A Multi-objective Optimization Method for Joint Feature Selection and Classifier Parameter Tuning

Yanyun Pang, Aimin Wang, Yuying Lian, Jiahui Li, and Geng Sun(✉)

College of Software and College of Computer Science and Technology,
Jilin University, Changchun 130012, China
{wangam,lianyuying,sungeng}@jlu.edu.cn

Abstract. Feature selection is an efficient method to extract useful information embedded in the data so that improving the performance of machine learning. However, as a direct factor that affects the result of feature selection, classifier performance is not widely considered in feature selection. In this paper, we formulate a multi-objective minimization optimization problem to simultaneously minimize the number of features and minimize the classification error rate by jointly considering the optimization of the selected features and the classifier parameters. Then, we propose an *Improved Multi-Objective Gray Wolf Optimizer* (IMOGWO) to solve the formulated multi-objective optimization problem. First, IMOGWO combines the discrete binary solution and the classifier parameters to form a mixed solution. Second, the algorithm uses the initialization strategy of tent chaotic map, sinusoidal chaotic map and *Opposition-based Learning* (OBL) to improve the quality of the initial solution, and utilizes a local search strategy to search for a new set of solutions near the Pareto front. Finally, a mutation operator is introduced into the update mechanism to increase the diversity of the population. Experiments are conducted on 15 classic datasets, and the results show that the algorithm outperforms other comparison algorithms.

Keywords: Feature selection · Classification · Multi-objective optimization · Grey wolf optimizer

1 Introduction

With the increasing application of computer technology [12,20,21] and artificial intelligence [11,13], different equipment in different industries will generate a large number of high-dimensional datasets [18,19,22]. These high-dimensional datasets promote the application of machine learning in many fields, such as data mining [6], physics [3], medical imaging [1] and finance [5]. However, as the complexity of machine learning models increases, an increasing number of high-dimensional feature space datasets are generated, often containing many

G. Memmi et al. (Eds.): KSEM 2022, LNAI 13369, pp. 237–248, 2022.
https://doi.org/10.1007/978-3-031-10986-7_19

irrelevant and redundant features. These features will cause the machine learning algorithm to consume too many computing resources when performing classification and other operations and reduce the classification accuracy. Therefore, it is necessary to use the method of feature selection to process the dataset.

The main idea of feature selection is to eliminate irrelevant features from all feature spaces and select the most valuable feature subset [23]. Because irrelevant features are deleted, the number of features specified in the dataset is reduced, so the classification accuracy of machine learning algorithms can be improved and computing resources can be saved. Generally speaking, feature selection methods are divided into three categories, namely, filter-based methods, wrapper-based methods, and embedded methods. Among them, the wrapper-based method trains the selected feature subset with the classifier and uses the training result (accuracy rate) as the evaluation index of the feature subset. Therefore, wrapper-based methods generally perform better than filter-based methods in classification accuracy.

The wrapper-based feature selection methods depend on the performance of the classifier. In general, the feature selection results obtained by using simple classifiers (e.g., KNN and NB) are still applicable in complex classifiers. However, when a complex classifier is used for feature selection, the obtained feature selection results are not applicable since the results are affected by the structure of the classifier. For that matter, the classification parameters also have some influence on the feature selection results. Therefore, the optimization of feature selection can be performed in combination with the classification parameters of the classifier. However, it may increase the complexity of the solution space.

With the continuous increase of the number of features, the search space of feature selection algorithms is getting larger and larger. Therefore, the key factor that affects the feature selection algorithm is the search technology. Traditional subset search techniques, such as *Sequential Forward Selection* (SFS) and *Sequential Backward Selection* (SBS), can find the best feature subset, but this search method is too inefficient and requires a lot of computing time. Compared with traditional search technology, meta-heuristic algorithms such as *dragonfly algorithm* (DA) [14], *particle swarm optimization* (PSO) [7], *whale optimization algorithm* (WOA) [15], *ant colony optimization* (ACO) [9], *genetic algorithm* (GA) [2] has good global search capabilities, and is widely used to solve feature selection problems.

The *grey wolf optimization* (GWO) [16] is a group intelligence optimization algorithm proposed by Mirjalili in 2014. It has the characteristics of strong convergence performance, few parameters, and easy implementation. However, no one algorithm can solve all optimization problems. In addition, the traditional GWO cannot solve the feature selection problem, and there may be some shortcomings, such as easy to fall into local optimality and convergence high speed. Therefore, it is necessary to improve the traditional GWO to make it more suitable for feature selection.

