IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received July 7, 2019, accepted August 3, 2019, date of publication August 16, 2019, date of current version August 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2935833

Feature Selection Using an Improved Gravitational Search Algorithm

LEI ZHU, SHOUSHUAI HE, LEI WANG, WEIJUN ZENG, AND JIAN YANG
College of Communication Engineering, Army Engineering University of PLA, Nanjing 210007, China

Corresponding authors: Shoushuai He (frldh@outlook.com) and Lei Wang (iponly@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 71501186 and Grant 61702543, and in part by the 333 High-Level Talent Training Project of Jiangsu Province of China under Grant BRA 2016542.

ABSTRACT Feature selection is an important issue in the field of machine learning, which can reduce misleading computations and improve classification performance. Generally, feature selection can be considered as a binary optimization problem. Gravitational Search Algorithm (GSA) is a population-based heuristic algorithm inspired by Newton's laws of gravity and motion. Although GSA shows good performance in solving optimization problems, it has a shortcoming of premature convergence. In this paper, the concept of global memory is introduced and the definition of exponential *Kbest* is used in an improved version of GSA called IGSA. In this algorithm, the position of the optimal solution obtained so far is memorized, which can effectively prevent particles from gathering together and moving slowly. In this way, the exploitation ability of the algorithm gets improved, and a proper balance between exploration and exploitation gets established. Besides, the exponential *Kbest* can significantly decrease the running time. In order to solve feature selection problem, a binary IGSA (BIGSA) is further introduced. The proposed algorithm is tested on a set of standard datasets and compared with other algorithms. The experimental results confirm the high efficiency of BIGSA for feature selection.

INDEX TERMS Feature selection, gravitational search algorithm, classification.

I. INTRODUCTION

Machine learning has been widely employed in many technology and engineering fields, such as intrusion detection [1], pattern recognition [2], image analysis, text categorization [3], data mining [4] and multimedia information retrieval [5]. These fields often involve datasets that contain large number of features. Some of the features may be irrelevant or misleading, which increase computational costs and even reduce classification accuracy. Therefore, feature selection is critical to analyze high-dimensional datasets [6].

Feature selection is the process that finding the optimal feature subset from original feature set. Features in the selected subset should be informative and discriminating. In other words, the purpose of feature selection is to decrease the number of features used to describe the dataset. By eliminating redundant noise, feature selection helps to reduce storage requirement and avoid over-fitting problem.

According to the evaluation modes for feature subsets, feature selection techniques are categorized into filter, wrapper and embedded methods. Filter methods use the intrinsic characteristics of dataset to evaluate features, so feature selection is independent of classifiers [7]. However, wrapper methods use classifiers to evaluate candidate subsets obtained by search algorithms, and the feedback from classifiers is helpful for feature selection [8]. They guarantee classification accuracy through complex computation. Thus, filter methods are efficient in computation, while wrapper methods are superior in performance. Embedded methods can be considered as a special kind of wrapper methods, in which feature selection is a part of training phase in machine learning and specific to learning algorithms [9].

In general, feature selection can be regarded as an optimization problem. Each state in the search space represents a candidate subset. The search strategies for finding the optimal feature subset include exhaustive search and heuristic search. However, the exhaustive evaluation for all feature subsets requires much computational costs. It is not feasible in practice, especially when the number of features is large. Hence, heuristic algorithms are often applied to find the approximate optimal feature subset. Several methods have been proposed to solve feature selection problem

The associate editor coordinating the review of this article and approving it for publication was Hongwei Du.

using evolutionary techniques, such as Particle Swarm Optimization [10], Ant Colony Optimization [11] and Genetic Algorithm [12].

Gravitational Search Algorithm is one of the latest evolutionary algorithms, which is inspired by Newton's laws of gravity and motion [13]. In this algorithm, the search agents are a group of interacting particles. It has been proved that GSA has some merits compared with other famous heuristic algorithms in optimization field. However, there is a major issue with the search performance of GSA. Due to the rapid reduction in diversity, the original algorithm suffers from premature convergence. And it is difficult to get a good balance between exploration and exploitation.

Some efforts have been made to overcome the disadvantages. In [14], a novel attractive-repulsive gravitational search algorithm is proposed, in which the uniform circular motion and centripetal force are introduced. A chaotic optimization mechanism making the parameter of the algorithm change chaotically is applied in [15]. A hybrid particle swarm optimization and gravitational search algorithm is utilized in [16] to solve binary optimization problems. And in our previous work [17], a modified gravitational search algorithm is proposed, which improves the exploration ability of the algorithm and provides good performance for function optimization.

Since the original algorithm is memoryless, it converges slowly in exploitation stage. Therefore, in this paper, a new version of GSA based on global memory is proposed to enhance the exploitation ability in late iterations and establish a proper balance between exploration and exploitation. Simultaneously, the exponential Kbest further balances the exploration and exploitation, and significantly improves the computational efficiency of the algorithm.

Feature selection problem is often set in binary space [18], while IGSA operates in continuous space. Thus, a binary form of IGSA is further introduced, which refers to the original binary GSA [19]. Based on BIGSA, a new feature selection method is proposed, in which the K-Nearest Neighbor (K-NN) technique is used as a classifier to perform wrapper method and evaluate candidate subsets. The classification accuracy and number of selected features, as the feedbacks to BIGSA, lead the algorithm to find the optimal feature subset.

