Differential search algorithm for biojective feature selection

HongJun Du\textsuperscript{1*}, JingJun Yao\textsuperscript{1}, Gang Sun\textsuperscript{1}, Ming Li\textsuperscript{1}, WuMing Zhou\textsuperscript{1}, HaiYan Yu\textsuperscript{1}, ChangBiao Guo\textsuperscript{1} and Shan Zhang\textsuperscript{2}

\textsuperscript{1}State Grid Liaoning Electric Power Supply CO.LTD, Shenyang, 110004, China
\textsuperscript{2}State Grid Anshan Electric Power Supply Company, Anshan, 114000, China
\*Corresponding author’s e-mail: duhj_sy@ln.sgcc.com.cn

Abstract. Feature selection (FS) is a complex optimization problem with important real-world applications. Generally, the main target of FS is to reduce the dimensionality of features and enhance the effectiveness of solving problems. Due to the population characteristics, various evolutionary algorithms (EAs) have been proposed to solve feature selection problems over the past decades. However, the majority of them only consider single-objective optimization, while many real-world problems have multiple objectives, therefore, it is need to design more suitable and effective EAs to deal with multi-objective FS. In this paper, a bi-objective FS algorithm based on differential search algorithm (DSA) is designed to solve FS problems, which minimizes the number of selected features and maximizes the classification accuracy. The results of simulation experiments and statistical analysis on 15 classification datasets compared with other four state-of-the-art EAs show that the proposed DSA algorithm can not only obtain better optimization performance, but also achieve competitive convergence accuracy.

1. Introduction

In recent years, with the development of information technologies that include machine learning and artificial intelligence huge amounts of data have been produced and collected [1]. However, in high dimensional data, there may be redundant, irrelevant and noisy features. Such features lead to adverse effects in the learning process and classification effectiveness by overextending the search space [2]. As the number of features increases, the classification accuracy decreases sharply and the time of training increases quickly [3]. By eliminating redundant or irrelevant features can effectively solve the curse of dimensionality. A feature selection (FS) algorithm can effectively reduce the dimensionality of data, shorten the training time and enhance the classification performance [4-5]. A multi-objective FS algorithm often contains two main objectives: 1) enhancing the classification precision and 2) reducing the number of features, which are potentially conflicting with each other [6]. Over the past decades, many meta-heuristics, including the evolutionary algorithms (EAs), have been proposed to solve FS problems. Compared with traditional methods, EAs need no domain knowledge and assumption about search space. Moreover, their population-based search strategy can generate a set of nondominated solutions with the trade-off between different conflicting objectives in a single run, which is particularly suitable for multi-objective optimization problems.

In this paper, a bi-objective feature selection algorithm based on differential search algorithm (DSA) is designed to solve feature selection problem, which minimizes the feature number and maximizes the classification accuracy.
2. The proposed DSA for FS

In this paper, a binary differential search algorithm is proposed for the biobjective feature selection task.

2.1. Differential search algorithm

Differential search algorithm (DSA) is a recently proposed derivative-free global heuristic algorithm for solving unconstrained optimization problems [7-8]. The DSA simulates the migration of a species of living beings.

In DSA, \( N \) represents the number of members in a superorganism, \( D \) denotes the dimension of the solved problem, and \( \text{MaxT} \) is the number of maximum iterations. Each artificial organisms \((X_i, i=1,2,\cdots,N)\) consists of many elements \((x_i^d, i=1,2,\cdots,D)\), i.e., \(X_i=(x_i^1, x_i^2, \cdots, x_i^D)\). The initial position of each individual is given by Eq.(1).

\[
x_i^d = LB^d + r_i \times (UB^d - LB^d)
\]

(1)

where \( x_i^d \) represents the \( d \)-th dimension of the \( i \)-th individual, \( LB^d \) and \( UB^d \) refer to the lower and upper boundaries of the \( d \)-th dimension, \( r_i \) is a uniformly distributed random number in \([0,1]\). After initialization, the mechanism of DSA for finding a stopover site \( S_i \) in the areas among the artificial organisms can be described by the Brownian-like random walk model. A randomly selected individual from the artificial organisms is used to generate a target vector:

\[
\text{Doner} = X_{\text{RandOrder}(i)}
\]

(2)

where the \( \text{RandOrder}() \) function randomly exchanges the order of the artificial organism members in the current population. The magnitude of changes in the positions of each individual is controlled by the \( \text{Scale} \) value, shown as Eq.(3).