In fact, feature selection is a problem with two goals of minimization: minimizing the classification error rate and minimizing the number of features. Usually these two goals are contradictory, and an optimization algorithm is needed to find their balance. Therefore, this paper proposes to use IMOGWO to solve the feature selection problem.

The main contributions of this research are summarized as follow:

- We consider a multi-objective optimization method for joint feature selection and classifier parameter tuning by formulating a problem with a hybrid solution space.
- We propose an IMOGWO algorithm to solve the formulated problem, in which an initialization strategy using tent chaotic maps, sinusoidal chaotic maps, and OBL to improve the initial solution. Moreover, we propose a local search strategy to search for a new set of solutions near the Pareto front. Furthermore, in the update mechanism, we introduce a mutation operator to improve the diversity of the population.
- Experiments are conducted on 15 classical datasets to evaluate the performance of using IMOGWO for feature selection, and the results are compared with other algorithms to verify the effectiveness of the improvement factor.

The rest of the paper is structured as follows. Section2 introduces the problem construction for feature selection. Section 3 introduces the conventional MOGWO, followed by the proposed IMOGWO. Section 4 presents the experimental results. Section 5 presents the conclusions and possible future work directions.

2 Problem Formulation

In the problem of feature selection, a dataset includes $N_{row} \times N_{dim}$ specific feature values, where each row represents a sample data, and each column represents the specific values of different sample data about the same feature. The purpose of feature selection is to select the most valuable R features to reduce the data dimension. However, in order to improve the classification performance of the SVM complex classifier, we consider the joint features and the parameters of the classifier, and adjust the parameters of the classifier while selecting the features.

We divide the solution space of feature selection into two parts, the first part is the feature vector N_{dim}, each bit has a value of 1 or 0 (1 and 0 indicate that the feature is selected and not selected, respectively). The second part is the parameter vector N_{parm} of the SVM classifier, which consists of three bits. The first one is the *kernel*, and the value is 1 or 0 (1 and 0 indicate the selection of the Gaussian radial basis kernel function and the hyperbolic tangent kernel function, respectively). The second bit is the parameter c, in the range $[2^{-1}, 2^{5}]$, and the third bit is the parameter *gamma*, in the range $[2^{-4}, 2^{5}]$. Therefore, the feature selection situation of the dataset can be represented by a one-dimensional array $X = (x_1, x_2, \cdots, x_{N_{dim}}, x_k, x_c, x_g)$, which is a solution to the feature selection problem.

Since feature selection has two objectives, the feature selection problem is a multi-objective optimization problem (MOP). The first objective to minimize the classification error rate can be expressed as:

$$f_1(X) = 1 - f_{acc}(X), \tag{1}$$

where $f_{acc}(X)$ represents the classification accuracy of the obtained feature subset. Furthermore, the second goal is to reduce the dimension of feature subsets, which can be expressed as:

$$f_2(X) = \frac{R}{N_{dim}}, \tag{2}$$

where R represents the number of features selected, and N_{dim} represents the total number of features. Accordingly, the feature selection problem can be formulated as follows:

$$\min_{\{X\}} \& F = \{f_1, f_2\}, \tag{3a}$$

$$\text{s.t. } C1 : x_i \in \{0, 1\}, \quad i = 1, 2, 3, \cdots, N_{dim} \tag{3b}$$

$$C2 : x_k \in \{0, 1\}, \tag{3c}$$

$$C3 : x_c \in \left[2^{-1}, 2^5\right], \tag{3d}$$

$$C4 : x_g \in \left[2^{-4}, 2^5\right], \tag{3e}$$

$$C5 : 1 < R < N_{dim}, \tag{3f}$$

$$C6 : 0 \leq f_{acc}(X) \leq 1, \tag{3g}$$

where $C1$ represents the restriction of the value range of each dimension of the feature vector in the solution, $C2$ represents the restriction of the classifier parameter *kernel* in the solution, $C3$ represents the restriction of the classifier parameter c in the solution, and $C4$ represents the constraints of the classifier parameter *gamma* in the solution, $C5$ represents the restriction of the number of selected features, and $C6$ represents the restriction of the classification accuracy.

3 The Proposed Approach

In this section, we introduce traditional MOGWO, then introduce IMOGWO with an improvement factor and use it to solve feature selection problems.