This paper is organized as follows. In the next section, the principle of GSA is briefly reviewed. In Section III, the proposed IGSA and its characteristics are described in detail. In Section IV, the feature selection problem is modeled, and the binary version of IGSA is introduced. In Section V, the performance of BIGSA is evaluated by a set of standard datasets and compared with the original algorithm and another modified algorithm. This paper is concluded in Section VI.

II. GRAVITATIONAL SEARCH ALGORITHM

In this section, the original Gravitational Search Algorithm is introduced, which is a newly developed heuristic algorithm used to solve optimization problems. It simulates the

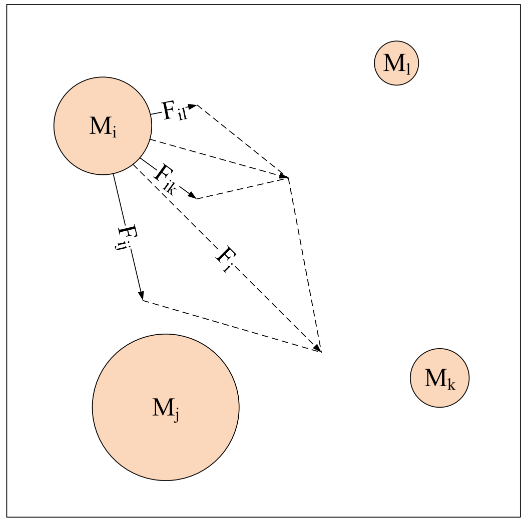


FIGURE 1. Principle of GSA.

interaction between particles under the effect of gravity. They move in the multidimensional search space to find the optimal solution.

GSA is constructed according to the law of gravity as depicted in Fig. 1. Particles attract each other by gravitational force, the force between two particles is directly proportional to their masses and inversely proportional to the distance between them. Hence, the gravitational force between heavy and close particles is large.

As iterations go on, the attraction of gravity causes all particles to move toward heavier particles. The global movement of population is the exploration stage of the algorithm. And the exploitation stage is accomplished by the movement of heavier particles, as they are closer to the optimal solution and move more slowly than lighter particles.

The performance of each particle is measured by its mass, which is evaluated by a fitness function. In this algorithm, the gravitational and inertial masses of each particle are equal and updated by the following equations:

$$M_{ai} = M_{pi} = M_{ii} = M_i, \tag{1}$$

$$m_i = \frac{fit_i - worst}{best - worst}, \tag{2}$$

$$M_i = \frac{m_i}{\sum_{j=1}^{N} m_j}, \tag{3}$$

where fit_i represents the fitness value of particle i, $best$ and $worst$ represent the best and worst fitness values among all particles, respectively. For maximization problems, they are defined as follows:

$$best = \max_{j\in\{1,\ldots,N\}} fit_j, \tag{4}$$

$$worst = \min_{j\in\{1,\ldots,N\}} fit_j. \tag{5}$$

But for minimization problems, they are quite opposite:

$$best = \min_{j\in\{1,\ldots,N\}} fit_j, \tag{6}$$

$$worst = \max_{j\in\{1,\ldots,N\}} fit_j. \tag{7}$$

At the beginning of the algorithm, a population containing N particles is randomly created. The position of each particle is considered as a candidate solution and defined as the following form:

$$x_i = (x_i^1, \ldots, x_i^d, \ldots, x_i^D), \quad i = 1, 2, \ldots, N, \tag{8}$$

where x_i^d represents the position of particle i in dimension d, and D represents the number of dimensions.

The gravitational force acting on particle i from particle j in dimension d is calculated as follows:

$$F_{ij}^d = G \times \frac{M_{pi} \times M_{aj}}{R_{ij} + \varepsilon} \times (x_j^d - x_i^d), \tag{9}$$

where M_{pi} is the passive gravitational mass of particle i, M_{aj} is the active gravitational mass of particle j, R_{ij} is the Euclidean distance between two particles, and ε is a small constant. G is named as the gravitational constant, but it is a function of iterations:

$$G = G_0 \times e^{-\alpha \frac{t}{T}}, \tag{10}$$

where G_0 is the initial value, α is a shrinking constant, and T is the total number of iterations.

The total force acting on particle i in dimension d is a randomly weighted sum of gravitational force exerted from other particles:

$$F_i^d = \sum_{j \in Kbest, j \neq i} rand_j \times F_{ij}^d, \tag{11}$$

where $rand_j$ is a uniform random variable within interval [0, 1]. In order to avoid falling into the local optimum and control the balance between exploration and exploitation, only $Kbest$ particles with the highest fitness values are employed to exert gravitational force on others. $Kbest$ is initialized to the population size at the beginning and linearly reduced to 1 with iterations:

$$Kbest = N \times \frac{per + \left(1 - \frac{t}{T}\right) \times (100 - per)}{100}, \tag{12}$$

where per represents the percent of particles that exert force on other particles in the end.

Based on the law of motion, the acceleration of particle i in dimension d is calculated as follows:

$$a_i^d = \frac{F_i^d}{M_{ii}}, \tag{13}$$

where M_{ii} is the inertial mass of particle i.

The next velocity of particle i in dimension d is updated according to its current velocity and acceleration:

$$v_i^d = rand_i \times v_i^d + a_i^d, \tag{14}$$

where $rand_i$ is a uniform random variable within interval [0, 1], which is applied to provide a random characteristic for the search.

