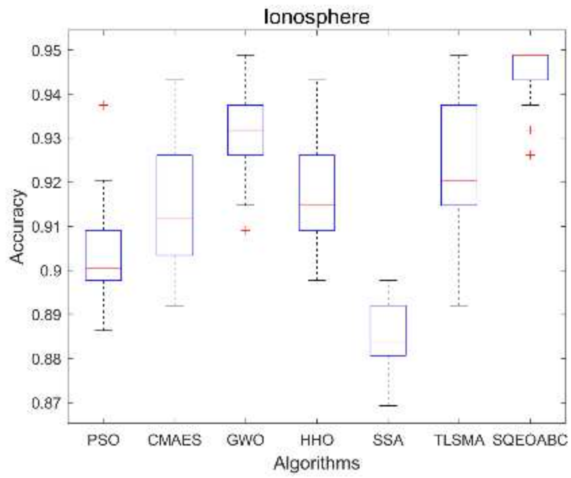
(c) CongressEW



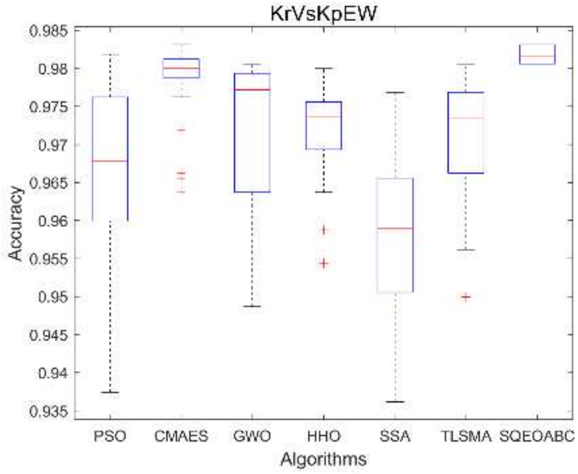
(d) Exactly



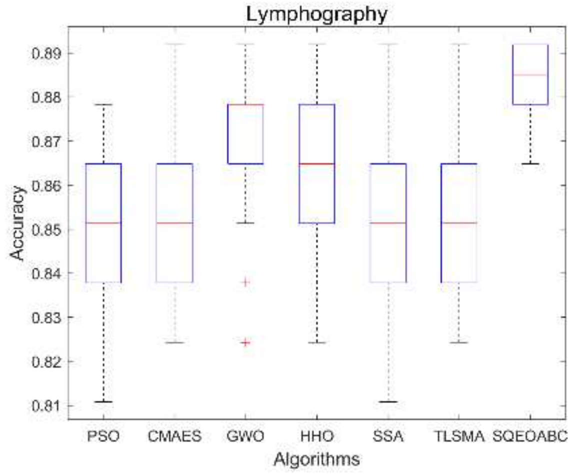
(e) HeartEW



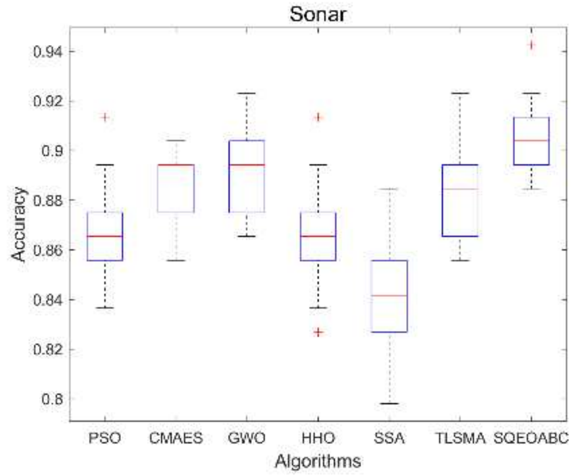
(f) Ionosphere



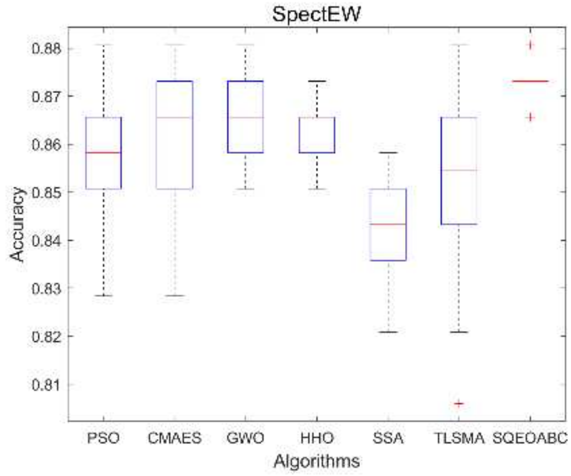
(g) KrVsKpEW



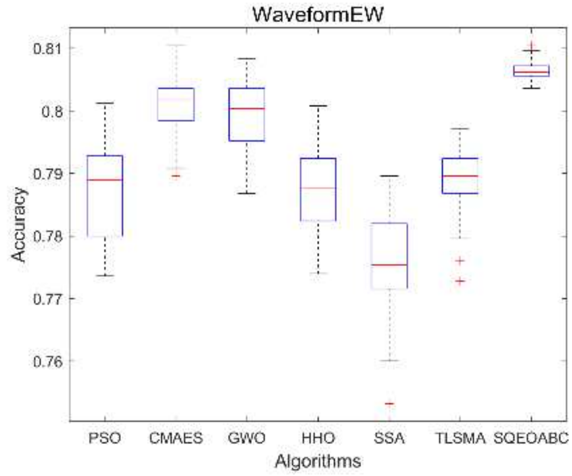
(h) Lymphography



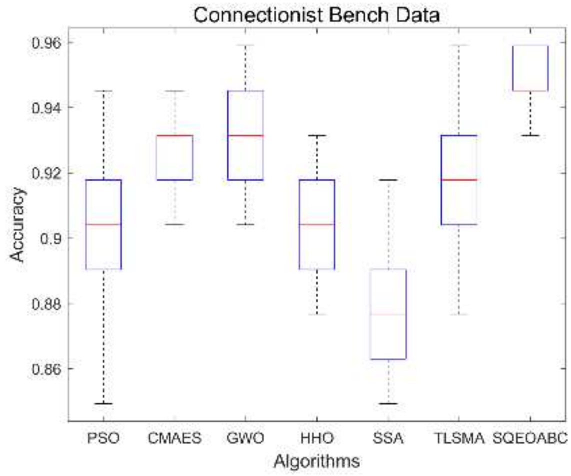
(i) Sonar



(j) SpectEW

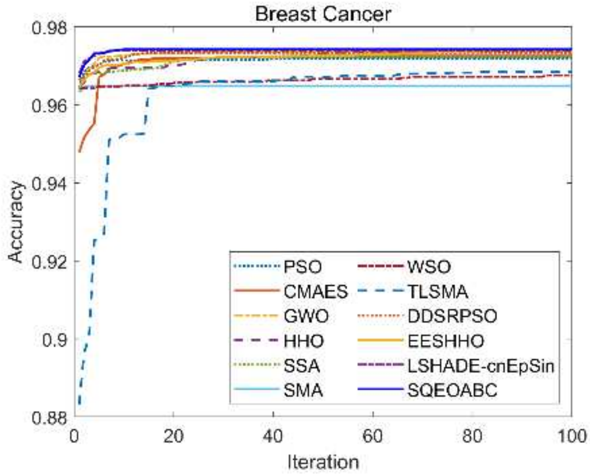


(k) WaveformEW

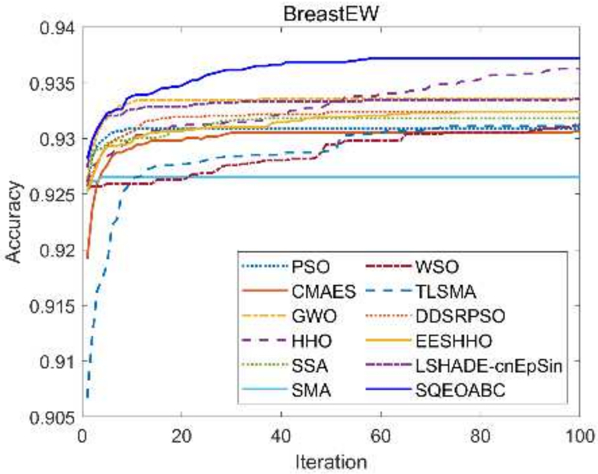


(l) Connectionist Bench Data

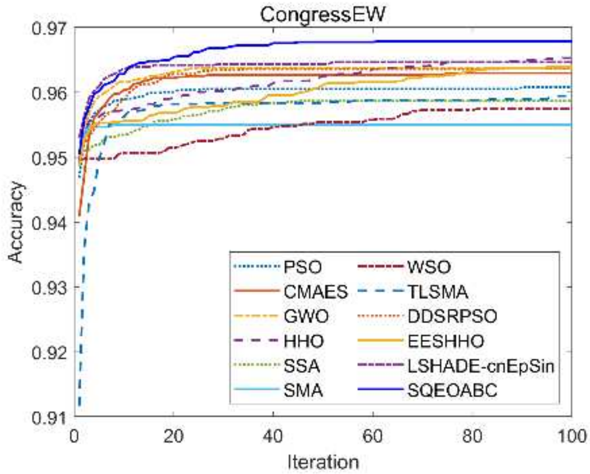
Fig. 2 Boxplots by SQEOABC and state-of-the-art algorithms