3.1 Traditional MOGWO

Although GWO is relatively new, it is designed to solve single-objective optimization problems. Therefore, it cannot be directly used to solve MOP. [17] proposed MOGWO as a multi-objective variant of GWO. In MOGWO, two new mechanisms are integrated: the first is the archive, which is used to store all non-dominated Pareto optimal solutions obtained so far, and the second is the leader selection strategy, which is responsible for selecting the first leader (α), the second leader (β), and the third leader (δ) from the archive.

Archive: As the algorithm is continuously updated, each iteration will produce a new solution. If there is a solution in the archive that dominates the new solution, it will not be added to the archive. If the new solution dominates one or more solutions in the archive, the new solution replaces the solution dominated by it in the archive. If the new solution and all the solutions in the archive do not dominate each other, the new solution is also added to the archive. However, in such an iterative process, there may be more and more individuals in the archive. Therefore, we will set an upper limit for the archive so that the number of individuals in the archive population does not exceed the upper limit while maintaining population diversity. If the number of individuals in the archive exceeds the set value, use the method in NSGA-II [4] to delete the population.

Leader Selection Strategy: In GWO, the leader (α, β, δ) can be determined by the value of the objective function, but in MOP, the pros and cons of the individual are determined by the dominance relationship and cannot be distinguished by simple function values. To redefine the selection mechanism of α, β and δ wolves. The current optimal solution of the algorithm is stored in the archive. Therefore, the individual is directly selected as the leader from the archive. At the same time, in order to improve the exploration ability of the algorithm, the wolf with a large crowding distance is selected as the leader.

3.2 IMOGWO

MOGWO has some shortcomings in solving optimization problems, such as strong global search ability but weak local search ability. Therefore, we propose IMOGWO, which introduces chaotic mapping and opposition-based learning initialization strategy, local search strategy and mutation operator to improve the performance of MOGWO. The proposed IMOGWO pseudocode is given in Algorithm 1, and the details of introducing the improvement factor are as follows.

Mixed Solution of MOGWO. MOGWO was originally to solve the continuous optimization problem and cannot be directly used to solve the feature selection problem. Therefore, for discrete binary solutions, we introduce a v-shaped transfer function to map the solution from continuous space to discrete space, so that the algorithm is suitable for feature selection problems. The description of the v-shaped transfer function is as follows:

$$P(x) = \left| \frac{x}{\sqrt{x^2+1}} \right|, \tag{4}$$

where x represents the position of the gray wolf, and the v-shaped transfer function can frequently change the search variable, which promotes the exploration of the solution space. The method to update the dimensions of the binary solution is as follows:

$$x_{t+1} = \begin{cases} \neg x_t, & N_{rand} < P(x) \\ x_t, & otherwise \end{cases} \tag{5}$$

where x_t is a dimension of a binary solution in iteration t. Through the above binary mechanism, the continuous solution of traditional MOGWO can be effectively transformed into discrete solution, so that the algorithm can be used for feature selection problem.

Algorithm 1: Pseudocode of the Proposed IMOGWO

```
Define and initialize the related parameters: population size $N_{pop}$, solution
  dimension $N_{dim}$ and objective functions, etc.;
Use Eqs. (6), (7) and (8) to initialize the population $pop$;
The opposition population $Opop$ is generated by Eq. (9);
Calculate the objective values for each solution;
Select $N_{pop}$ fittest ones from $\{pop \cup Opop\}$ as the initial population;
Find non-dominated solutions and put them in the Archive;
Sort the non-dominated solutions in descending order of crowding distance;
while $t < T$ do
    Update $X_\alpha$, $X_\beta$, $X_\delta$ according to the crowded distance;
    Update the $j$th dimension of $X_\alpha$ by using Eqs. (11), (12);
    for $i = 1$ to $N_{pop}$ do
        Update the position of $X_i$ using Algorithm 2;
    end
    Calculate the objective values of new solutions and update the archive;
    if the archive is full then
        Use methods in NSGA-II to remove partial members in archive;
    end
    t=t+1;
end
Return archive;
```

TS-OBL Initialization Strategy. The traditional GWO algorithm uses a random initialization method, which is easy to implement, but this may lead to uneven distribution of wolves and low coverage of the solution space. Therefore, we introduce chaotic map and *Opposition-based Learning* (OBL) strategies to improve the initial solution.