Then, the next position of particle i in dimension d is updated as the following equation:

$$x_i^d = x_i^d + v_i^d. \tag{15}$$

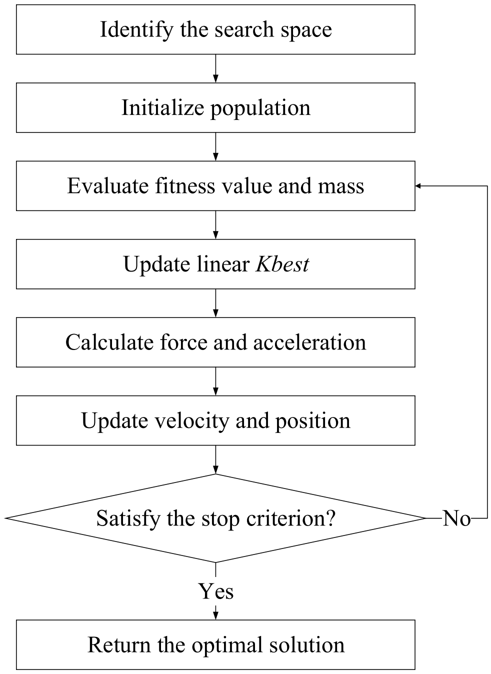


FIGURE 2. **Process of GSA.**

As shown in Fig. 2, the above steps will be repeated until the total number of iterations is reached or an acceptable solution is found.

III. IMPROVED GRAVITATIONAL SEARCH ALGORITHM

Finding the approximate optimal solution within reasonable running time is the prime goal of heuristic algorithms. One way to accomplish this goal is to provide a proper balance between exploration and exploitation. In this section, a new version of GSA is proposed to improve the exploitation ability and computational efficiency of the algorithm.

A. GLOBAL MEMORY

In the original algorithm, particles explore the search space in early iterations, then find the global optimal solution among good solutions in late iterations. As iterations go on, particles get closer to good solutions, they become heavier and move more slowly. For this reason, the convergence rate of the algorithm decreases in exploitation stage. Besides, the original algorithm has no memory to preserve the optimal solution obtained so far, so it may be lost as the heaviest particle with the best fitness value is attracted by other particles.

In order to overcome these disadvantages, a global memory called $gbest$ is introduced in the proposed algorithm, which memorizes the optimal solution obtained so far. All particles can observe its position and move toward it. This strategy is presented in Fig. 3. It depicts a simple one-dimensional minimization problem, where fit denotes the fitness function whose minimum value is 0. As shown in the figure, particles j and k are attracted by particle i, while these two particles also attract particle i and make it slightly deviate from the global optimal solution. In other words, when heavy particles

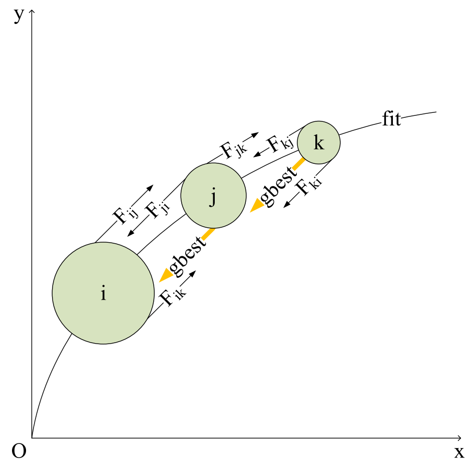


FIGURE 3. Effect of global memory.

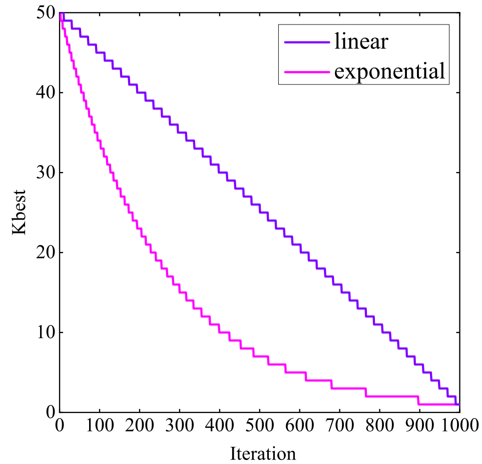


FIGURE 4. Reduced processes of exponential *Kbest* and linear *Kbest*.

are close to the global optimal solution, they may not be able to move toward it, but instead move toward the centroid of particles in the neighborhood. Thus, the global memory *gbest* is applied to prevent particles from stagnating in suboptimal situations. In this way, the movement of heavy particles gets enhanced, and the exploitation ability of the algorithm gets improved.

As a sort of social intelligence, *gbest* provides each particle with an additional velocity component toward the position of the heaviest particle. Hence, the equation of velocity updating is modified as follows:

$$v_i^d = rand_i \times v_i^d + c_1 \times a_i^d + (2 - c_1) \times (x_{gbest}^d - x_i^d), \quad (16)$$

where c_1 is an acceleration coefficient, and x_{gbest}^d is the position of *gbest* in dimension d. It can be seen that the front of the equation is similar to the original algorithm, which aims to maintain the exploration ability of the algorithm. And the rear of the equation is added to guide particles toward the optimal solution obtained so far, which helps particles to approach the global optimal solution in exploitation stage. Since there is no clear boundary between exploration and exploitation stages in heuristic algorithms, c_1 is crucial for balancing them [20]. It is reduced with iterations as the following equation:

$$c_1 = 2 - 2 \times \frac{t^3}{T^3}, \quad (17)$$

where t denotes the current iteration, and T denotes the total number of iterations.