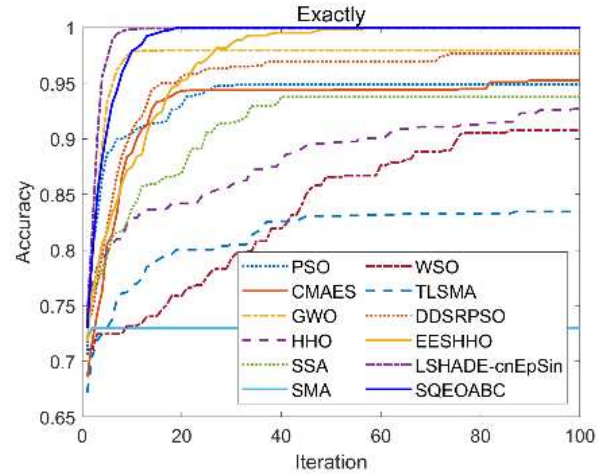
(a) Breast Cancer



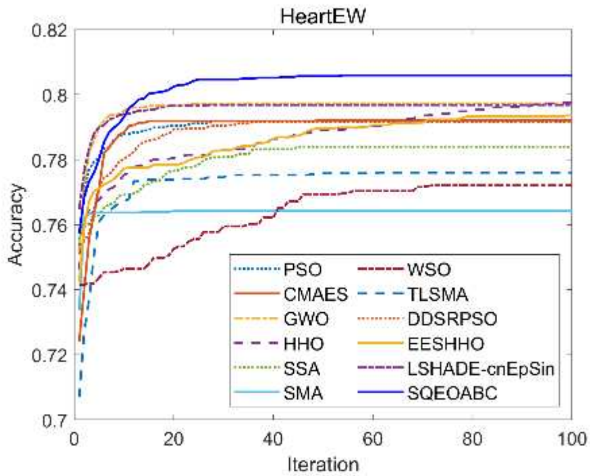
(b) BreastEW



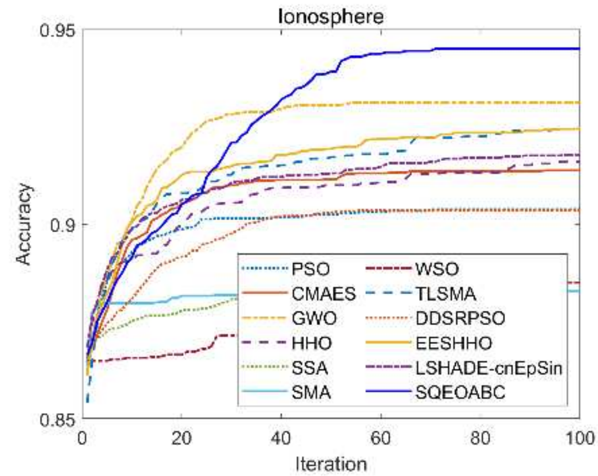
(c) CongressEW



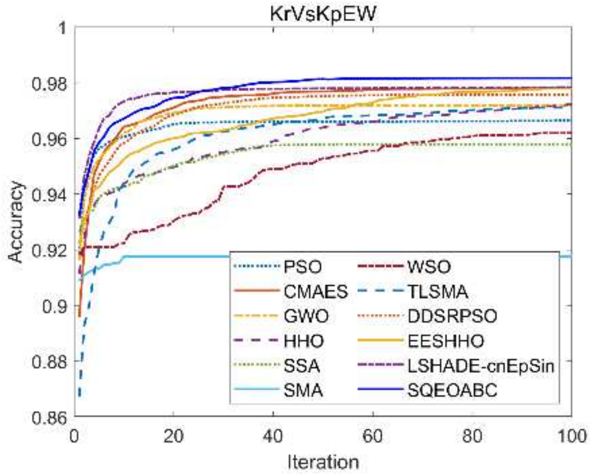
(d) Exactly



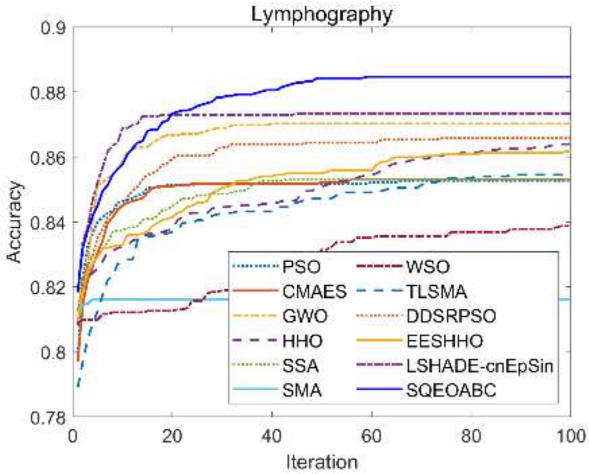
(e) HeartEW



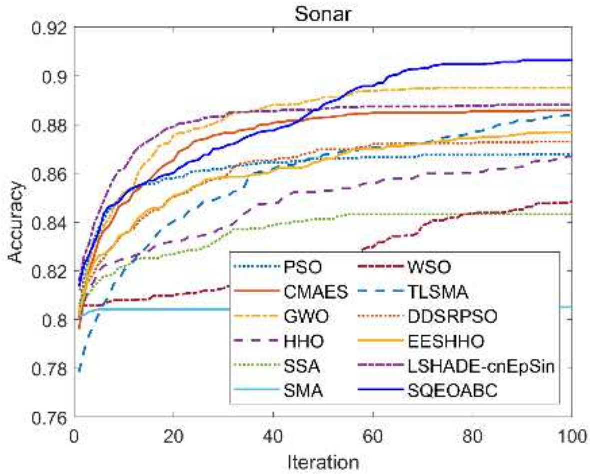
(f) Ionosphere



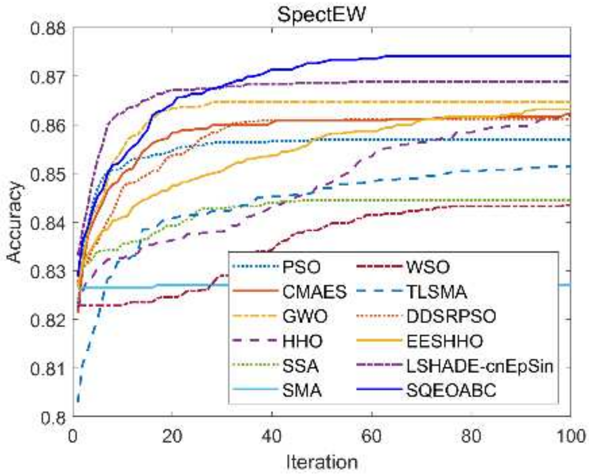
(g) KrVsKpEW



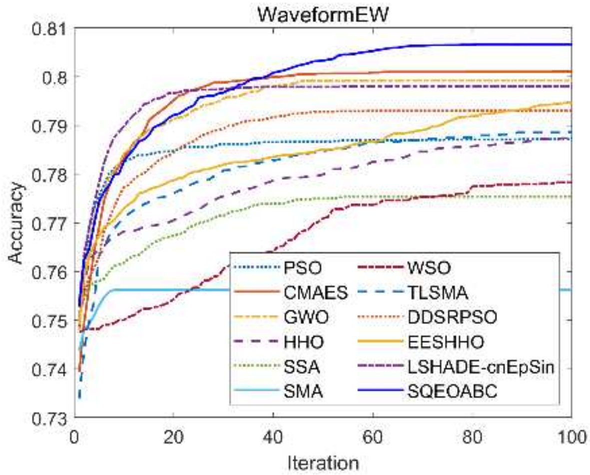
(h) Lymphography



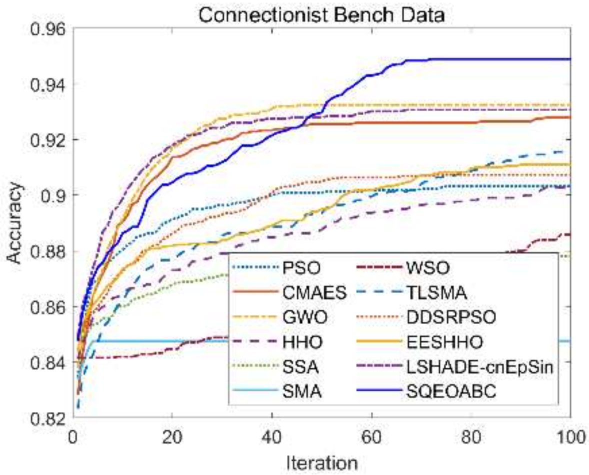
(i) Sonar



(j) SpectEW



(k) WaveformEW



(l) Connectionist Bench Data

Fig. 3 Convergence curves of accuracy values by SQEOABC and state-of-the-art algorithms

5.3 Comparison of SQEOABC with variants of EO

To further investigate the influence of different operators in the proposed SQEOABC algorithm, several variants of EO are also compared in the FS problems, including SEO (self-adaptive EO), QEO (quantum EO), EOABC (EO with artificial bee colony), SQEO (Self-adaptive quantum EO). Besides, an advanced EO algorithm called comprehensive learning Harris hawks-equilibrium optimizer (CLHHEO) [13] is also considered for comparison. The algorithmic parameters of EO and variants are the same as Table 2.

Table 5 summarizes the fitness values of the standard EO and variants of EO. Based on the results, SQEOABC achieves the first rank in 23 out of 25 datasets (92%), except for the datasets of Wine and UJIIndoorLoc. For comparison, the basic EO and its variants also perform well on some datasets, where EO ranks the first in 3 datasets (12%), SEO in 2 datasets (8%), CLHHEO in 4 datasets (16%), EOABC and SQEO in 6 datasets (24%), QEO in 7 datasets (28%). However, the variants of EO still cannot provide better results than the proposed SQEOABC algorithm. The effects of different operators on the modified EO are also discussed. The QEO has better or equal fitness values than that of EO in 18 out of 25 datasets (72%), illustrating the effectiveness of the quantum operator. Besides, EOABC has better or equal fitness values than EO in 22 out of 25 datasets (88%), SQEO outperforms or equals to EO in 24 out of 25 datasets (96%), demonstrating the effectiveness of operators of self-adaptive mechanism and artificial bee colony. It should be noted that, the self-adaptive EO cannot achieve better solutions than EO, but the SQEO has obviously advantages than EO, indicating that the ensemble strategies are more useful in the modification of the basic EO algorithm. The numbers of rank are also provided in Table 5, illustrating that SQEOABC has the best Friedman mean rank (1.2), which is better than that of SQEO (2.44), EOABC (2.56), QEO (3.48), CLHHEO (4.4), EO (4.48) and SEO (5.04). The above experimental results indicate that SQEOABC outperforms the variants of EO, and the ensemble strategies are more effective than the single operator into EO in solving FS problems.

The statistical results of accuracy values evaluated by SQEOABC and variants of EO are provided in Table 6, showing that SQEOABC still has very good performance and ranks the first in 23 out of 25 datasets (92%). In comparison, EO, SEO, QEO, EOABC, SQEO, CLHHEO achieve the first rank in 4 (16%), 3 (12%), 8 (32%), 9 (36%), 7 (28%), 5 (20%) out of 25 datasets, respectively. It should be mentioned that, similarly to the fitness values, the ensemble strategies are effective on EO to obtain the accuracy of classification, where EOABC and SQEO has better or equal solutions than EO in 23 (92%) and 25 (100%) out of 25 datasets, respectively. Moreover, SQEOABC ranks the first with the Friedman mean rank (1.36), followed by EOABC (2.28), SQEO (2.28), QEO (3.48), CLHHEO (4.12), EO (4.36) and SEO (4.72). The results indicate that the three enhanced operators from self-adaptive mechanism, quantum theory, and artificial bee colony employed in EO can improve the accuracy of classification in FS problems.

Table 5 The fitness values evaluated by SQEOABC and EO variants

No.	Datasets	Algo	EO	SEO	QEO	EOABC	SQEO	CLHHEO	SQEOABC
1	BreastCancer	Mean	**0.03694**	**0.03694**	**0.03694**	**0.03694**	**0.03694**	0.03724	**0.03694**
		STD	**2.82E-17**	**2.82E-17**	**2.82E-17**	**2.82E-17**	**2.82E-17**	0.00079	**2.82E-17**
		Rank	**1**	**1**	**1**	**1**	**1**	2	**1**
2	BreastEW	Mean	0.05268	0.05375	**0.05047**	0.05049	0.05060	0.05101	**0.05047**
		STD	0.00277	0.00366	**1.41E-17**	0.00008	0.00064	0.00128	**1.41E-17**
		Rank	5	6	**1**	2	3	4	**1**
3	CongressEW	Mean	0.04263	0.04328	0.03752	0.03649	0.03824	0.03924	**0.03641**
		STD	0.00358	0.00302	0.00225	0.00122	0.00326	0.00314	**0.00128**
		Rank	6	7	3	2	4	5	**1**
4	Exactly	Mean	0.01283	0.07047	**0.00462**	**0.00462**	**0.00462**	**0.00462**	**0.00462**
		STD	0.04497	0.12213	**1.76E-18**	**1.76E-18**	**1.76E-18**	**1.76E-18**	**1.76E-18**
		Rank	2	3	**1**	**1**	**1**	**1**	**1**
5	Exactly2	Mean	**0.24233**	**0.24233**	**0.24233**	**0.24233**	**0.24233**	**0.24233**	**0.24233**
		STD	**1.13E-16**	**1.13E-16**	**1.13E-16**	**1.13E-16**	**1.13E-16**	**1.13E-16**	**1.13E-16**
		Rank	**1**	**1**	**1**	**1**	**1**	**1**	**1**
6	HeartEW	Mean	0.16950	0.17195	0.16650	0.16612	0.16662	0.16845	**0.16595**
		STD	0.00320	0.00744	0.00154	0.00092	0.00174	0.00277	**2.82E-17**
		Rank	6	7	3	2	4	5	**1**
7	Ionosphere	Mean	0.05802	0.05701	0.05598	0.05658	0.05553	0.06245	**0.05209**
		STD	0.00596	0.00778	0.00735	0.00566	0.00667	0.00779	**0.00498**
		Rank	6	5	3	4	2	7	**1**
8	KrVsKpEW	Mean	0.02838	0.02936	0.02467	0.02346	0.02315	0.02467	**0.02215**
		STD	0.00855	0.00796	0.00183	0.00104	0.00098	0.00167	**0.00068**
		Rank	6	7	5	3	2	4	**1**
9	Lymphography	Mean	0.13453	0.13424	0.11775	0.11707	0.11967	0.12719	**0.11562**
		STD	0.01527	0.01492	0.00804	0.00766	0.00934	0.01098	**0.00651**
		Rank	7	6	3	2	4	5	**1**
10	M-of-n	Mean	**0.00462**	0.00926	**0.00462**	**0.00462**	**0.00462**	**0.00462**	**0.00462**
		STD	**1.76E-18**	0.02545	**1.76E-18**	**1.76E-18**	**1.76E-18**	**1.76E-18**	**1.76E-18**
		Rank	**1**	2	**1**	**1**	**1**	**1**	**1**
11	PenglungEW	Mean	0.00792	0.00804	0.02038	0.01442	0.00710	0.00996	**0.00624**
		STD	0.01200	0.01202	0.01206	0.01351	0.01140	0.01286	**0.01087**
		Rank	3	4	7	6	2	5	**1**
12	Sonar	Mean	0.06832	0.07290	0.08185	0.05993	0.06100	0.07613	**0.04575**
		STD	0.01632	0.01301	0.01277	0.01411	0.01421	0.01526	**0.01340**
		Rank	4	5	7	2	3	6	**1**
13	SpectEW	Mean	0.15208	0.15445	0.14891	0.14807	0.15100	0.15275	**0.14564**