\[
\text{Scale} = 2 \cdot \text{RandGm}(\cdot) \cdot r_2 \cdot (r_3 - r_4)
\]

(3)

where \( \text{RandGm}(\cdot) \) represents a gamma random number generator. \( r_2 \), \( r_3 \) and \( r_4 \) are uniformly distributed random numbers within \([0,1]\). Then, a stopover site \( (S_i) \) position is obtained by Eq.(4).

\[
S_i = X_i + \text{Scale} \times (\text{Doner} - X_i)
\]

(4)

A greedy selection operation is used to select the next stop point, shown as Eq.(5).

\[
X_i(t+1) = \begin{cases} S_i(t), & \text{if } f(S_i(t)) \leq f(X_i(t)) \\ X_i(t), & \text{if } f(S_i(t)) > f(X_i(t)) \end{cases}
\]

(5)

where \( t \) is the current iteration, \( f(\cdot) \) represents the objective function.

2.2. Binarization of features

To establish a binary DSA algorithm for solving feature selection problems, the individuals (solutions) are restricted to the binary \( \{0,1\} \) values, \( 1 \) represents a feature will be selected, while \( 0 \) denotes a feature will not be selected. Therefore, the lower boundary and upper boundary of each feature should be set to 0 and 1, respectively, i.e., \( LB^d = 0, UB^d = 1 \) for all \( d = 1,2,\cdots,D \). During population initialization, each feature is initialized by Eq.(1) and will be restricted to 0 or 1 as Eq.(6).

\[
x_i^d = \begin{cases} 1, & \text{if } x_i^d > 0.5 \\ 0, & \text{otherwise} \end{cases}
\]

(6)

At the same time, the position of each stopover site \( (S_i) \) will be updated by Eq.(7).

\[
S_i = X_i + \text{Scale} \times (\text{Doner} - X_i) > 0.5
\]

(7)
As for those features that are out of bounds after updating, the reinitialization mechanism will be used to restrict them to 0 or 1.

2.3. Methodology and fitness function
Suppose that Dataset is a dataset containing K instances with D features, a feature selection problem can be described as follows: to select \( L_f \) features (\( L_f \leq D \)) from all the D features so that some objectives, such as classification accuracy or classification error, are optimized. Because the number of the selected features determines the computational cost of a classification algorithm, it is also a key objective function [9]. Therefore, this paper will consider the following two objectives: maximizing the classification accuracy (\( ACC_{\text{max}} \)) and minimizing the number of the selected features (\( L_f \)). In order to unify the two objectives into maximization and create a single objective fitness function which combines the two objectives into one, the following fitness function is modelled by Eq.(8).

\[
\text{Fitness} = ACC_{\text{max}} + w_f \cdot \left(1 - \frac{L_f}{D}\right)
\]  

(8)

In this paper, the whole dataset was randomly divided into training set and test set by 5-fold cross-validation method. The \( k \)-nearest neighbor (\( k \)-NN) classifier [10] is adopted, and \( k = 3 \). In Eq.(8), \( ACC \) is the classification accuracy calculated by dividing the number of correctly classified instances by the total number of instances. \( ACC_{\text{max}} \) represents the maximum \( ACC \) among the 5-fold cross-validation. \( w_f \in (0,1) \) denotes the weight factor for the number of selected features. In this paper, \( w_f \) will be set to 0.8.

3. Experimental results and analysis

3.1. Comparison algorithms and related parameter settings
To evaluate the performance of the proposed DSA for solving FS problems, 5-fold cross-validation is employed. Based on 5-fold cross-validation, 80% of training partition is used for training, while 20% is used for the fitness function validation [11]. The classification accuracy of the \( k \)-NN classifier with \( k = 3 \) and the number of selected features are used to establish the fitness function. The performance is compared with four classical FS algorithms including particle swarm optimization (PSO) [12], cuckoo search [13], butterfly optimization algorithm (BOA) [14], pathfinder algorithm (PFA) [15]. These comparison algorithms, as detailed in Table 1. The related parameter settings for all comparison algorithms are also listed in Table 1, which are based on the suggestions in the corresponding references.