The effect of *gbest* on particles is independent of their masses and is not constrained by the low of gravity, which effectively prevents particles from gathering together and moving slowly. Such modification maintains exploration ability in early iterations, enhances exploitation ability in late iterations and provides a proper transition between them.

B. EXPONENTIAL KBEST

In order to improve the computational efficiency of the algorithm and enhance the balance between exploration and exploitation, the definition of *Kbest* is modified in this paper.

In the original algorithm, *Kbest* is the number of particles that exert force on all particles. It is a function of iterations and linearly reduced from the population size to 1. But in the proposed algorithm, *Kbest* is modified and exponentially reduced from the population size to 1 with iterations as the following equation:

$$Kbest = N \times \left(\frac{per}{100}\right)^{\frac{t}{T}}, \quad (18)$$

where N represents the population size, *per* represents the percent of particles that exert force on other particles in the end. The reduced processes of the exponential *Kbest* and linear *Kbest* are shown in Fig. 4, where N is 50, *per* is 2, and T is 1000.

Neglecting the inconsequential particles with small masses can effectively decrease the redundant computations then improve the computational efficiency. Simultaneously, such modification makes the algorithm do comprehensive exploration in early iterations and precise exploitation in late iterations [21]. So the exploration and exploitation abilities of the algorithm get further balanced.

Besides, other equations in the proposed algorithm such as force and acceleration calculations, velocity and position updatings are the same as the original algorithm.

In summary, the detailed process of IGSA is listed as Algorithm 1.

The time complexity of the proposed algorithm is analyzed as follows:

- Population initialization requires $O(N \times D)$, where D represents the dimension of search space.
- Fitness value evaluation requires $O(N \times D \times T)$.
- Mass calculation requires $O(N \times T)$.
- Global memory and exponential *Kbest* updatings require $O(T)$ each.
- Force and acceleration calculations approximately require $O(N^2 \times D \times T)$ each, when *Kbest* is exponentially reduced with iterations and T is large enough.
- Velocity and position updatings require $O(N \times D \times T)$ each.

TABLE 1. Standard datasets.

Dataset	Name	Number of features	Number of instances	Number of classes
D_1	Wine	13	178	3
D_2	Zoo	16	101	7
D_3	Image Segmentation	19	2310	7
D_4	Ionosphere	34	351	2
D_5	Landsat Satellite	36	6435	6
D_6	Spambase	57	4601	2
D_7	Sonar	60	208	2
D_8	Libras Movement	90	360	15

Algorithm 1 Improved Gravitational Search Algorithm

Input:
  N: population size
  T: total number of iterations
Output:
  S: optimal solution

```
Identify the search space
Initialize population
while the stop criterion is not reached do
    Evaluate the fitness value of each particle by the fitness function
    Calculate the mass of each particle by Eq. (3)
    Update global memory and exponential Kbest
    Calculate the force and acceleration of each particle in different dimensions by Eq. (11) and (13)
    Update the velocity and position of each particle in different dimensions by Eq. (16) and (15)
end while
return S
```

Considering the complexities of the above steps, the total time complexity of IGSA is $O(N^2 \times D \times T)$, which is equal to the original algorithm.

The space requirement of the proposed algorithm is related to the population size and dimension of search space. Therefore, the total space complexity of IGSA is $O(N \times D)$, which is also equal to the original algorithm.

In IGSA, the position of the optimal solution obtained so far is memorized, and the parameter *Kbest* is exponentially reduced with iterations. All particles move in the search space under the additional effect of the global memory and exponential *Kbest*. In this way, the search ability and efficiency of the algorithm get significant improvements.

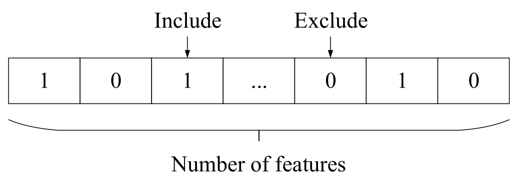


FIGURE 5. Binary vector of a candidate subset.

IV. FEATURE SELECTION PROBLEM AND ITS SOLUTION

In this section, a binary version of IGSA is further introduced, and a new feature selection technique based on the wrapper method is proposed, in which the K-Nearest Neighbor algorithm is used as a classifier to evaluate classification accuracy.