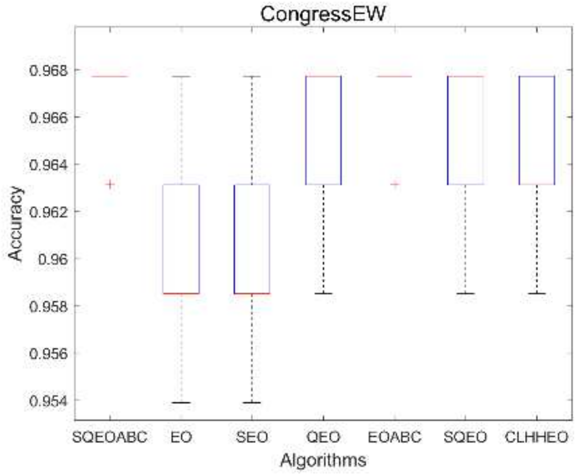
		STD	0.00563	0.01095	0.00730	0.00541	0.00603	0.00622	**0.00399**
		Rank	5	7	3	2	4	6	**1**
14	Tic-tac-toe	Mean	0.21631	0.21717	**0.21541**	**0.21541**	**0.21541**	**0.21541**	**0.21541**
		STD	0.00403	0.00604	**1.13E-16**	**1.13E-16**	**1.13E-16**	**1.13E-16**	**1.13E-16**
		Rank	2	3	**1**	**1**	**1**	**1**	**1**
15	Vote	Mean	0.05099	0.04995	0.04895	0.04957	0.04957	0.05025	**0.04850**
		STD	0.00471	0.00368	0.00195	0.00236	0.00285	0.00362	**0.00132**
		Rank	7	5	2	3	4	6	**1**
16	WaveformEW	Mean	0.19685	0.19936	0.20235	0.19359	0.19603	0.20286	**0.19055**
		STD	0.00428	0.00595	0.00352	0.00377	0.00404	0.00555	**0.00310**
		Rank	4	5	6	2	3	7	**1**
17	Wine	Mean	0.04162	0.04377	**0.03799**	0.03863	0.03801	0.04086	0.03828
		STD	0.00453	0.00450	**0.00000**	0.00243	0.00014	0.00442	0.00161
		Rank	6	7	**1**	4	2	5	3
18	Zoo	Mean	0.01590	0.01638	0.00502	**0.00500**	0.00801	0.01045	**0.00500**
		STD	0.01019	0.00877	0.00011	**1.76E-18**	0.00684	0.00847	**1.76E-18**
		Rank	5	6	2	**1**	3	4	**1**
19	CNAE	Mean	0.12538	0.14050	0.13326	0.11037	0.10120	0.11678	**0.07942**
		STD	0.01243	0.01205	0.00967	0.01276	0.01228	0.01595	**0.00666**
		Rank	5	7	6	3	2	4	**1**
20	Connectionist Bench Data	Mean	0.06079	0.06652	0.07565	0.05942	0.05533	0.07037	**0.04802**
		STD	0.01923	0.02328	0.01575	0.01500	0.01422	0.01607	**0.01293**
		Rank	4	5	7	3	2	6	**1**
21	Lung Cancer	Mean	0.12404	0.11220	0.07983	0.03532	0.05277	0.10336	**0.03167**
		STD	0.06855	0.07707	0.04517	0.04411	0.04519	0.05635	**0.04293**
		Rank	7	6	4	2	3	5	**1**
22	Optical Recognition of Handwritten	Mean	0.03228	0.03362	0.03573	0.02978	0.03013	0.03137	**0.02705**
		STD	0.00342	0.00403	0.00406	0.00352	0.00257	0.00356	**0.00256**
		Rank	5	6	7	2	3	4	**1**
23	QSAR biodegradation	Mean	0.14385	0.14338	0.14194	0.13557	0.13663	0.14140	**0.13290**
		STD	0.00730	0.00655	0.00604	0.00418	0.00509	0.00531	**0.00486**
		Rank	7	6	5	2	3	4	**1**
24	SPECT Heart	Mean	0.09184	0.09182	0.09750	0.09540	0.09047	0.10826	**0.07564**
		STD	0.03548	0.04105	0.03459	0.03280	0.03306	0.05285	**0.02685**
		Rank	4	3	6	5	2	7	**1**
25	UJIIndoorLoc	Mean	0.00003	0.00009	**0.00002**	0.00145	**0.00002**	0.00006	0.00005
		STD	0.00002	0.00008	**0.00000**	0.00031	**0.00000**	0.00020	0.00010
		Rank	2	5	**1**	6	**1**	4	3
Friedman mean rank			4.48	5.04	3.48	2.56	2.44	4.4	**1.2**
Final rank			6	7	4	3	2	5	**1**

Table 6 The accuracy values evaluated by SQEOABC and EO variants

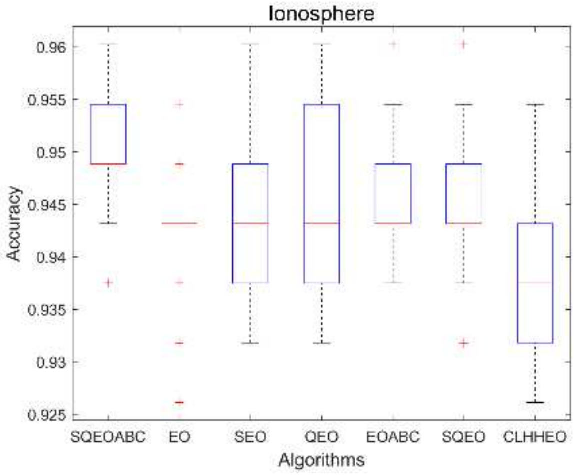
No.	Datasets	Algo	EO	SEO	QEO	EOABC	SQEO	CLHHEO	SQEOABC
1	BreastCancer	Mean	**0.96571**	**0.96571**	**0.96571**	**0.96571**	**0.96571**	**0.96571**	**0.96571**
		STD	**3.39E-16**	**3.39E-16**	**3.39E-16**	**3.39E-16**	**3.39E-16**	**3.39E-16**	**3.39E-16**
		Rank	**1**	**1**	**1**	**1**	**1**	**1**	**1**
2	BreastEW	Mean	0.94836	0.94730	**0.95070**	**0.95070**	0.95059	0.95023	**0.95070**
		STD	0.00297	0.00387	**3.39E-16**	**3.39E-16**	0.00064	0.00122	**3.39E-16**
		Rank	4	5	**1**	**1**	2	3	**1**
3	CongressEW	Mean	0.95991	0.95914	0.96621	**0.96743**	0.96528	0.96436	0.96728
		STD	0.00439	0.00378	0.00279	**0.00117**	0.00396	0.00362	0.00141
		Rank	6	7	3	**1**	4	5	2
4	Exactly	Mean	0.99173	0.93353	**1.00000**	**1.00000**	**1.00000**	**1.00000**	**1.00000**
		STD	0.04528	0.12330	**0**	**0**	**0**	**0**	**0**
		Rank	2	3	**1**	**1**	**1**	**1**	**1**
5	Exactly2	Mean	**0.75600**	**0.75600**	**0.75600**	**0.75600**	**0.75600**	**0.75600**	**0.75600**
		STD	**1.13E-16**	**1.13E-16**	**1.13E-16**	**1.13E-16**	**1.13E-16**	**1.13E-16**	**1.13E-16**
		Rank	**1**	**1**	**1**	**1**	**1**	**1**	**1**
6	HeartEW	Mean	0.83210	0.82963	0.83630	0.83679	0.83605	0.83358	**0.83704**
		STD	0.00405	0.00778	0.00226	0.00135	0.00256	0.00376	**5.65E-16**
		Rank	6	7	3	2	4	5	**1**
7	Ionosphere	Mean	0.94318	0.94413	0.94527	0.94545	0.94583	0.93864	**0.94962**
		STD	0.00615	0.00791	0.00753	0.00550	0.00696	0.00795	**0.00489**
		Rank	6	5	4	3	2	7	**1**
8	KrVsKpEW	Mean	0.97495	0.97422	0.98085	0.98073	0.98125	0.97970	**0.98290**
		STD	0.00889	0.00828	0.00248	0.00161	0.00178	0.00188	**0.00165**
		Rank	6	7	3	4	2	5	**1**
9	Lymphography	Mean	0.86757	0.86757	0.88514	0.88604	0.88333	0.87523	**0.88739**
		STD	0.01603	0.01563	0.00851	0.00768	0.00971	0.01160	**0.00648**
		Rank	7	6	3	2	4	5	**1**
10	M-of-n	Mean	**1**	0.99533	**1**	**1**	**1**	**1**	**1**
		STD	**0**	0.02556	**0**	**0**	**0**	**0**	**0**
		Rank	**1**	2	**1**	**1**	**1**	**1**	**1**
11	PenglungEW	Mean	0.99279	0.99279	0.98108	0.98739	0.99369	0.99099	**0.99459**
		STD	0.01216	0.01216	0.01260	0.01371	0.01163	0.01296	**0.01100**
		Rank	4	3	7	6	2	5	**1**
12	Sonar	Mean	0.93301	0.92853	0.92115	0.94295	0.94135	0.92564	**0.95673**
		STD	0.01665	0.01329	0.01273	0.01427	0.01458	0.01535	**0.01354**
		Rank	4	5	7	2	3	6	**1**
13	SpectEW	Mean	0.84876	0.84627	0.85199	0.85348	0.85000	0.84826	**0.85572**
		STD	0.00586	0.01102	0.00735	0.00536	0.00599	0.00630	**0.00408**
		Rank	5	7	3	2	4	6	**1**
14	Tic-tac-toe	Mean	0.78824	0.78733	**0.78914**	**0.78914**	**0.78914**	**0.78914**	**0.78914**
		STD	0.00424	0.00636	**0**	**0**	**0**	**0**	**0**