For the sake of fairness, each algorithm is independently run 30 times, the population size \( N = 30 \), and the terminal condition of all algorithms is the maximum function evaluations (MaxFEs) which is set to 3000.

| Algorithm | Year | Parameter informations |
|-----------|------|-------------------------|
| PSO       | 1995 | \( \omega = 0.9 \sim 0.4, c_1 = c_2 = 2 \) |
| CS        | 2009 | Discovery rate of alien eggs: \( pa = 0.25 \) |
| BOA       | 2018 | Sensor modality: \( c = 0.01 \), Power exponent: \( 0.1 \sim 0.3 \), Switch probability: \( p = 0.8 \) |
| PFA       | 2019 | \( \alpha = [LB,UB], \beta = [LB,UB] \) |
| DSA       | presented | Method = 1, \( p_1 = 0.3 \text{rand}, p_2 = 0.3 \text{rand}, R = 1. / \text{gamrnd}(1,0.5) \) |

3.2. Datasets
To verify the performance of the proposed DSA, we carried out a comparison over 15 benchmark datasets from the UCI machine learning datasets repository (http://archive.ics.uci.edu/ml/index.php),
which has been employed extensively in numerous model- and algorithm- evaluation studies. The detailed information is listed in Table 2. As Table 2 shows, the features of all the datasets ranges from 5 to 60, and the datasets contain the binary and multi-class classification datasets.

3.3. Experimental results and analysis
To evaluate the performance of all algorithms for solving FS problems, six quantitative indicators will be adopted, including the average (Mean), standard deviation (Std), worst value (Worst), rank of each algorithm on every dataset (Rank), total rank and final rank of each algorithm on all the datasets. The Mean records the average of 30 runs independently. The larger the Mean, the better the performance. Std is a measure of the dispersion of a set of data averages. The smaller the Std, the more stable the performance. Rank is sorted according to the Mean value, the best performance, i.e., the maximum Mean is recorded 1, the second performance is recorded 2, and so on. Total rank is the cumulative sum of Rank on all datasets. The smaller total rank, the better the overall performance in all datasets. Table 3 shows the comparison results of all algorithms on the selected datasets. According to the principle of no free lunch (NFL), there is no a algorithm can obtain the best performance on all datasets.

As Table 3 shows, the best Mean and Rank are shown in bold. Among the 15 datasets, the proposed DSA ranks first on 9 datasets, including D1, D2, D5, D8 - D12 and D15. DSA ranks second on 3 datasets, i.e., D3, D4 and D7. The rest datasets, DSA ranks third. DSA has a total rank 24 and final rank first on all the 15 datasets. On dataset D1, all algorithms have the same performance. As for final rank, BOA ranks second, PSO ranks third, PFA ranks fourth, and CS ranks fifth. In summary, DSA has competitive performance on quantitative analysis among 15 datasets.

3.4. Convergence analysis
In order to intuitively compare the convergence performance of all algorithms, convergence curves of all algorithms on different datasets are shown in Figure1. It can be seen from Figure1, DSA obtains better performance on both convergence accuracy and rate under different types of datasets.

3.5. Statistical analysis
Statistical analysis of the experimental results of the proposed algorithm is an important and meaningful work [16]. For pairwise comparison of the problem-solving success of evolutionary algorithms, a
problem-based or multi-problem-based statistical comparison method can be used [17]. In this paper, the Wilcoxon Signed-Rank Test (WSRT) is used to perform a pairwise statistical test on the experimental results from Table 3, with the significance level $\alpha = 0.05$. WSRT is a non-parametric test.

Table 3. Experimental results of DSA and other algorithms.

| Algorithm | DSA | PFA | CS | BOA | PSO |
|-----------|-----|-----|----|-----|-----|
| D1 Mean   | 1.5333 | 1.5333 | 1.5333 | 1.5333 | 1.5333 |
| Std       | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| Worst     | 1.5333 | 1.5333 | 1.5333 | 1.5333 | 1.5333 |
| Rank      | 1 1 1 1 1 |

| D2 Mean   | 1.4839 | 1.4561 | 1.4229 | 1.4699 | 1.4678 |
| Std       | 0.0333 | 0.0419 | 0.0413 | 0.0208 | 0.0431 |
| Worst     | 1.4500 | 1.4000 | 1.3895 | 1.4571 | 1.4095 |
| Rank      | 1 4 5 3 2 |

| D3 Mean   | 1.3619 | 1.3494 | 1.3403 | 1.3632 | 1.3533 |
| Std       | 0.0114 | 0.0083 | 0.0140 | 0.0059 | 0.0079 |
| Worst     | 1.3533 | 1.3387 | 1.3133 | 1.3521 | 1.3467 |
| Rank      | 2 4 5 1 3 |

| D4 Mean   | 1.3747 | 1.3426 | 1.3676 | 1.4205 | 1.3714 |
| Std       | 0.0115 | 0.0601 | 0.0559 | 0.0310 | 0.0305 |
| Worst     | 1.3500 | 1.2667 | 1.2652 | 1.4083 | 1.2957 |
| Rank      | 2 5 4 1 3 |

| Total Rank | 24 | 54 | 63 | 34 | 40 |
| Final Rank | 1 4 5 2 3 |
that can be employed to test which algorithm has better statistical performance when compared between different datasets. The WSRT results are listed in Table 4. '+' denotes that DSA outperforms the compared algorithm, '=' represents that there is no significant statistical difference between two algorithms, and '-' indicates that DSA shows poorer performance in comparison. The last row shows the total count in the +/=/- format. As Table 4 shows, the performance of DSA is significantly better than the other compared algorithms, especially compared with CS and PFA.