A. FEATURE SELECTION MODEL

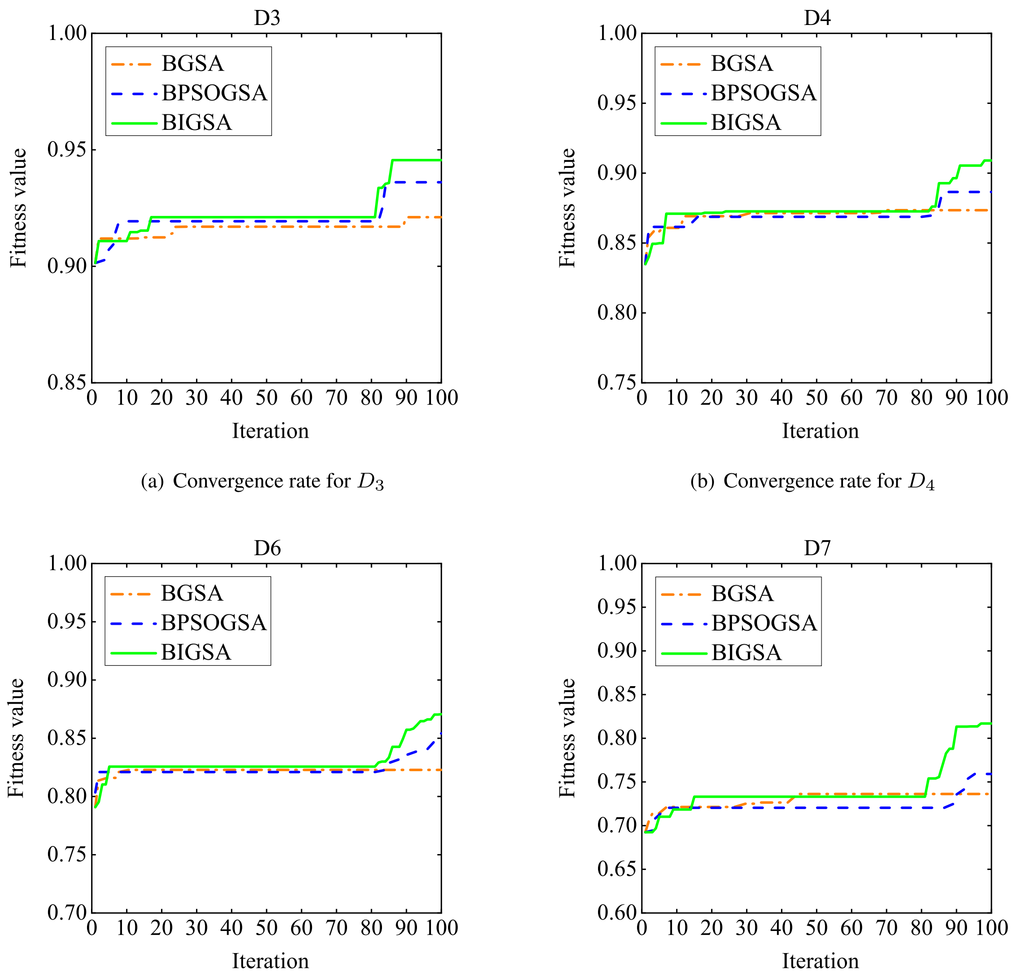
Feature selection is one of the most active issues in the field of machine learning. The task of feature selection is to reduce the number of features used to characterize datasets and improve the performance of classifiers.

Generally, feature selection can be expressed as an optimization problem, and each candidate subset can be represented by a binary vector as shown in Fig. 5.

Each bit of the vector takes a value of 1 or 0 to indicate whether the corresponding feature exists in the candidate subset. If the value of the i-th bit is 1, the i-th feature will be included in the subset. Otherwise, the feature will be excluded [22]. The length of the vector is equal to the number of all features.

B. BINARY IGSA

Since IGSA operates in continuous space while feature selection problem is set in binary space, a binary IGSA is further introduced, which refers to the original BGSA proposed in [19].



(a) Convergence rate for D_3 (b) Convergence rate for D_4

(c) Convergence rate for D_6 (d) Convergence rate for D_7

FIGURE 6. Convergence rate.

As mentioned earlier, the position of each particle corresponds to a candidate subset. Thus, it is defined as a binary vector. The value of each dimension is 1 or 0, which indicates the corresponding feature is included or excluded, respectively. At the beginning of BIGSA, the position of each particle is randomly initialized to a series of binary values. Moving in a dimension means that the value changes from 1 to 0 or vice versa.

The velocity of each particle in BIGSA is updated as IGSA. And a probability function that transforms the velocity into probability is defined as follows:

$$S(v_i^d) = \left|\tanh(v_i^d)\right|, \tag{19}$$

where $S(v_i^d)$ is bounded by interval [0, 1]. The probability will increase as the velocity increases.

In BIGSA, the trajectory of a particle is determined by the changes in probability. The position of each particle is updated according to the calculated probability and the following rule:

$$\begin{aligned} & \textit{if } r\ and\ < S(v_i^d) \textit{ then} \\ & \quad x_i^d = complement(x_i^d) \\ & \textit{else} \\ & \quad x_i^d = x_i^d \end{aligned} \tag{20}$$

where *rand* is a uniformly distributed random number in interval [0, 1].

C. FITNESS FUNCTION

The fitness function for evaluating candidate subsets should be designed based on classification accuracy and the number of selected features [23]. In order to solve this dual-objective problem, a fitness function is constructed and needs to be maximized:

$$fit = w \times acc + (1 - w) \times \frac{(tot - sel)}{tot}, \tag{21}$$

where *acc* denotes the classification accuracy obtained by classifier, *tot* denotes the total number of features in dataset,

Algorithm 2 BIGSA for Feature Selection

Input:
  D: dataset
  N: population size
  T: total number of iterations
Output:
  S: optimal feature subset

Identify the feature dimension of dataset
Initialize feature subsets as population
while the stop criterion is not reached **do**
  Train the K-NN algorithm as the classifier
  Evaluate fitness value and mass
  Update global memory and exponential *Kbest*
  Calculate force and acceleration
  Update velocity and position
end while
return S

sel denotes the number of selected features in candidate subset, and w is a weighting factor that controls the relative importance of classification accuracy and feature reduction proportion.