The population initialization strategy of OBL is to find the optimal solution from two directions. It can be combined with other strategies. After using a certain strategy to generate the initial solution, OBL is used to generate its opposite solution, so as to make the distribution of the initial solution more uniform and the diversity of the population higher. Therefore, we combine OBL with chaotic map initialization strategy and propose a TS-OBL initialization strategy. The specific process is as follows:

First, when the number of features is less than 30, half of the individuals in the population adopt the strategy of random initialization, and half of the individuals adopt the strategy of tent chaotic map initialization. When the number of features is greater than or equal to 30, all individuals in the population use the sinusoidal chaotic map initialization strategy. The calculation method of the chaotic map is as follows:

$$tent: y_{i+1} = \begin{cases} \frac{y_i}{0.7}, & y_i < 0.7 \\ \frac{10}{3}(1-y_i), & y_i \geq 0,7 \end{cases} \quad i = 1,2,3,\cdots,T/2 \tag{6}$$

$$Sinusoidal: y_{i+1} = ay_i^2 sin(\pi y_i), \quad a = 2.3 \quad i = 1,2,3,\cdots,T \tag{7}$$

where y_i is the ith chaotic variable and T is the maximum number of iterations. The calculation method for mapping the chaotic variable y_i from the interval $[0,1]$ to the corresponding solution space is as follows:

$$x_i = l_i + (u_i - l_i)\, y_i, \tag{8}$$

where l_i and u_i are the minimum and maximum values of the variable value range.

Second, OBL is used to generate the opposite solution of the initial solution. The calculation formula of each dimension of the inverse solution is as follows:

$$\tilde{x_d} = l_d + u_d - x_d. \tag{9}$$

Finally, calculate the two objective function values of $2N_{pop}$ initial solutions, use NSGA-II to sort the initial solutions, and select the best N_{pop} solutions as the initial solutions.

Local Search Strategy. Because MOGWO easily falls into the local optimal value in the later stage of the calculation operation. Only a small perturbation of the non-dominated solution set can make the wolf search in a new direction to increase the probability of finding a new optimal value. Therefore, in the later iteration stage, we change the value of the classifier parameter c or *gamma* of each solution in the non-dominated solution set as the newly generated nearby solution. At the same time, to not increase the computational time complexity, we discard the poorer half of the population and select the better half for updates. The method to find the solutions around the non-dominated solution set is as follows:

$$x^{new} = x + rx, \tag{10}$$

where $r \in [-0.5, 0.5]$, x is the value of the parameter c or *gamma* in the solution. The main steps of the update mechanism using the local search strategy are shown in Algorithm 2.

Novel Update Mechanism with Mutation Operators. The traditional MOGWO may reduce the diversity of the population. For example, when x_α is far from the optimal solution or falls into the local optimal, the performance of the algorithm may be greatly reduced. Therefore, we set a threshold φ to determine whether a certain feature dimension of α wolf has been mutated. The formula for calculating the threshold φ is as follows:

$$\varphi = 0.5(\frac{t}{T}), \tag{11}$$

where t is the current number of iterations. The mutation method of α is as follows:

$$x_\alpha^j = \begin{cases} 1 - x_\alpha^j, & N_{rand} < \varphi \ , N_{dim} < 60 \\ x_\alpha^j, & otherwise \end{cases} \tag{12}$$

where x_α^j represents an unselected feature dimension of α wolf.

Algorithm 2: Update Mechanism using Local Search Strategy

```
Define and initialize the related parameters;
while $t < T$ do
    if $t < 175$ or $N_{dim} >= 60$ then
        $N_{pop} = N_{pop}$;
    end
    else
        $N_{pop} = N_{pop}/2$;
        Sort the solutions in the population from good to bad according to
        NSGA-II;
    end
    for $i = 1$ to $N_{pop}$ do
        Updating $X_i$ according to the mathematical model of the grey wolf
        algorithm;
    end
    Calculate the objective values of new solutions and update the archive;
    if $t < 175$ or $N_{dim} >= 60$ then
        for each non-dominated solution do
            Update the position of the current non-dominated solution by Eq.
            (10) to generate a new solution;
        end
    end
end
Return new solutions;
```

4 Experimental Results and Analysis

In this section, we will conduct tests to verify the performance of IMOGWO on the feature selection problem. In addition, we introduce the datasets and setups used in the experiments, and the test results of IMOGWO and several other multi-objective feature selection methods are given and analyzed.