		Rank	2	3	**1**	**1**	**1**	**1**	**1**
15	Vote	Mean	0.95067	0.95178	0.95289	0.95267	0.95222	0.95156	**0.95311**
		STD	0.00483	0.00379	0.00169	0.00203	0.00253	0.00347	**0.00122**
		Rank	7	5	2	3	4	6	**1**
16	WaveformEW	Mean	0.80548	0.80275	0.80000	0.80947	0.80732	0.80085	**0.81251**
		STD	0.00439	0.00591	0.00342	0.00385	0.00377	0.00492	**0.00313**
		Rank	4	5	7	2	3	6	**1**
17	Wine	Mean	0.96180	0.95918	0.96629	0.96554	**0.96629**	0.96292	0.96592
		STD	0.00560	0.00551	6.78E-16	0.00285	**6.78E-16**	0.00524	0.00205
		Rank	6	7	2	4	**1**	5	3
18	Zoo	Mean	0.98824	0.98758	**1.00000**	**1.00000**	0.99673	0.99412	**1.00000**
		STD	0.01104	0.00961	**0.00000**	**0.00000**	0.00743	0.00914	**0.00000**
		Rank	4	5	**1**	**1**	2	3	**1**
19	CNAE	Mean	0.87637	0.86111	0.87346	0.89303	0.90476	0.88898	**0.92566**
		STD	0.01264	0.01221	0.00917	0.01297	0.01213	0.01570	**0.00682**
		Rank	5	7	6	3	2	4	**1**
20	Connectionist	Mean	0.94064	0.93470	0.92694	0.94338	0.94703	0.93151	**0.95434**
	Bench Data	STD	0.01948	0.02350	0.01582	0.01515	0.01427	0.01609	**0.01314**
		Rank	4	5	7	3	2	6	**1**
21	Lung Cancer	Mean	0.87576	0.88788	0.92121	0.96667	0.94848	0.89697	**0.96970**
		STD	0.06954	0.07803	0.04613	0.04456	0.04582	0.05716	**0.04359**
		Rank	7	6	4	2	3	5	**1**
22	Optical Recognition	Mean	0.97152	0.97029	0.97029	0.97486	0.97495	0.97333	**0.97743**
	of Handwritten	STD	0.00348	0.00408	0.00448	0.00371	0.00256	0.00362	**0.00253**
		Rank	5	6	7	3	2	4	**1**
23	QSAR	Mean	0.85784	0.85847	0.86144	0.86748	0.86694	0.86162	**0.87027**
	biodegradation	STD	0.00740	0.00683	0.00658	0.00423	0.00553	0.00519	**0.00536**
		Rank	7	6	5	2	3	4	**1**
24	SPECT Heart	Mean	0.90833	0.90833	0.90238	0.90595	0.90952	0.89167	**0.92500**
		STD	0.03594	0.04161	0.03501	0.03314	0.03347	0.05345	**0.02710**
		Rank	4	3	6	5	2	7	**1**
25	UJIIndoorLoc	Mean	**1**	**1**	**1**	**1**	**1**	**1**	**1**
		STD	**0**	**0**	**0**	**0**	**0**	**0**	**0**
		Rank	**1**	**1**	**1**	**1**	**1**	**1**	**1**
Friedman mean rank			4.36	4.72	3.48	2.28	2.28	4.12	**1.12**
Final rank			5	6	3	2	2	4	**1**

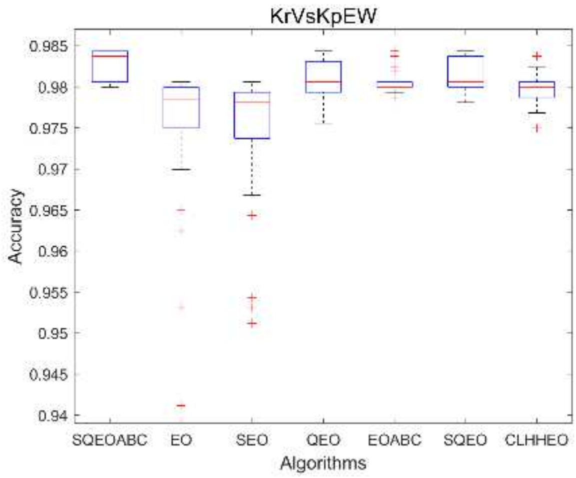
The quality of solutions generated by SQEOABC and the variants of EO are analyzed, and the boxplots of tested datasets are depicted in Fig. 4. X-axis is the name of each algorithm, and y-axis is the accuracy. According to the boxplots, SQEOABC has higher boxplots in the tested datasets with the high median and narrow spread, such as CongressEW, Ionosphere, KrVsKpEW, Sonar, SpectEW, WaveformEW, CNAE, QSAR biodegradation.



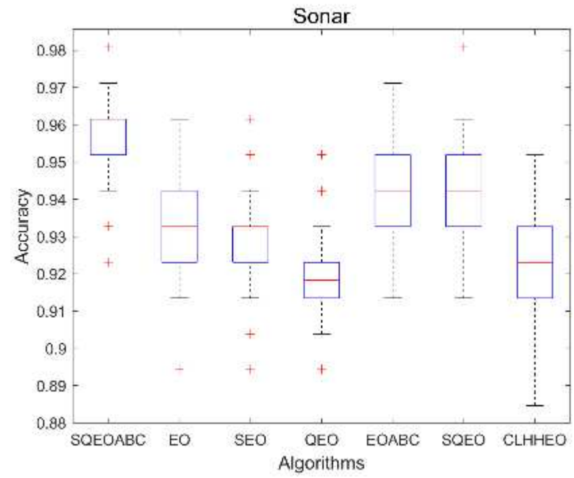
(a) CongressEW



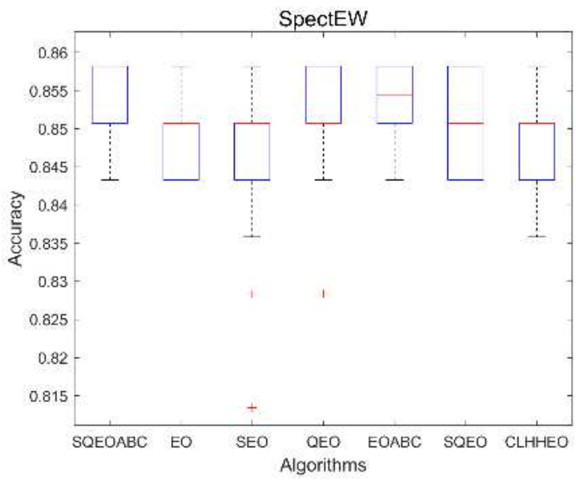
(b) Ionosphere



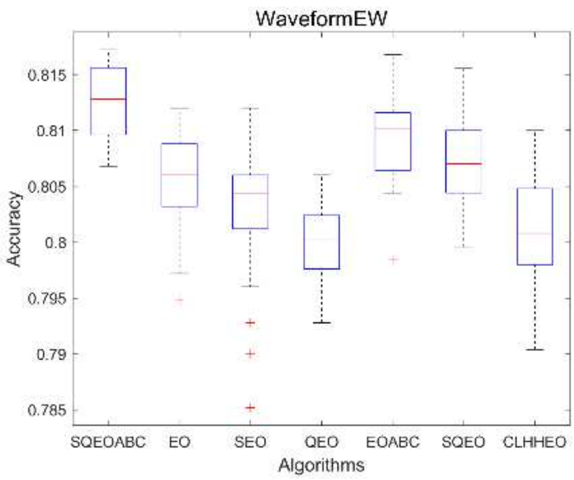
(c) KrVsKpEW



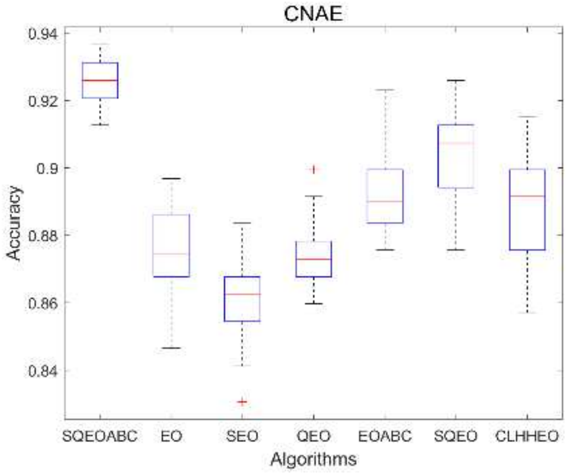
(d) Sonar



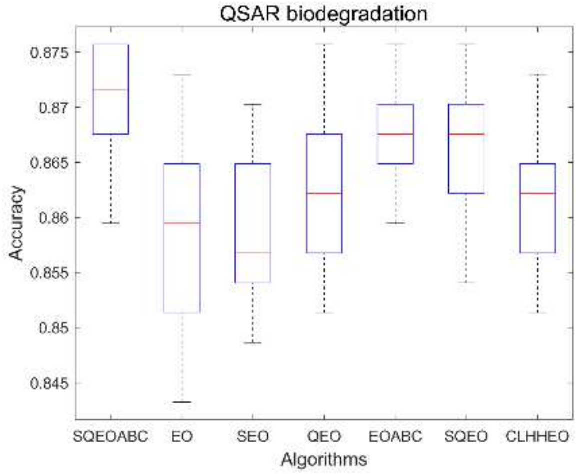
(e) SpectEW



(f) WaveformEW



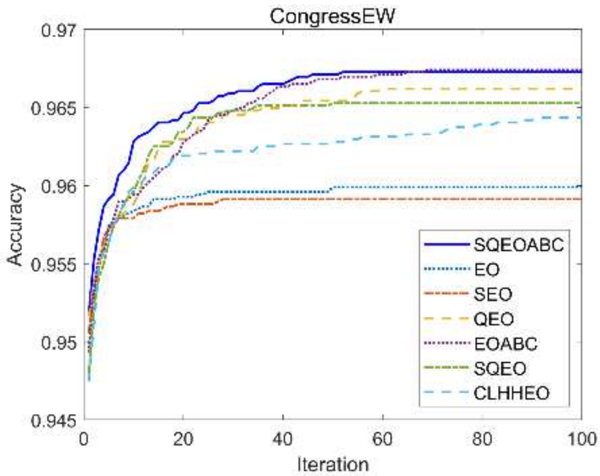
(g) CNAE



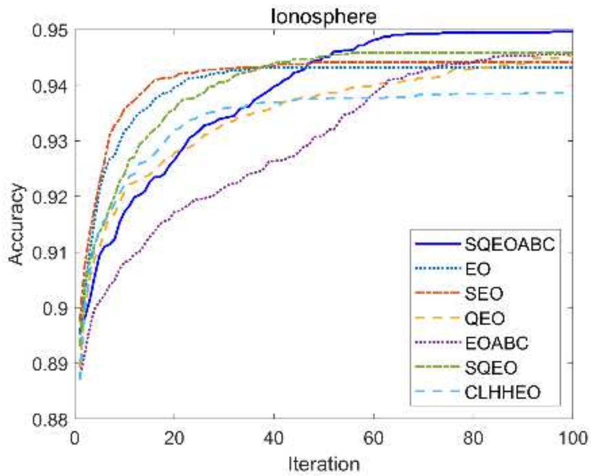
(h) QSAR biodegradation

Fig. 4 Boxplots by SQEOABC and EO variants

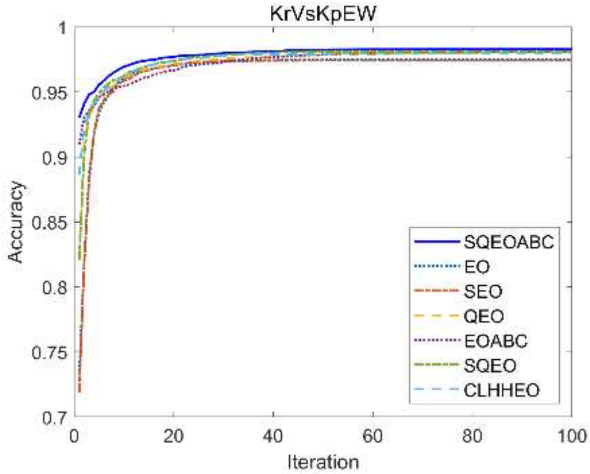
The convergence curves of accuracy obtained by SQEOABC and the variants of EO in the tested datasets are summarized in Fig. 5, illustrating that the proposed SQEOABC algorithm has good performance in the convergence for feature selection problems. Although several compared algorithms can provide competitive convergence behaviors in some datasets, SQEOABC still has better convergence rate than the variants of EO in the datasets, such as Ionosphere, Sonar, SpectEW, WaveformEW, CNAE, QSAR degradation. In a word, the proposed SQEOABC algorithm can achieve the highest classification accuracy and good in dimensionality reduction at most of the datasets.