A better fitness value means that the candidate subset has higher classification accuracy and fewer selected features.

D. K-NEAREST NEIGHBOR ALGORITHM

In the experiment, the K-Nearest Neighbor algorithm is applied to implement wrapper method and evaluate classification accuracy [24].

K-NN is one of the most popular classification algorithms, which classifies samples by calculating the minimum distance between them. The parameter K denotes the number of the nearest neighbors, which is the key to the classification performance. Since only one parameter needs to be determined, it is convenient to use K-NN for classification. If a sample is close to the K nearest neighbors, it will be classified into the corresponding category.

In this paper, BIGSA is used as a search algorithm to find the optimal feature subset. The detailed process of feature selection is summarized as Algorithm 2.

V. EXPERIMENTAL RESULTS

Satisfactory solution and reasonable computational costs are two important criteria of heuristic algorithms for solving optimization problems [25]–[27]. In this section, in order to research the performance of BIGSA for feature selection, it is compared with the original algorithm and another modified algorithm: binary hybrid particle swarm optimization and gravitational search algorithm (BPSOGSA) [16], which is a classical modified version of GSA.

The existing and proposed algorithms are tested on a set of standard datasets from the machine learning repository of UCI as presented in Table 1 [28]–[30], followed by the numbers of features, instances and classes of each dataset.

TABLE 2. Parameter settings.

Parameter	BGSA	BPSOGSA	BIGSA
N	20	20	20
T	100	100	100
G_0	100	1	100
$Kbest$	linear	N	exponential

TABLE 3. Fitness values.

Dataset	BGSA	BPSOGSA	BIGSA
D_1	0.9093	0.9104	**0.9206**
D_2	0.9196	0.9197	**0.9295**
D_3	0.9321	0.9357	**0.9422**
D_4	0.8823	0.9061	**0.9106**
D_5	0.8376	0.8446	**0.8456**
D_6	0.8308	0.8543	**0.8641**
D_7	0.7563	0.7790	**0.7894**
D_8	0.8148	0.8379	**0.8450**

TABLE 4. Classification accuracy.

Dataset	BGSA	BPSOGSA	BIGSA
D_1	95.39%	95.72%	**96.04%**
D_2	97.45%	97.15%	**98.06%**
D_3	96.65%	96.96%	**96.99%**
D_4	91.61%	92.83%	**93.31%**
D_5	85.67%	**85.71%**	85.62%
D_6	88.27%	88.19%	**88.68%**
D_7	78.82%	81.79%	**82.43%**
D_8	84.94%	86.04%	**86.46%**

And 10-fold cross validation is used in the test to avoid overfitting problem.

In all experiments, w is set to 0.8, the parameter K of K-NN is set to 1, the population size is set to 20, and the total number of iterations is set to 100. In BGSA, *Kbest* is linearly reduced from N to 1 with iterations. In BPSOGSA,

TABLE 5. Numbers of selected features.

Dataset	BGSA	BPSOGSA	BIGSA
D_1	3.5	3.6	**3.1**
D_2	4.8	4.6	**4.4**
D_3	3.9	3.8	**3.2**
D_4	8.6	6.2	**6.1**
D_5	8.6	7.4	**7.1**
D_6	21.5	14.6	**12.9**
D_7	22.3	22.6	**21.0**
D_8	29.1	22.7	**21.0**

TABLE 6. Running time (in seconds).

Dataset	BGSA	BPSOGSA	BIGSA
D_1	29.66	30.84	**29.45**
D_2	30.52	30.90	**29.97**
D_3	86.87	79.27	**73.44**
D_4	36.62	36.76	**35.79**
D_5	468.51	431.39	**386.21**
D_6	996.90	948.03	**923.36**
D_7	36.21	36.38	**35.87**
D_8	52.49	52.83	**52.15**

Kbest is considered as the population size. And in BIGSA, G_0 is set to 100, α is set to 20, *per* is set to 2, and *Kbest* is exponentially reduced from N to 1 with iterations. The detailed parameter settings of three algorithms are summarized in Table 2.

The fitness values, classification accuracy, numbers of selected features and running time listed in Table 3-6 are the average of 10 independent runs. The best results of each standard dataset are presented in the bold form.

A. CLASSIFICATION ACCURACY AND NUMBERS OF SELECTED FEATURES

The global memory prevents the proposed algorithm from falling into the local optimum. Table 3-5 show that BIGSA obtains better fitness values, higher classification accuracy and less numbers of selected features compared with other algorithms in most of the standard datasets, except for D_5, in which BPSOGSA provides slightly better classification result.

B. RUNNING TIME

The exponential *Kbest* neglects the force exerted from the inconsequential particles with small masses. Hence, it can improve the computational efficiency of the algorithm. It is clear from Table 6 that BIGSA has shorter running time than other algorithms in all cases, especially for the datasets with large number of instances.

C. CONVERGENCE RATE

Under the additional effect of the global memory and exponential *Kbest*, BIGSA provides better results and faster convergence rate than others in late iterations. The convergence trends of three algorithms for part of representative standard datasets are shown in Fig. 6. This figure confirms the high performance of BIGSA for feature selection.