4.1 Datasets and Setups

In order to verify the proposed multi-objective optimization algorithm, we used 15 benchmark datasets collected from UC Irvine Machine Learning Repository. [10] and [8] describe the main information of the used datasets.

In this paper, we use MOPSO, MODA, MOGWO and NSGA-II as comparative experiments. Note that the multi-objective algorithm here is different from the traditional algorithm in that the solutions are all mixed solutions that combine discrete binary solutions and continuous solutions. They are the same as the solution form of IMOGWO. In addition, for the fairness of comparison, each algorithm sets the same population size (20) and iteration (200). In order to avoid the randomness of the experiment, each algorithm runs 10 times on the selected dataset, and 70% of the instances are used for training and 30% for testing.

We used Python 3.8 to conduct the experiment and use SVM to solve the data classification problem.

Table 1. Classification error rate and number of selected features obtained by different algorithms.

Dataset	MOPSO		MODA		MOGWO		NSGA-II		IMOGWO	
	err.	N_{fea}	err.	N_{fea}	err.	N_{fea}	err.	N_{fea}	err.	N_{fea}
Arrhythmia	0.3642	99.13	0.4020	64.67	0.4332	4.33	0.3283	135.09	**0.2962**	**3.15**
Breastcancer	0.0329	**2.20**	0.0296	2.40	0.0379	2.30	0.0288	2.55	**0.0263**	**2.20**
BreastEW	0.0514	5.00	0.0586	3.60	0.0822	3.80	0.0308	9.80	**0.0255**	**2.60**
Dermatology	0.1559	**3.33**	0.1855	4.00	0.1533	6.10	0.1407	14.15	**0.1338**	3.95
Class	0.3144	**2.00**	0.2508	2.40	0.2988	3.00	0.2608	3.33	**0.2411**	2.20
HeartEW	0.1578	2.20	**0.1512**	3.00	0.1650	**2.00**	0.1602	2.75	0.1521	2.10
Hepatitis	0.2908	3.10	0.2900	3.00	0.3305	4.11	0.2655	6.16	**0.2600**	**2.05**
Hillvalley	0.4435	27.00	0.4519	23.25	0.4657	4.00	0.4466	46.00	**0.4238**	**2.21**
Ionosphere	0.1037	4.67	0.1151	6.74	0.1943	4.00	0.0786	10.00	**0.0672**	**2.80**
Krvskp	0.0710	6.00	0.0979	5.20	0.0758	2.80	**0.0645**	9.56	0.0858	**2.60**
Lymphography	0.5519	**2.06**	0.5545	2.43	0.5784	2.60	**0.5385**	3.50	**0.5385**	2.10
Sonar	**0.1778**	14.25	0.2306	8.30	0.2862	**2.67**	0.1844	19.00	0.1815	3.33
WDBC	0.0414	4.60	0.0441	3.33	0.0950	**2.89**	0.0299	9.00	**0.0257**	3.00
Wine	0.0364	3.15	0.0527	2.80	0.0755	**2.66**	0.0347	5.10	**0.0320**	2.70
Zoo	0.1479	**2.30**	**0.1033**	**2.30**	0.1843	2.80	0.1268	2.95	**0.1033**	**2.30**