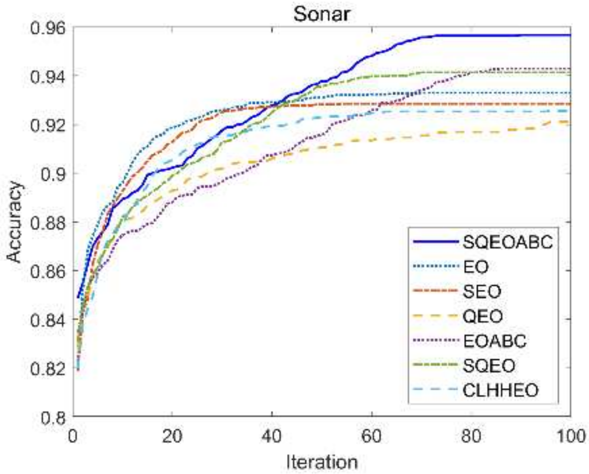
(a) CongressEW



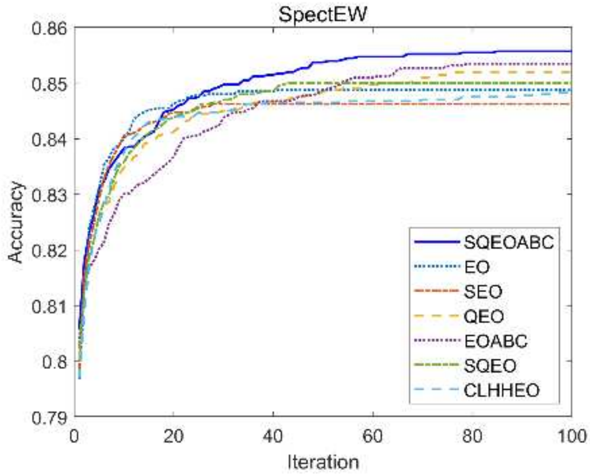
(b) Ionosphere



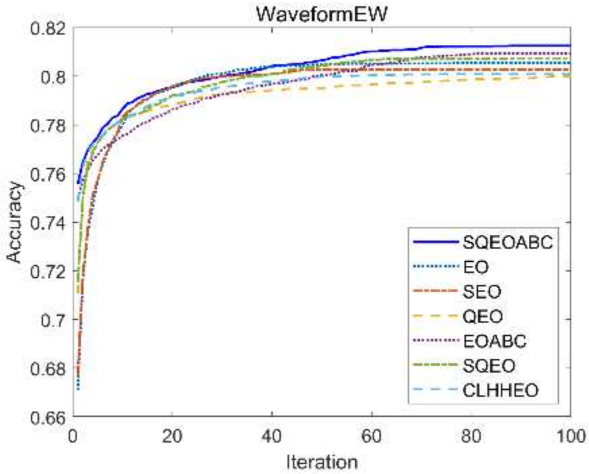
(c) KrVsKpEW



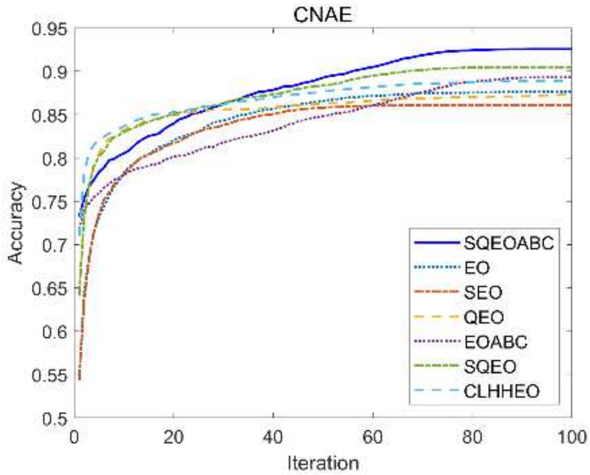
(d) Sonar



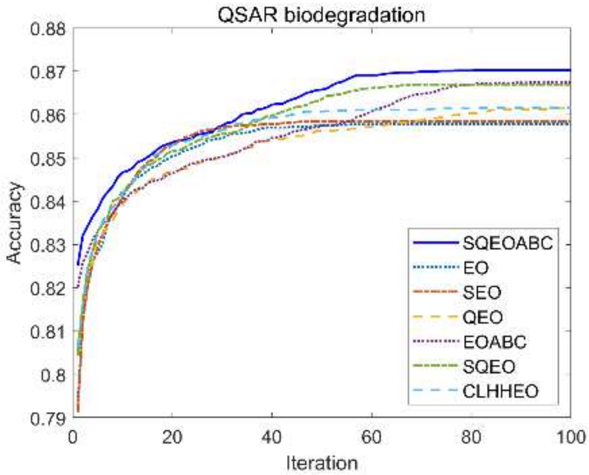
(e) SpectEW



(f) WaveformEW



(g) CNAE



(h) QSAR biodegradation

Fig. 5 Convergence curves of accuracy values by SQEOABC and EO variants

5.4 Real application of feature selection of COVID-19 case

In the above-mentioned experiment, the performance of SQEOABC is verified by 24 datasets from UCI repository. Furthermore, an additional experiment is implemented to demonstrate the effectiveness of SQEOABC for real-world application. COVID-19 is a worldly widespread disease from severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), which caused huge losses to humankind. Until to Sep 2022, it infected more than 600 million people and caused more than 6 million people deaths [112]. Metaheuristic algorithms were also utilized to help researchers in resisting COVID-19 such as prevention and detection [113]. In this subsection, the patient health prediction of COVID-19 is considered to verify the proposed algorithm. This dataset composes of 863 samples and 15 features, including the basic information and some symptoms of patients, where the source data can be found in [32]. The features are transferred into the numerical format, and the dataset is divided into the training dataset and validating dataset, respectively. In this real-world problem, the proposed algorithm is compared with 11 state-of-the-art algorithms, PSO, CMAES, GWO, HHO, SSA, SMA, WSO, TLSMA, DDSRPSO, EESHHO, LSHADE-cnEpSin. The parameters of these algorithms are the same with Table 2.

The statistical results of the proposed algorithm and compared algorithms are provided in Table 7, including the fitness values and accuracy. The results show that SQEOABC obtained the highest classification accuracy of 95.833% and the smallest fitness values of 0.04587, which outperforms other compared algorithms. The Friedman mean rank also illustrates that SQEOABC ranks the first in this FS problem, which is better than PSO, CMAES, GWO, HHO, SSA, SMA, WSO, TLSMA, DDSRPSO, EESHHO, LSHADE-cnEpSin. Besides, some of the compared algorithms can also achieve the min fitness value and the max accuracy, but SQEOABC has better robustness than the compared algorithms. The convergence analysis of this problem is also investigated. Fig. 6 provides the convergence behaviors of fitness values and accuracy obtained by SQEOABC and compared algorithms. It can be seen that SQEOABC provides good convergence rate and obtains the final solution, after 19 iterations in fitness values and 17 iterations in the accuracy, respectively. The results illustrate the effectiveness of SQEOABC in solving real-world feature selection problems.

Table 7 The fitness values of SQEOABC and state-of-the-art algorithms

Algo	Fitness					Accuracy				
	Best	Mean	Worst	STD	Rank	Best	Mean	Worst	STD	Rank
PSO	0.04587	0.04915	0.05732	0.00349	9	0.94676	0.95517	0.95833	0.00336	7
CMAES	0.04587	0.04844	0.06495	0.00487	8	0.93750	0.95571	0.95833	0.00511	6
GWO	0.04587	0.04765	0.05960	0.00389	5	0.94213	0.95625	0.95833	0.00440	5
HHO	0.04587	0.04594	0.04663	0.00023	2	**0.95833**	**0.95833**	**0.95833**	**4.52E-16**	**1**
SSA	0.04587	0.04943	0.05809	0.00353	10	0.94676	0.95471	0.95833	0.00358	8
SMA	0.04968	0.05951	0.07186	0.00643	12	0.93519	0.94460	0.95602	0.00614	11
WSO	0.04587	0.04839	0.06420	0.00411	7	0.93750	0.95162	0.95833	0.00612	10
TLSMA	0.04587	0.05098	0.05960	0.00522	11	0.94213	0.95231	0.95833	0.00582	9
DDSRPSO	0.04587	0.04714	0.05428	0.00251	4	0.95139	0.95725	0.95833	0.00225	3
EESHHO	0.04587	0.04790	0.05351	0.00344	6	0.95139	0.95648	0.95833	0.00312	4
LSHADE-cnEpSin	0.04587	0.04650	0.05351	0.00153	3	0.95139	0.95756	0.95833	0.00165	2
SQEOABC	**0.04587**	**0.04587**	**0.04587**	**2.12E-17**	**1**	**0.95833**	**0.95833**	**0.95833**	**4.52E-16**	**1**

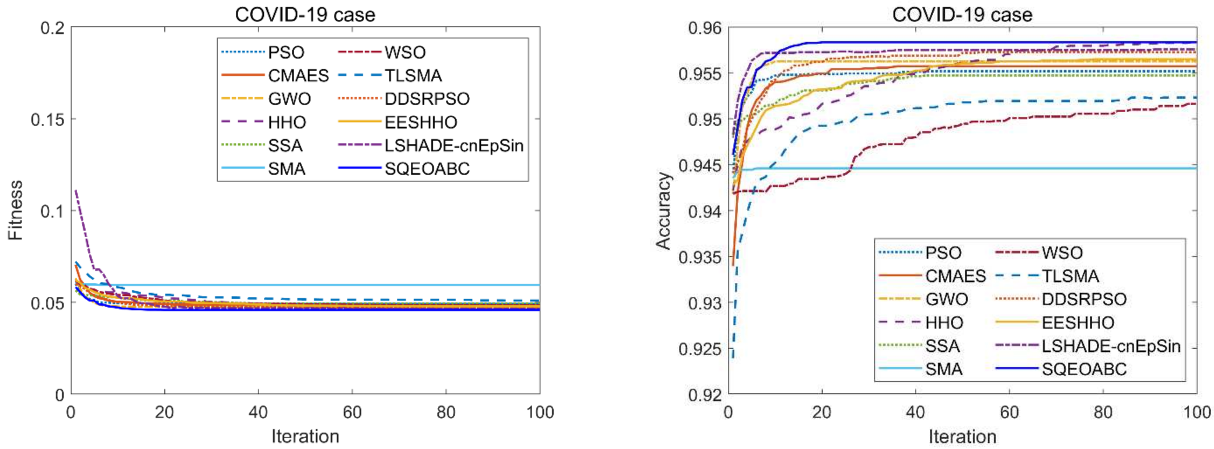


Fig. 6 Convergence curves in the COVID-19 case

5.5 Discussion

According to the above-mentioned results, the proposed SQEOABC algorithm displays superiority on the fitness value and classification accuracy compared to EO in the feature selection problems. SQEOABC also has stable solutions in terms of smaller mean and STD fitness values. Compared with several state-of-the-art metaheuristic algorithms, the proposed SQEOABC algorithm has superiority on the fitness values and provides competitive results on the classification accuracy. In most of the tested datasets, SQEOABC also has faster convergence speed when searching for the optimal solution. The real-world problem in patient health prediction of COVID-19 demonstrates the effectiveness and superiority of SQEOABC.