VI. CONCLUSION

In recent years, the demand for heuristic algorithms to solve feature selection problem has increased dramatically. As one of the newly heuristic algorithms, GSA is constructed based on Newton's laws of gravity and motion. In this paper, a new version of GSA called IGSA is proposed. In this algorithm, the concept of global memory is introduced to memorize the position of the optimal solution obtained so far. Particles move in the search space under the additional effect of the global memory. In this way, the exploitation ability of the algorithm gets enhanced, and a proper balance between exploration and exploitation gets established. Simultaneously, the exponential *Kbest* can not only further balance the exploration and exploitation but also significantly improve the computational efficiency. In order to evaluate the binary version of the proposed algorithm for feature selection, a set of standard datasets are utilized in the experiment. The experimental results show that, in most cases, BIGSA can obtain better solutions and provide higher performance compared with other algorithms. Therefore, it is confirmed that BIGSA is suitable for solving feature selection problem.

REFERENCES

[1] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, Oct. 2017. doi: 10.1109/ACCESS.2017.2762418.

[2] F. Attal, A. Boubezoul, L. Oukhellou, and S. Espié, "Powered two-wheeler riding pattern recognition using a machine-learning framework," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 475–487, Feb. 2015. doi: 10.1109/TITS.2014.2346243.

[3] M. Krendzelak and F. Jakab, "Text categorization with machine learning and hierarchical structures," in *Proc. IEEE ICETA*, Stary Smokovec, Slovakia, Nov. 2016, pp. 1–5.

[4] Z. Ge, Z. Song, S. X. Ding, and B. Huang, "Data mining and analytics in the process industry: The role of machine learning," *IEEE Access*, vol. 5, pp. 20590–20616, Sep. 2017. doi: 10.1109/ACCESS.2017.2756872.

[5] X. Zhao, X. Li, and Z. Zhang, "Multimedia retrieval via deep learning to rank," *IEEE Signal Process. Lett.*, vol. 22, no. 9, pp. 1487–1491, Mar. 2015. doi: 10.1109/LSP.2015.2410134.

[6] G. Taşkın, H. Kaya, and L. Bruzzone, "Feature selection based on high dimensional model representation for hyperspectral images," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2918–2928, Jun. 2017. doi: 10.1109/TIP.2017.2687128.

[7] C. Yao, Y.-F. Liu, B. Jiang, J. Han, and J. Han, ''LLE score: A new filter-based unsupervised feature selection method based on nonlinear manifold embedding and its application to image recognition,'' *IEEE Trans. Image Process.*, vol. 26, no. 11, pp. 5257–5269, Nov. 2017. doi: 10.1109/TIP.2017.2733200.

[8] L. Ma, M. Li, Y. Gao, T. Chen, X. Ma, and L. Qu, ''A novel wrapper approach for feature selection in object-based image classification using polygon-based cross-validation,'' *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 3, pp. 409–413, Mar. 2017. doi: 10.1109/LGRS.2016.2645710.

[9] J. Zhao, L. Chen, W. Pedrycz, and W. Wang, ''Variational inference-based automatic relevance determination kernel for embedded feature selection of noisy industrial data,'' *IEEE Trans. Ind. Electron.*, vol. 66, no. 1, pp. 416–428, Jan. 2019. doi: 10.1109/TIE.2018.2815997.

[10] A. A. Naeini, M. Babadi, S. M. J. Mirzadeh, and S. Amini, ''Particle swarm optimization for object-based feature selection of VHSR satellite images,'' *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 3, pp. 379–383, Mar. 2018. doi: 10.1109/LGRS.2017.2789194.

[11] H. Peng, C. Ying, S. Tan, B. Hu, and Z. Sun, ''An improved feature selection algorithm based on ant colony optimization,'' *IEEE Access*, vol. 6, pp. 69203–69209, Nov. 2018. doi: 10.1109/ACCESS.2018.2879583.

[12] P. Alirezazadeh, A. Fathi, and F. Abdali-Mohammadi, ''A genetic algorithm-based feature selection for kinship verification,'' *IEEE Signal Process. Lett.*, vol. 22, no. 12, pp. 2459–2463, Dec. 2015. doi: 10.1109/LSP.2015.2490805.

[13] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, ''GSA: A gravitational search algorithm,'' *Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, Jun. 2009. doi: 10.1016/j.ins.2009.03.004.

[14] H. Zandevakili, E. Rashedi, and A. Mahani, ''Gravitational search algorithm with both attractive and repulsive forces,'' *Soft Comput.*, vol. 2, pp. 783–825, Feb. 2017. doi: 10.1007/s00500-017-2785-2.

[15] H. Mittal, R. Pal, A. Kulhari, and M. Saraswat, ''Chaotic Kbest gravitational search algorithm (CKGSA),'' in *Proc. IEEE IC*, Noida, India, Aug. 2016, pp. 1–6.

[16] S. Mirjalili, G.-G. Wang, and L. D. S. Coelho, ''Binary optimization using hybrid particle swarm optimization and gravitational search algorithm,'' *Neural Comput. Appl.*, vol. 25, no. 6, pp. 1423–1435, Nov. 2014. doi: 10.1007/s00521-014-1629-6.

[17] S. He, L. Zhu, L. Wang, L. Yu, and C. Yao, ''A modified gravitational search algorithm for function optimization,'' *IEEE Access*, vol. 7, pp. 5984–5993, Jan. 2019. doi: 10.1109/ACCESS.2018.2889854.