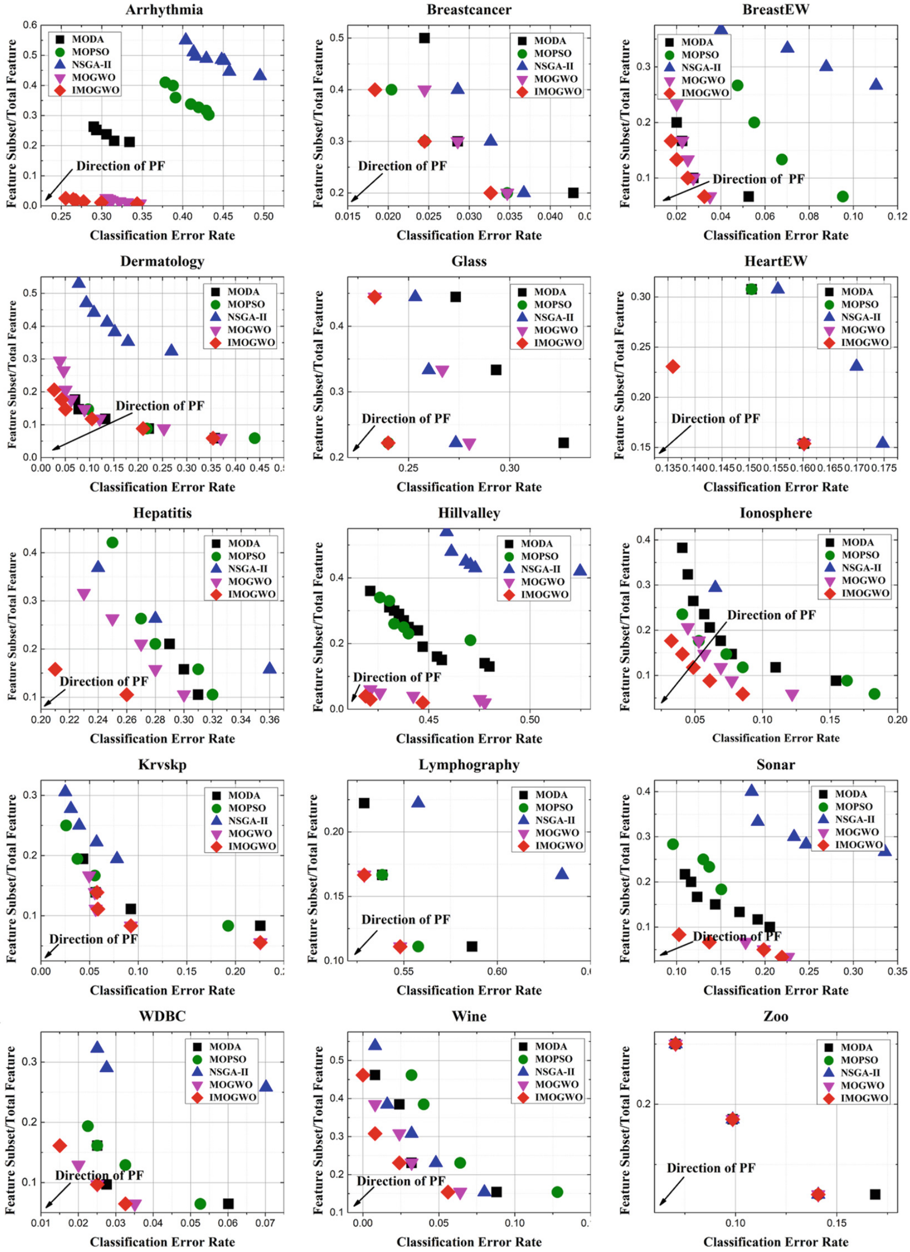

Fig. 1. Solution distribution results obtained by different algorithms on different datasets.

4.2 Feature Selection Results

In this section, the feature selection results obtained by different algorithms are introduced. Figure 1 shows the experimental results of IMOGWO on 15 datasets and compares them with MOPSO, MODA, MOGWO and NSGA-II. Each graph represents a different dataset. The name of the dataset is displayed at the top of the graph, the classification error rate is displayed on the horizontal axis, and the feature subset/total feature is displayed on the vertical axis. It can be seen from Fig. 1 that IMOGWO selected fewer features in most of the datasets and achieved a lower classification error rate. Table 1 shows the average statistics of the classification error rate and the number of selected features for each of the 10 runs based on feature selection for MOPSO, MODA, MOGWO, NSGA-II, and IMOGWO. It is evident from the table that the proposed IMOGWO has better performance on the two objectives of feature selection compared to the other four algorithms. On the whole, compared with other algorithms, IMOGWO has the best performance. Therefore, it is proved that these improvement factors can reasonably improve the performance of traditional MOGWO.

5 Conclusion

This paper studied the problem of feature selection in machine learning and then proposed IMOGWO for jointly selecting features and classifier parameters. In IMOGWO, we proposed an initialization strategy based on a combination of tent chaotic map, sinusoidal chaotic map and opposition-based learning to improve the initial solution. Then, we introduced a local search method to search for solutions near the Pareto front, improving the development capability of the algorithm. Finally, a mutation operator was proposed to enhance the diversity of the population. Experimental results showed that the algorithm has the best performance on 15 datasets, obtaining fewer features and smaller error rates compared to MOPSO, MODA, MOGWO and NSGA-II.