The reasons for higher accuracy and better convergence of SQEOABC in solving FS problems compared with other algorithms can be discussed as follows. The self-adaptive mechanism in SQEOABC achieves good exploration capacity to enhance the global convergence in FS problems. Besides, the quantum theory is embedded into the updating rule of EO to provide better diversity of search agents, which can increase the searching ability in the design space. Moreover, the combination of ABC and quantum EO plays a key role, where quantum EO is used to control the exploration by calculating the exponential term and generation rate, the updating rule of ABC is introduced to control the exploitation to improve the convergence and accuracy of EO for FS problems. According to the results from Tables 5-6 and Fig. 5, we observe that SQEOABC has better performance than the origin and EO's variants, indicating that the hybridization of three operators in EO is more effective than the single operator into EO in solving FS problems. In all, the SQEOABC achieves good accuracy and convergence for FS problems with the contributions of the above operators for EO.

Except for the benefits, there also exist some limitations in SQEOABC for FS problems. Firstly, SQEOABC is derived from EO and enhanced by three operators, which requires more expensive computational cost than other swarm-intelligent algorithms. Secondly, the kNN classifier is used as the learning algorithm for feature selection, which is easy to be affected by the noisy data especially for large-scale dataset, so other classifier can be further utilized in FS problems, such as support vector machine and random forest. Moreover, compared with the basic EO, the number of algorithmic parameters is not added in the proposed algorithm, but we still expect to develop parameter-free technique in solving various FS problems.

6 Conclusions and future works

To overcome the limitations of equilibrium optimizer (EO) in feature selection, this paper presents the self-adaptive quantum equilibrium optimizer-artificial bee colony (SQEOABC) algorithm to solve FS problems. In SQEOABC, the coefficient of EO is enhanced by the self-adaptive mechanism, and the quantum theory is embedded into the updating rule of EO to increase the diversity of solutions. Besides, the artificial bee colony is also hybridized into the proposed algorithm to avoid the local optimum. The performance of SQEOABC is firstly examined by 25 datasets from UCI repository. The compared metaheuristic algorithms consist of PSO, CMAES, GWO, HHO, SSA, SMA, WSO, TLSMA, DDSRPSO, EESHHO, LSHADE-cnEpSin. The statistical results of fitness values and accuracy, the boxplots of accuracy, and the convergence curves of accuracy are analyzed. The experimental results indicate the superiority of SQEOABC in solving FS problems. Besides, to investigate the influence of different operators of modified EO, several variants of EO are utilized to compare SQEOABC. The results show that SQEOABC outperforms the standard EO and variants of EO in FS problems, and the ensemble strategies employed in EO can enhance the accuracy of classification in FS problems. Finally, the effectiveness of SQEOABC is investigated by a real-world FS problem from COVID-19, and the results also illustrate that SQEOABC can performs better than the compared algorithms in terms of the fitness values and accuracy.

In future studies, the proposed SQEOABC can be hybridized with other algorithms or operators, such as Levy flight, chaos, or other metaheuristic algorithms. SQEOABC versions can be extended to deal with large-scale data problems, by combing with different learning strategies. Furthermore, the modification of SQEOABC will be considered to solve multi-objective feature selection, or other real-world multi-objective optimization problems.

CRediT authorship contribution statement

Changting Zhong: Conceptualization, Methodology, Software, Validation, Writing – original draft, Writing - review & editing. **Gang Li**: Conceptualization, Methodology, Supervision, Funding acquisition, Writing - review & editing. **Zeng Meng**: Writing – review & editing. **Haijiang Li**: Supervision, Writing – review & editing. **Wanxin He**: Writing – review & editing.

Declaration of competing interest

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

Acknowledgement

This work was supported by the National Key Research and Development Program (Grant No. 2019YFA0706803) and the National Natural Science Foundation of China (Grant No. 11872142).

References

[1] B.H. Nguyen, B. Xue, M. Zhang, A survey on swarm intelligence approaches to feature selection in data mining, *Swarm and Evolutionary Computation*, 54 (2020) 100663, https://doi.org/10.1016/j.swevo.2020.100663.

[2] M.H. Nadimi-Shahraki, H. Zamani, S. Mirjalili, Enhanced whale optimization algorithm for medical feature selection: A COVID-19 case study, *Computers in Biology and Medicine*, 148 (2022) 105858, https://doi.org/10.1016/j.compbiomed.2022.105858

[3] X. Wang, Y. Wang, K.C. Wong, X. Li, A self-adaptive weighted differential evolution approach for large-scale feature selection, *Knowledge-Based Systems*, 235 (2022) 107633, https://doi.org/10.1016/j.knosys.2021.107633.

[4] M. Rostami, K. Berahmand, E. Nasiri, S. Forouzandeh, Review of swarm intelligence-based feature selection methods, *Engineering Applications of Artificial Intelligence*, 100 (2021) 104210, https://doi.org/10.1016/j.engappai.2021.104210.

[5] Z. Hu, Y. Bao, T. Xiong, R. Chiong, Hybrid filter-wrapper feature selection for short-term load forecasting, *Engineering Applications of Artificial Intelligence*, 40 (2015) 17−27, https://doi.org/10.1016/j.engappai.2014.12.014.

[6] A. Rashno, M. Shafipour, S. Fadaei, Particle ranking: an efficient method for multi-objective particle swarm optimization feature selection, *Knowledge-Based Systems*, 245 (2022) 108640, https://doi.org/10.1016/j.knosys.2022.108640.

[7] Q. Hu, D. Yu, J. Liu, C. Wu, Neighborhood rough set based heterogeneous feature subset selection, *Information Science*, 178 (2008) 3577−3594, https://doi.org/10.1016/j.ins.2008.05.024.

[8] H. Zhao, P. Wang, Q. Hu, P. Zhu, Fuzzy rough set based feature selection for large-scale hierarchical classification, *IEEE Transactions on Fuzzy Systems*, 27(10) (2019) 1891−1903, https://doi.org/10.1109/TFUZZ.2019.2892349

[9] P. Moradi, M. Gholampour, A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy, *Applied Soft Computing*, 43 (2016) 117−130, http://dx.doi.org/10.1016/j.asoc.2016.01.044.

[10] G. Li, H. Hu, Risk design optimization using many-objective evolutionary algorithm with application to performance-based wind engineering of tall buildings, *Structural Safety*, 48 (2014) 1−14, https://doi.org/10.1016/j.strusafe.2014.01.002.

[11] L. Cui, G. Li, Q. Lin, Z. Du, W. Gao, J. Chen, N. Lu, A novel artificial bee colony algorithm with depth-first search framework and elite-guided search equation, *Information Sciences*, 367-368 (2016) 1012−1044, https://doi.org/10.1016/j.ins.2016.07.022.

[12] Z. Meng, G. Li, X. Wang, S.M. Sait, A.R. Yildiz, A comparative study of metaheuristic algorithms for reliability-based design optimization problems, *Archives of Computational Methods in Engineering*, 28 (2021) 1853−1869, https://doi.org/10.1007/s11831-020-09443-z.

[13] C. Zhong, G. Li, Comprehensive learning harris hawks-equilibrium optimization with terminal replacement mechanism for constrained optimization problems, *Expert Systems With Applications*, 192 (2022) 116432, https://doi.org/10.1016/j.eswa.2021.116432.

[14] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proc. Sixth Int. Symp. Micro Mach. Hum. Sci, 1995 MHS 95, 1995, pp. 39−43, http://dx.doi.org/10.1109/MHS.1995.494215.

[15] N. Hansen, S.D. Müller, P. Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES), *Evolutionary Computation*, 11(1) (2003), 1−18,

https://doi.org/10.1162/106365603321828970.

[16] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems, *Computer-Aided Design*, 43 (2011) 303−315, http://dx.doi.org/10.1016/j.cad.2010.12.015.

[17] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, *Advances in Engineering Software*, 69 (2014) 46−61, http://dx.doi.org/10.1016/j.advengsoft.2013.12.007.

[18] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *Journal of Global Optimization*, 39 (2007) 459−471, http://dx.doi.org/10.1007/s10898-007-9149-x.

[19] L. Cui, G. Li, X. Wang, Q. Lin, J. Chen, N. Lu, L. Jian, A ranking-based adaptive artificial bee colony algorithm for global numerical optimization, Knowledge-Based Systems, 417 (2017) 169−185, https://doi.org/10.1016/j.ins.2017.07.011.

[20] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: algorithm and applications, *Future Generation Computer Systems*, 97 (2019) 849−872, https://doi.org/10.1016/j.future.2019.02.028.

[21] J. Hu, Z. Han, A.A. Heidari, Y. Shou, H. Ye, L. Wang, X. Huang, H. Chen, Y. Chen, P. Wu, Detection of COVID-19 severity using blood gas analysis parameters and Harris hawks optimized extreme learning machine, *Computers in Biology and Medicine*, 142 (2022) 105166, https://doi.org/10.1016/j.compbiomed.2021.105166

[22] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, *Knowledge-Based Systems*, 89 (2015) 228−249, http://dx.doi.org/10.1016/j.knosys.2015.07.006

[23] S. Li, H. Chen, M. Wang, A.A. Heidari, S. Mirjalili. Slime mould algorithm: a new method for stochastic optimization, *Future Generation Computer Systems*, 111 (2020) 300−323, https://doi.org/10.1016/j.future.2020.03.055

[24] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, Salp swarm algorithm: a bio-inspired optimizer for engineering design problems, *Advances in Engineering Software*, 114 (2017) 163−191, http://dx.doi.org/10.1016/j.advengsoft.2017.07.002.

[25] F.A. Hashim, A.G. Hussien, Snake optimizer: a novel meta-heuristic optimization algorithm, *Knowledge-Based Systems*, 242 (2022) 108320, https://doi.org/10.1016/j.knosys.2022.108320.

[26] M. Braik, A. Hammouri, J. Atwan, M.A. Al-Betar, M.A. Awadallah, White shark optimizer: a novel bio-inspired meta-heuristic algorithm for global optimization problems, *Knowledge-Based Systems*, 243 (2022) 108457, https://doi.org/10.1016/j.knosys.2022.108457.

[27] C. Zhong, G. Li, Z. Meng. Beluga whale optimization: A novel nature-inspired metaheuristic algorithm, *Knowledge-Based Systems*, 251 (2022) 109215, https://doi.org/10.1016/j.knosys.2022.109215.

[28] L. Ke, Z. Feng, Z. Ren, An efficient ant colony optimization approach to attribute reduction in rough set theory, *Pattern Recognition Letters*, 29 (2008) 1351−1357, https://doi.org/10.1016/j.patrec.2008.02.006.

[29] H. Chen, T. Li, X. Fan, C. Luo, Feature selection for imbalanced data based on neighborhood rough sets, *Information Sciences*, 483 (2019) 1−20, https://doi.org/10.1016/j.ins.2019.01.041.