![Figure 1. Convergence curve of algorithms.](image)
4. Conclusion

A feature selection (FS) algorithm can effectively reduce the dimensionality of data, shorten the training time and enhance the classification performance, so that the research on feature selection algorithms is a hot topic in the field of machine learning. In this paper, a binary DSA is proposed to solving biobjective feature selection problems. Firstly, the initial population is randomly generated in binary search space. Then, the binary mechanism is also used for position update and out-of-bounds processing. Meanwhile, maximizing classification accuracy and minimizing the number of selected features are combined a unified fitness function. The proposed DSA is compared with PSO, CS, BOA and PFA using 15 UCI datasets. The experimental and statistical results show the proposed DSA outperforms other algorithms and has better convergence ability. As a future work, the proposed binary mechanism and fitness function will be employed to other swarm based evolutionary algorithms. We will also evaluate the performance of the proposed DSA on complex high dimensional datasets and real-world optimization problems.

References

[1] Nouri-Moghaddam B., Ghazanfari M., Fathian M. (2021) A novel multi-objective forest optimization algorithm for wrapper feature selection. Expert Systems with Applications, 175: 114737, 2021.

[2] Xue B., Zhang M., Browne W.N, Yao X. (2016) A Survey on Evolutionary Computation Approaches to Feature Selection. IEEE Transactions on Evolutionary Computation, 20(4): 606-626.

[3] Han F., Chen W.T., Ling Q.H., Han H. (2021) Multi-objective particle swarm optimization with adaptive strategies for feature selection. Swarm and Evolutionary Computation, 62: 100847.

[4] Gheyas I.A., Smith L.S. (2010) Feature subset selection in large dimensionality domains, Pattern Recognition, 43(1): 5-13.

[5] Zhang Y., Gong D., Gao X., Sun X. (2020) Binary differential evolution with self-learning for multi-objective feature selection. Information Sciences, 507: 67-85.

[6] Nguyen B.H., Xue B., Andreae P., Ishibuchi H., Zhang M. (2020) Multiple Reference Points-Based Decomposition for Multiobjective Feature Selection in Classification: Static and Dynamic Mechanisms. IEEE Transactions on Evolutionary Computation, 24(1): 170–184.

[7] Civicioglu P. (2012) Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm. Computers & Geosciences, 46(0): 229-247.

[8] Liu J., Teo K.L., Wang X., Wu C. (2016) An exact penalty function-based differential search algorithm for constrained global optimization. Soft Computing, 20(4): 1305-1313.
[9] Zhang Y., Gong D., Cheng J. (2017) Multi-Objective Particle Swarm Optimization Approach for Cost-Based Feature Selection in Classification. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 14(1): 64-75.

[10] Altman N.S. (1992) An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. The American Statistician, 46(3): 175-185.

[11] Kelidari M., Hamidzadeh J. (2021) Feature selection by using chaotic cuckoo optimization algorithm with levy flight, opposition-based learning and disruption operator. Soft Computing, 25(4): 2911-2933.

[12] Kennedy J., Eberhart R. (1995) Particle swarm optimization. In: IEEE International Conference on Neural Networks. Perth. pp. 1942-1948.

[13] Yang X.S., Deb S. (2009) Cuckoo search via Levy flights. In: World Congress on Nature & Biologically Inspired Computing. Coimbatore. pp. 210-214.

[14] Arora S., Singh S. (2019) Butterfly optimization algorithm: a novel approach for global optimization. Soft Computing, 23(3): 715-734.

[15] Yapici H., Cetinkaya N. (2019) A new meta-heuristic optimizer: Pathfinder algorithm. Applied Soft Computing, 78: 545-568.

[16] Garcia S., Fernandez A., Luengo J., Herrera F. (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences, 180: 2044-2064.

[17] Derrac J., Garcia S., Molina D., Herrera F. (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation, 1: 3-18.