[18] H. Bostani and M. Sheikhan, ''Hybrid of binary gravitational search algorithm and mutual information for feature selection in intrusion detection systems,'' *Soft Comput.*, vol. 21, no. 9, pp. 2307–2324, May 2017. doi: 10.1007/s00500-015-1942-8.

[19] E. Rashedi, H. Nezamabadi-Pourand, and S. Saryazdi, ''BGSA: Binary gravitational search algorithm,'' *Natural Comput.*, vol. 9, no. 3, pp. 727–745, Sep. 2010. doi: 10.1007/s11047-009-9175-3.

[20] S. Mirjalili and A. Lewis, ''Adaptive gbest-guided gravitational search algorithm,'' *Neural Comput. Appl.* vol. 25, nos. 7–8, pp. 1569–1584, Dec. 2014. doi: 10.1007/s00521-014-1640-y.

[21] H. Mittal and M. Saraswat, ''An optimum multi-level image thresholding segmentation using non-local means 2D histogram and exponential Kbest gravitational search algorithm,'' *Eng. Appl. Artif. Intell.*, vol. 71, pp. 226–235, May 2018. doi: 10.1016/j.engappai.2018.03.001.

[22] X. Han, X. Chang, L. Quan, X. Xiong, J. Li, Z. Zhang, and Y. Liu, ''Feature subset selection by gravitational search algorithm optimization,'' *Inf. Sci.*, vol. 281, pp. 128–146, Oct. 2014. doi: 10.1016/j.ins.2014.05.030.

[23] S. Nagpal, S. Arora, S. Dey, and Shreya, ''Feature selection using gravitational search algorithm for biomedical data,'' *Procedia Comput. Sci.*, vol. 115, pp. 258–265, Jan. 2017. doi: 10.1016/j.procs.2017.09.133.

[24] J. Xiang, X. Han, F. Duan, Y. Qiang, X. Xiong, Y. Lan, and H. Chai, ''A novel hybrid system for feature selection based on an improved gravitational search algorithm and k-NN method,'' *Appl. Soft Comput.*, vol. 31, pp. 293–307, Jun. 2015. doi: 10.1016/j.asoc.2015.01.043.

[25] W. Wang, C. Li, X. Liao, and H. Qin, ''Study on unit commitment problem considering pumped storage and renewable energy via a novel binary artificial sheep algorithm,'' *Appl. Energy*, vol. 187, pp. 612–626, Feb. 2017. doi: 10.1016/j.apenergy.2016.11.085.

[26] S. K. Reddy, L. Panwar, B. K. Panigrahi, and R. Kumar, ''Binary whale optimization algorithm: A new metaheuristic approach for profit-based unit commitment problems in competitive electricity markets,'' *Eng. Optim.*, vol. 51, pp. 369–389, May 2018. doi: 10.1080/0305215X.2018.1463527.

[27] S. K. Reddy, L. K. Panwar, B. K. Panigrahi, R. Kumar, and A. Alsumaiti, ''Binary grey wolf optimizer models for profit based unit commitment of price-taking GENCO in electricity market,'' *Swarm Evol. Comput.*, vol. 44, pp. 957–971, Feb. 2019. doi: 10.1016/j.swevo.2018.10.008.

[28] S. Sun, J. Fu, F. Zhu, and N. Xiong, ''A compound structure for wind speed forecasting using MKLSSVM with feature selection and parameter optimization,'' *Math. Problems Eng.*, vol. 2018, Nov. 2018, Art. no. 9287097. doi: 10.1155/2018/9287097.

[29] M. Luo, C. Li, X. Zhang, R. Li, and X. An, ''Compound feature selection and parameter optimization of ELM for fault diagnosis of rolling element bearings,'' *ISA Trans.*, vol. 65, pp. 556–566, Nov. 2016. doi: 10.1016/j.isatra.2016.08.022.

[30] C. Li, Z. Xiao, X. Xia, W. Zou, and C. Zhang, ''A hybrid model based on synchronous optimisation for multi-step short-term wind speed forecasting,'' *Appl. Energy*, vol. 215, pp. 131–144, Apr. 2018. doi: 10.1016/j.apenergy.2018.01.094.

LEI ZHU received the Ph.D. degree from the College of Communications Engineering, PLA University of Science and Technology, China, in 2002. He is currently a Professor in system engineering. He is the author of more than 35 scientific articles. His research interests include network planning and system simulation.

SHOUSHUAI HE received the B.S. degree in information security from the Harbin Institute of Technology, in 2017. He is currently pursuing the master's degree with the Army Engineering University of PLA. His research interests include artificial intelligence and network security.

LEI WANG received the Ph.D. degree in military operational research from the PLA University of Science and Technology, in 2014, China. He is currently a Postdoctoral Researcher in communication and information system, engineer of the optimization, and system engineering with the College of Communications Engineering, Army Engineering University of PLA, China. He is the author of more than 25 scientific articles. His research interests include knowledge engineering, data mining, artificial intelligence, network planning, and military operational research.

WEIJUN ZENG received the Ph.D. degree from the Institute of Communications Engineering, PLA University of Science and Technology, China, in 2016. He is currently a Lecturer with the College of Communications Engineering, Army Engineering University of PLA. His current research interests include signal processing and machine learning.

JIAN YANG received the M.S. degrees in computer application from the Institute of Communications Engineering, China, in 1998. He is currently an Associate Professor with the College of Communications Engineering, Army Engineering University of PLA. His current research interest includes network management.