[30] X.M. Hu, S.R. Zhang, M. Li, J.D. Deng, Multimodal particle swarm optimization for feature selection, *Applied Soft Computing*, 113 (2021) 107887, https://doi.org/10.1016/j.asoc.2021.107887.

[31] H. Ghimatgar, K. Kazemi, M.S. Helfroush, A. Aarabi, An improved feature selection algorithm based on graph clustering and ant colony optimization, *Knowledge-Based Systems*, 159 (2018) 270−285, https://doi.org/10.1016/j.knosys.2018.06.025.

[32] J. Too, S. Mirjalili. A hyper learning binary dragonfly algorithm for feature selection: a COVID-19 case study, *Knowledge-Based Systems*, 212 (2021) 106553, https://doi.org/10.1016/j.knosys.2020.106553.

[33] D.H. Wolpert, W.G. Macready, No free lunch theorem for optimization, *IEEE Transactions on Evolutionary Computation*, 1(1) (1997) 67−82, https://doi.org/10.1109/4235.585893.

[34] A. Faramarzi, M. Heidarinejad, B. Stephens, S. Mirjalili, Equilibrium optimizer: a novel optimization algorithm, *Knowledge-Based Systems*, 191 (2020) 105190, https://doi.org/10.1016/j.knosys.2019.105190.

[35] J. Wang, B. Yang, D. Li, C. Zeng, Y. Chen, Z. Guo, X. Zhang, T. Tan, H. Shu, T. Yu. Photovoltaic cell parameter estimation based on improved equilibrium optimizer algorithm, *Energy Conversion and Management*, 236 (2021) 114051, https://doi.org/10.1016/j.enconman.2021.114051.

[36] M. Abdel-Basset, V. Chang, R. Mohamed, A novel equilibrium optimization algorithm for multi-thresholding image segmentation problems, *Neural Computing and Applications*, 33 (2021) 10685−10718, https://doi.org/10.1007/s00521-020-04820-y.

[37] M. Mahmoodjanloo, R. Tavakkoli-Moghaddama, A. Baboli, A.Bozorgi-Amiri, Distributed job-shop rescheduling problem considering reconfigurability of machines: a self-adaptive hybrid equilibrium optimizer, *International Journal of Production Research*, 2021, https://doi.org/10.1080/00207543.2021.1946193.

[38] A.M. Shaheen, A.M. Elsayed, R.A. El-Sehiemy, A.Y. Abdelaziz, Equilibrium optimization algorithm for network reconfiguration and distributed generation allocation in power systems, *Applied Soft Computing*, 98 (2021) 106867, https://doi.org/10.1016/j.asoc.2020.106867.

[39] E.H. Houssein, M. Dirar, L. Abualigah, W.M. Mohamed, An efficient equilibrium optimizer with support vector regression for stock market prediction, *Neural Computing and Applications*, 34 (2022) 3165−3200, https://doi.org/10.1007/s00521-021-06580-9.

[40] A. ATES, Enhanced equilibrium optimization method with fractional order chaotic and application engineering, *Neural Computing and Applications*, 33 (2021) 9849−9876, https://doi.org/10.1007/s00521-021-05756-7.

[41] S.K. Dinkar, K. Deep, S. Mirjalili, S. Thapliyal, Opposition-based Laplacian equilibrium optimizer with application in image segmentation using multilevel thresholding, *Expert Systems With Applications*, 174 (2021) 114766, https://doi.org/10.1016/j.eswa.2021.114766.

[42] J. Liu, W. Li, Y. Li, LWMEO: an efficient equilibrium optimizer for complex functions and engineering design problems, *Expert Systems With Applications*, 198 (2022) 116828, https://doi.org/10.1016/j.eswa.2022.116828.

[43] Q. Fan, H. Huang, K. Yang, S. Zhang, L. Yao, Q. Xiong, A modified equilibrium optimizer using opposition-based learning and novel update rules, *Expert Systems With Applications*, 170 (2021) 114575, https://doi.org/10.1016/j.eswa.2021.114575.

[44] X. Zhang, Q. Lin, Information-utilization strengthened equilibrium optimizer, *Artificial Intelligence Review*, 55 (2022) 4241−4274, https://doi.org/10.1007/s10462-021-10105-0.

[45] S. Ahmed, K.K. Ghosh, S. Mirjalili, R. Sarkar, AIEOU: Automata-based improved equilibrium optimizer with U-shaped transfer function for feature selection, *Knowledge-Based Systems*, 228 (2021) 107283, https://doi.org/10.1016/j.knosys.2021.107283.

[46] S. Ouadfel, M.A. Elaziz, Efficient high-dimension feature selection based on enhanced equilibrium optimizer, *Expert Systems With Applications*, 187 (2022) 115882, https://doi.org/10.1016/j.eswa.2021.115882.

[47] D.A. Elmanakhly, M.M. Saleh, E.A. Rashed, An improved equilibrium optimizer algorithm for features

selection: methods and analysis, *IEEE Access*, 9 (2021) 120309−120327, https://doi.org/10.1109/ACCESS.2021.3108097.

[48] G.I. Sayed, G. Khoriba, M.H. Haggag, A novel chaotic equilibrium optimizer algorithm with S-shaped and V-shaped transfer functions for feature selection, *Journal of Ambient Intelligence and Humanized Computing*, 13(6) (2022) 3137−3162, https://doi.org/10.1007/s12652-021-03151-7.

[49] S. Minocha, B. Singh, A novel phishing detection system using binary modified equilibrium optimizer for feature selection, *Computers and Electrical Engineering*, 98 (2022) 107689, https://doi.org/10.1016/j.compeleceng.2022.107689.

[50] U. Agrawal, V. Rohatgi, R. Katarya, Normalized mutual information-based equilibrium optimizer with chaotic maps for wrapper-filter feature selection, *Expert Systems With Applications*, 207 (2022) 118107. https://doi.org/10.1016/j.eswa.2022.118107.

[51] C.L. Huang, J.F. Dun, A distributed PSO-SVM hybrid system with feature selection and parameter optimization, *Applied Soft Computing*, 8 (2008) 1381−1391, https://doi.org/10.1016/j.asoc.2007.10.007.

[52] Y. Liu, G. Wang, H. Chen, H. Dong, X. Zhu, S. Wang, An improved particle swarm optimization for feature selection, *Journal of Bionic Engineering*, 8 (2011) 191−200, https://doi.org/10.1016/S1672-6529(11)60020-6.

[53] J. Wei, R. Zhang, Z. Yu, R. Hu, J. Tang, C. Gui, Y. Yuan, A BPSO-SVM algorithm based on memory renewal and enhanced mutation mechanism for feature selection, *Applied Soft Computing*, 58 (2017) 176−192, http://dx.doi.org/10.1016/j.asoc.2017.04.061.

[54] B. Xue, M. Zhang, W.N. Browne, Particle swarm optimization for feature selection in classification: a multi-objective approach, *IEEE Transactions on Cybernetics*, 43(6) (2013) 1656−1671, https://doi.org/10.1109/TSMCB.2012.2227469.

[55] Y. Prasad, K.K. Biswas, M. Hanmandlu, A recursive PSO scheme for gene selection in microarray data, *Applied Soft Computing*, 71 (2018) 213−225, https://doi.org/10.1016/j.asoc.2018.06.019.

[56] Y. Xue, T. Tang, W. Pang, A.X. Liu, Self-adaptive parameter and strategy based particle swarm optimization for large-scale feature selection problems with multiple classifiers, *Applied Soft Computing*, 88 (2020) 106031, https://doi.org/10.1016/j.asoc.2019.106031.

[57] E. Hancer, B. Xue, D. Karaboga, M. Zhang, A binary ABC algorithm based on advanced similarity scheme for feature selection, *Applied Soft Computing*, 36 (2015) 334−348, http://doi.org/10.1016/j.asoc.2015.07.023.

[58] X. Wang, Y. Zhang, X. Sun, Y. Wang, C. Du, Multi-objective feature selection based on artificial bee colony: an acceleration approach with variable sample size, *Applied Soft Computing*, 88 (2020) 106041, https://doi.org/10.1016/j.asoc.2019.106041.

[59] B. Guan, T. Xu, Y. Zhao, Y. Li, X. Dong, A random grouping-based self-regulating artificial bee colony algorithm for interactive feature selection, *Knowledge-Based Systems*, 243 (2022) 108434, https://doi.org/10.1016/j.knosys.2022.108434.

[60] B.Z. Dadaneh, H.Y. Markid, A. Zakerolhosseini, Unsupervised probabilistic feature selection using ant colony optimization, *Expert Systems With Applications*, 53 (2016) 27−42, http://dx.doi.org/10.1016/j.eswa.2016.01.021.

[61] M. Mafarja, S. Mirjalili, Whale optimization approaches for wrapper feature selection, *Applied Soft Computing*, 62 (2018) 441−453, https://doi.org/10.1016/j.asoc.2017.11.006.

[62] R. Kundu, S. Chattopadhyay, R. Cuevas, R. Sarkar, AltWOA: altruistic whale optimization algorithm for feature selection on microarray datasets, *Computers in Biology and Medicine*, 144 (2022) 105349,

https://doi.org/10.1016/j.compbiomed.2022.105349.

[63] M.A. Awadallah, A.I. Hammouri, M.A. Al-Betar, M.S. Braik, M.A. Elaziz, Binary horse herd optimization algorithm with crossover operators for feature selection, *Computers in Biology and Medicine*, 141 (2022) 105152, https://doi.org/10.1016/j.compbiomed.2021.105152.

[64] E. Emary, H.M. Zawbaa, A.E. Hassanien, Binary grey wolf optimization approaches for feature selection, *Neurocomputing*, 172 (2016) 371−381, http://dx.doi.org/10.1016/j.neucom.2015.06.083.

[65] B. Turkoglu, S.A. Uymaz, R. Kaya, Binary artificial algae algorithm for feature selection, *Applied Soft Computing*, 120 (2022) 108630, https://doi.org/10.1016/j.asoc.2022.108630.

[66] K.K. Ghosh, R. Guha, S.K. Bera, N. Kumar, R. Sarkar, S-shaped versus V-shaped transfer functions for binary Manta ray foraging optimization in feature selection problem, *Neural Computing and Applications*, 33 (2021) 11027−11041, https://doi.org/10.1007/s00521-020-05560-9.

[67] W. Long, M. Xu, J. Jiao, T. Wu, M. Tang, S. Cai, A velocity-based butterfly optimization algorithm for high-dimensional optimization and feature selection, *Expert Systems With Applications*, 201 (2022) 117217, https://doi.org/10.1016/j.eswa.2022.117217.

[68] E. Pashaei, E. Pashaei, An efficiency binary chimp optimization algorithm for feature selection in biomedical data classification, *Neural Computing and Applications*, 34 (2022) 6427−6451, https://doi.org/10.1007/s00521-021-06775-0.

[69] M.A. Awadallah, M.A. Al-Betar, M.S. Braik, A.I. Hammouri, I.A. Doush, R.A. Zitar, An enhanced binary rat swarm optimizer based on local-best concepts of PSO and collaborative crossover operators for feature selection, *Computers in Biology and Medicine*, 147 (2022) 105675, https://doi.org/10.1016/j.compbiomed.2022.105675.

[70] Y. Liu, A.A. Heidari, Z. Cai, G. Liang, H. Chen, Z. Pan, A. Alsufyani, S. Bourouis, Simulated annealing-based dynamic step shuffled frog leaping algorithm: optimal performance design and feature selection, *Neurocomputing*, 503 (2022) 325−362, https://doi.org/10.1016/j.neucom.2022.06.075.