Acknowledgment. This study is supported in part by the National Natural Science Foundation of China (62172186, 62002133, 61872158, 61806083), in part by the Science and Technology Development Plan (International Cooperation) Project of Jilin Province (20190701019GH, 20190701002GH, 20210101183JC, 20210201072GX) and in part by the Young Science and Technology Talent Lift Project of Jilin Province (QT202013). Geng Sun is the corresponding author.

References

1. Alber, M., Buganza Tepole, A., et al.: Integrating machine learning and multiscale modeling-perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences. NPJ Digit. Med. **2**(1), 1–11 (2019)
2. Aličković, E., Subasi, A.: Breast cancer diagnosis using GA feature selection and rotation forest. Neural Comput. Appl. **28**(4), 753–763 (2017)

3. Butler, K.T., Davies, D.W., et al.: Machine learning for molecular and materials science. Nature **559**(7715), 547–555 (2018)
4. Deb, K., Pratap, A., et al.: A fast and elitist multiobjective genetic algorithm: NSGA-ii. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
5. Ghoddusi, H., Creamer, G.G., Rafizadeh, N.: Machine learning in energy economics and finance: a review. Energy Econ. **81**, 709–727 (2019)
6. Han, J., Pei, J., Kamber, M.: Data Mining: Concepts and Techniques. Elsevier, Amsterdam (2011)
7. Huda, R.K., Banka, H.: Efficient feature selection and classification algorithm based on PSO and rough sets. Neural Compu. Appl. **31**(8), 4287–4303 (2019)
8. Ji, B., Lu, X., Sun, G., et al.: Bio-inspired feature selection: an improved binary particle swarm optimization approach. IEEE Access **8**, 85989–86002 (2020)
9. Kashef, S., Nezamabadi-pour, H.: An advanced ACO algorithm for feature subset selection. Neurocomputing **147**, 271–279 (2015)
10. Li, J., Kang, H., et al.: IBDA: improved binary dragonfly algorithm with evolutionary population dynamics and adaptive crossover for feature selection. IEEE Access **8**, 108032–108051 (2020)
11. Li, Y., Song, Y., et al.: Intelligent fault diagnosis by fusing domain adversarial training and maximum mean discrepancy via ensemble learning. IEEE TII **17**(4), 2833–2841 (2020)
12. Liu, M., Zhang, S., et al.: H infinite state estimation for discrete-time chaotic systems based on a unified model. IEEE Trans. SMC (B) **42**(4), 1053–1063 (2012)
13. Lu, Z., Wang, N., et al.: IoTDeM: an IoT Big Data-oriented MapReduce performance prediction extended model in multiple edge clouds. JPDC **118**, 316–327 (2018)
14. Mafarja, M., Aljarah, I., et al.: Binary dragonfly optimization for feature selection using time-varying transfer functions. Knowl.-Based Syst. **161**, 185–204 (2018)
15. Mafarja, M., Mirjalili, S.: Whale optimization approaches for wrapper feature selection. Appl. Soft Comput. **62**, 441–453 (2018)
16. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Adv. Eng. Softw. **69**, 46–61 (2014)
17. Mirjalili, S., Saremi, S., et al.: Multi-objective Grey Wolf Optimizer: a novel algorithm for multi-criterion optimization. Expert Syst. Appl. **47**, 106–119 (2016)
18. Qiu, L., et al.: Optimal big data sharing approach for tele-health in cloud computing. In: IEEE SmartCloud, pp. 184–189 (2016)
19. Qiu, M., Cao, D., et al.: Data transfer minimization for financial derivative pricing using Monte Carlo simulation with GPU in 5G. Int. J. Commun. Syst. **29**(16), 2364–2374 (2016)
20. Qiu, M., Liu, J., et al.: A novel energy-aware fault tolerance mechanism for wireless sensor networks. In: IEEE/ACM GCC (2011)
21. Qiu, M., Xue, C., et al.: Energy minimization with soft real-time and DVS for uniprocessor and multiprocessor embedded systems. In: IEEE DATE, pp. 1–6 (2007)
22. Wu, G., Zhang, H., et al.: A decentralized approach for mining event correlations in distributed system monitoring. JPDC **73**(3), 330–340 (2013)
23. Yu, L., Liu, H.: Feature selection for high-dimensional data: a fast correlation-based filter solution. In: 20th IEEE Conference on ICML, pp. 856–863 (2003)