[71] J. Hu, W. Gui, A.A. Heidari, Z. Cai, G. Liang, H. Chen, Z. Pan, Dispersed foraging slime mould algorithm: continuous and binary variants for global optimization and wrapper-based feature selection, *Knowledge-Based Systems*, 237 (2022) 107761, https://doi.org/10.1016/j.knosys.2021.107761.

[72] J. Yang, V. Honavar, Feature subset selection using a genetic algorithm, *IEEE Intelligent Systems and Their Applications*, 13(2) (1998) 44−48, https://doi.org/10.1109/5254.671091.

[73] C.L. Huang, C.J. Wang, A GA-based feature selection and parameters optimization for support vector machines, *Expert Systems With Applications*, 31 (2006) 231−240, https://doi.org/10.1016/j.eswa.2005.09.024.

[74] S. Sayed, M. Nassef, A. Badr, I. Farag, A nested genetic algorithm for feature selection in high-dimensional cancer microarray datasets, *Expert Systems With Applications*, 121 (2019) 233−243, https://doi.org/10.1016/j.eswa.2018.12.022.

[75] E.D. Carvalho, R.R.V. Silva, F.H.D. Araújo, R.D.A.L. Rabelo, A.O. de Carvalho Filho, An approach to the classification of COVID-19 based on CT scans using convolutional features and genetic algorithms, *Computers in Biology and Medicine*, 136 (2021) 104744, https://doi.org/10.1016/j.compbiomed.2021.104744.

[76] L. Meenachi, S. Ramakrishnan, Metaheuristic search based feature selection methods for classification of cancer, *Pattern Recognition*, 119 (2021) 108079, https://doi.org/10.1016/j.patcog.2021.108079.

[77] A. Al-Ani, A. Alsukker, R.N. Khushaba, Feature subset selection using differential evolution and a wheel based search strategy, *Swarm and Evolutionary Computation*, 9 (2013) 15−26,

http://doi.org/10.1016/j.swevo.2012.09.003.

[78] Y. Zhang, D. Gong, X. Gao, T. Tian, X. Sun, Binary differential evolution with self-learning for multi-objective feature selection, *Information Sciences*, 507 (2020) 67−85, https://doi.org/10.1016/j.ins.2019.08.040.

[79] S. Benchaou, M. Nasri, O. El Melhaoui, Feature selection based on evolution strategy for character recognition, *International Journal of Image and Graphics*, 18(3) (2018) 1850014, https://doi.org/10.1142/S0219467818500146.

[80] D. Albashish, A.I. Hammouri, M. Braik, J. Atwan, S. Sahran, Binary biogeography-based optimization based SVM-RFE for feature selection, *Applied Soft Computing*, 101 (2021) 107026, https://doi.org/10.1016/j.asoc.2020.107026.

[81] J.C.W. Debuse, V.J. Rayward-smith, Feature subset selection within a simulated annealing data mining algorithm, *Journal of Intelligent Information Systems*, 9 (1997) 57−81, https://doi.org/10.1023/A:1008641220268.

[82] R. Meiri, J. Zahavi, Using simulated annealing to optimize the feature selection problem in marketing applications, *European Journal of Operational Research*, 171(3) (2006) 842−858. https://doi.org/10.1016/j.ejor.2004.09.010.

[83] X. Han, X. Chang, L. Quan, X. Xiong, J. Li, Z. Zhang, Y. Liu, Feature subset selection by gravitational search algorithm optimization, *Information Sciences*, 281 (2014) 128−146, http://dx.doi.org/10.1016/j.ins.2014.05.030.

[84] R. Guha, M. Ghosh, A. Chakrabarti, R. Sarkar, S. Mirjalili, Introducing clustering based population in binary gravitational search algorithm for feature selection, *Applied Soft Computing*, 93 (2020) 106341, https://doi.org/10.1016/j.asoc.2020.106341.

[85] G. Al-rawashdeh, R. Mamat, N.H.B.A. Rahim, hybrid water cycle optimization algorithm with simulated annealing for spam e-mail detection, *IEEE Access*, 7 (2019) 143721−143734, https://doi.org/10.1109/ACCESS.2019.2944089.

[86] H. Faris, M.A. Hassonah, A.M. Al-Zoubi, S. Mirjalili, I. Aljarah, A multi-verse optimizer approach for feature selection and optimizing SVM parameters based on a robust system architecture, *Neural Computing and Applications*, 30 (2018) 2355−2369, https://doi.org/10.1007/s00521-016-2818-2.

[87] D. Rodrigues, L.A. Pereira, J.P. Papa, C.C. Ramos, A.N. Souza, L.P. Papa, Optimizing feature selection through binary charged system search, Springer, International Conference on Computer Analysis of Images and Patterns, 377−384.

[88] E. Pashaei, N. Aydin, Binary black hole algorithm for feature selection and classification on biological data, *Applied Soft Computing*, 56 (2017) 94−106, http://dx.doi.org/10.1016/j.asoc.2017.03.002.

[89] J. Too, A.R. Abdullah, Chaotic atom search optimization for feature selection, *Arabian Journal of Science and Engineering*, 45 (2020) 6063−6079, https://doi.org/10.1007/s13369-020-04486-7.

[90] Z.W. Geem, J.H. Kim, A new heuristic optimization algorithm: harmony search, *Simulation*, 76(2) (2001) 60−68, https://doi.org/10.1177/003754970107600201.

[91] S. Mirjalili, SCA: a sine cosine algorithm for solving optimization problems, *Knowledge-Based Systems*, 96 (2016) 120−133, http://doi.org/10.1016/j.knosys.2015.12.022.

[92] H.H. Inbarani, M. Bagyamathi, A. T. Azar, A novel hybrid feature selection method based on rough set and improved harmony search, *Neural Computing and Applications*, 26 (2015): 1859−1880. http://doi.org/ 10.1007/s00521-015-1840-0

[93] A.K. Shukla, P. Singh, M. Vardhan, A new hybrid wrapper TLBO and SA with SVM approach for gene

expression data, *Information Sciences*, 503 (2019) 238−254, https://doi.org/10.1016/j.ins.2019.06.063.

[94] D. Pradhan, B. Sahoo, B.B. Misra, S. Padhy, A multiclass SVM classifier with teaching learning based feature subset selection for enzyme subclass classification, *Applied Soft Computing*, 96 (2020) 106664, https://doi.org/10.1016/j.asoc.2020.106664.

[95] F.O. Sameer, M.J. Al-obaidi, W.W. Al-bassam, A.H. Ad'haih, Multi-objectives TLBO hybrid method to select the related risk features with rheumatism disease, *Neural Computing and Applications*, 33 (2021) 9025−9034, https://doi.org/10.1007/s00521-020-05665-1.

[96] J. Gholami, F. Pourpanah, X. Wang, Feature selection based on improved binary global harmony search for data classification, *Applied Soft Computing*, 93 (2020) 106402, https://doi.org/10.1016/j.asoc.2020.106402.

[97] K. Hussain, N. Neggaz, W, Zhu, E.H. Houssein, An efficient hybrid sine-cosine Harris hawks optimization for low and high-dimensional feature selection, *Expert Systems With Applications*, 176 (2021) 114778, https://doi.org/10.1016/j.eswa.2021.114778.

[98] G.A. Kale, U. Yüzgeç, Advanced strategies on update mechanism of sine cosine optimization algorithm for feature selection in classification problems, *Engineering Applications of Artificial Intelligence*, 107 (2022) 104506, https://doi.org/10.1016/j.engappai.2021.104506.

[99] M. Tubishat, S. Ja'afar, N. Idris, M.A. Al-Betar, M. Alswaitti, H. Jarrah, M.A. Ismail, M.S. Omar, Improved sine cosine algorithm with simulated annealing and singer chaotic map for Hadith classification, *Neural Computing and Applications*, 34 (2022) 1385−1406, https://doi.org/10.1007/s00521-021-06448-y.

[100] M. Moradkhani, A. Amiri, M. Javaherian, H. Safari, A hybrid algorithm for feature subset selection in high-dimensional datasets using FICA and IWSSr algorithm, *Applied Soft Computing*, 35 (2015) 123−135, http://dx.doi.org/10.1016/j.asoc.2015.03.049.

[101] L. Abualigah, A. Diabat, Chaotic binary group search optimizer for feature selection, *Expert Systems With Applications*, 192 (2022) 116368, https://doi.org/10.1016/j.eswa.2021.116368.

[102] J. Sun, W. Fang, V. Palade, X. Wu, W. Xu, Quantum-behaved particle swarm optimization with Gaussian distributed local attractor point, *Applied Mathematics and Computation*, 218 (2011) 3763−3775, https://doi.org/10.1016/j.amc.2011.09.021.

[103] C. Yu, A.A. Heidari, X. Xue, L. Zhang, H. Chen, W. Chen, Boosting quantum rotation gate embedded slime mould algorithm, *Expert Systems With Applications*, 181 (2021) 115082. https://doi.org/10.1016/j.eswa.2021.115082.

[104] R.K. Agrawal, B. Kaur, P. Agarwal, Quantum inspired particle swarm optimization with guided exploration for function optimization, *Applied Soft Computing Journal*, 102 (2021) 107122, https://doi.org/10.1016/j.asoc.2021.107122.

[105] S. Nama, S. Sharma, A.K. Saha, A.H. Gandomi, A quantum mutation-based backtracking search algorithm, Artificial Intelligence Review, 55 (2022) 3019−3073, https://doi.org/10.1007/s10462-021-10078-0.

[106] M. Clerc, J. Kennedy, The particle swarm-explosion, stability, and convergence in a multidimensional complex space, *IEEE Transactions on Evolutionary Computation*, 6(1) (2002) 58−73. https://doi.org/10.1109/4235.985692.

[107] H. Su, D. Zhao, F. Yu, A.A. Heidari, Y. Zhang, H. Chen, C. Li, J. Pan, S. Quan, Horizontal and vertical search artificial bee colony for image segmentation of COVID-19 X-ray images, *Computers in Biology and Medicine*, 142 (2022) 105181, https://doi.org/10.1016/j.compbiomed.2021.105181.

[108] C. Zhong, G. Li, Z. Meng, A hybrid teaching-learning slime mould algorithm for global optimization and reliability-based design optimization problems, *Neural Computing and Applications*, 34 (2022) 16617−16642, https://doi.org/10.1007/s00521-022-07277-3.

[109] M.R. Tanweer, R. Auditya, S. Suresh, N. Sundararajan, N. Srikanth, Directionally driven self-regulating particle swarm optimization algorithm, *Swarm and Evolutionary Computation*, 28 (2016) 98−116, https://doi.org/10.1016/j.swevo.2016.01.006.

[110] C. Li, J. Li, H. Chen, A.A. Heidari, Memetic Harris hawks optimization: developments and perspectives on project scheduling and QoS-aware web service composition, *Expert Systems With Applications*, 171 (2021) 114529, https://doi.org/10.1016/j.eswa.2020.114529.

[111] N.H. Awad, M.Z. Ali, P.N. Suganthan, Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems. *In: 2017 IEEE Congress on Evolutionary Computation, CEC 2017 - Proceedings. Institute of Electrical and Electronics Engineers Inc.*, pp 372–379.

[112] WHO Coronavirus (COVID-19) Dashboard, https://covid19.who.int

[113] A.T. Sahlol, D. Yousri, A.A. Ewees, M.A.A. Al-qaness, R. Damasevicius, M.A. Elaziz. COVID-19 image classification using deep features and fractional-order marine predators algorithm. *Scientific Reports*, 10 (2020), 15364. https://doi.org/10.1038/s41598-020-71294-2.