Journal of Network Intelligence ©2022 ISSN 2414-8105 (Online)
Taiwan Ubiquitous Information Volume **7**, Number **1**, February **2022**

Binary Equilibrium Optimizer Algorithm

Pei Hu

College of Computer Science and Engineering
Shandong University of Science and Technology
No.579 Qianwan'gang Road, Qingdao, Shandong, 266590, China
School of Computer and Software
Nanyang Institute of Technology
No.80 Changjiang Road, Nanyang, Henan, 473004, China
huxiaopei163@163.com

Shu-Chuan Chu

College of Computer Science and Engineering
Shandong University of Science and Technology
No.579 Qianwan'gang Road, Qingdao, Shandong, 266590, China
College of Science and Engineering
Flinders University
Sturt Rd, Bedford Park SA 5042, Australia
scchu0803@gmail.com

Václav Snášel

Department of Computer Science
VŠB—Technical University of Ostrava
17.listopadu 2172/15, 708 00 Ostrava-Poruba, Czech Republic
vaclav.snasel@vsb.cz

Jeng-Shyang Pan*

College of Computer Science and Engineering
Shandong University of Science and Technology
No.579 Qianwan'gang Road, Qingdao, Shandong, 266590, China
Department of Information Management
Chaoyang University of Technology
No.168 Jifeng E. Rd., Wufeng District, Taichung, 413310, Taiwan
*Corresponding Author, jengshyangpan@gmail.com

Received March 2021; revised September 2021

Abstract. *Equilibrium optimizer (EO) is a new proposed meta-heuristic algorithm by utilizing the mass balance model of the control volume. In order to solve the binary applications, this paper proposes a binary version of equilibrium optimizer (BEO). BEO takes advantage of the structure of EO, only modifying the equations of equilibrium concentration and position updating. Through the benchmark functions and Wilcoxon's rank sum test, it compares BEO with binary bat algorithm, binary differential evolution, binary grey wolf optimizer, binary particle swarm optimization, and a binary hybrid algorithm of particle swarm optimization and gravitational search algorithm. BEO shows exceptional performance in the solving quality, time complexity and convergence. Simulation results prove that BEO acquires the least classification errors while having a small number of features in the KNN.*

Keywords: Equilibrium optimizer; Binary; Feature selection; KNN

---

1. **Introduction.** With the development of information technology and the growth of sensors, people have produced a large number of data [1–4]. But the data does not provide sufficient knowledge, and even noise data weakens the decision-making ability [5–7]. The dimension reduction is one of the most basic preprocessing methods in data mining [8–10].

Feature construction transforms the data set from high dimensional space to low space, which is more suitable for the learning process [11, 12]. However, it changes the structure of features and they are not easily explained [13, 14]. Feature selection chooses a feature subset from the original features and discards the features that are harmful to the subsequent learning process [15–17].

Feature selection contains filter, wrapper and embedded approaches [18]. The filter method is based on the statistical information of features and does not involve a specific learning algorithm. Hence, it has efficient and fast. The wrapper adopts a given algorithm to the feature subset and its performance is better than filter [19, 20].

Since the main goal of feature selection is to maximize the classification accuracy while minimizing the number of selected features, it is regarded as an optimization task [21, 22]. If the original data set has n features, then $2^n - 1$ subsets are generated. It is impractical when n is very large [23]. Meta-heuristics have the powerful abilities of global search and local search [24–27], and they contain evolutionary computation, swarm intelligence and algorithms based on physical phenomena [28–31].

Neggaz et al. used sine function to update the follower's position in sine cosine algorithm (SSA) to improve the exploration stage and avoid local stagnation [32]. O'Neill et al. claimed three initialization strategies of particle swarm optimization (PSO) and new update mechanisms of personal and global optimal solutions to implement feature selection [33]. Mafarja et al. brought a hybrid meta-heuristic method to overcome the shortcomings of immature convergence and stagnation [34]. Arora et al. introduced a binary butterfly optimization algorithm (BOA) and applied it to the classification problem in wrapper mode [35]. Faris et al. utilized eight transfer functions to convert continuous salp swarm algorithm (SSA) to binary versions and a crossover operator to promote the exploration of the algorithm [36]. Hu et al. improved the binary gray wolf optimizer (GWO) and proposed new transfer functions and position equation [37].

Equilibrium optimizer (EO) is a novel meta-heuristic [38]. It has shown excellent performance in the optimizations of engineering and parameter, and image processing [39]. This paper proposes a binary version of EO and adopts it in the feature selection.

The organization of this paper is as follows: Section 2 describes the concise introduction of equilibrium optimizer. Section 3 presents the fundamental principles of the proposed binary equilibrium optimizer (BEO). Section 4 discusses the experimental results of the benchmark functions and BEO is compared with five famous binary optimization methods, including: 1) BDE [40], a binary version of evolutionary algorithm differential evolution (DE). 2) BPSO [41], a binary version of swarm intelligence algorithm PSO. 3) BBA [42] and BGWO [43], two binary versions of newly swarm intelligence algorithms bat algorithm (BA) and GWO. 4) BPSOGSA [44], a binary version of hybrid algorithm of PSO and gravitational search algorithm (GSA). BEO achieves great results in the solving quality, time complexity and convergence. Section 5 implements feature selection by the compared algorithms and argues the results. Section 6 concludes the works and suggests several directions for further studies.

2. **Equilibrium Optimizer.** Equilibrium optimizer is a newly physics-based algorithm, proposed by Faramarzi et al. in 2020. It utilizes the dynamic models of sink and source for estimating equilibrium concentration.

2.1. **Inspiration Analysis.** The mass conservation equation represents the equilibrium calculation for adding and removing mass in a defined fluid region. Suppose V is a fixed, undeformed fluid volume, called the control volume. The mass conservation requires that the change rate of mass in the control volume (V) over time is equal to the mass rate entering the V plus the incremental/lost mass rate in the V due to the source and sink effects. The integral form of the mass conservation in the control volume is expressed as follows:

$$V\frac{dC}{dt} = QC_{eq} - QC + G \tag{1}$$

where C represents the concentration in the V. $\frac{dC}{dt}$ means the change rate of mass in the V. Q denotes the volume flow in and out of the V. C_{eq} is the concentration in the equilibrium state, in which there is no production in the V. G is the generation rate of mass within the V. When $\frac{dC}{dt}$ equals zero, V reaches a steady equilibrium state. Then, the function of time (t) for the concentration (C) in the V is obtained by rearranging Eq (1).

$$\frac{dC}{\lambda C_{eq} - \lambda C + \frac{G}{V}} = dt \tag{2}$$

where $\lambda = \frac{Q}{V}$, Eq (3) is computed by integrating of Eq (2).

$$\int_{C_0}^{C} \frac{dC}{\lambda C_{eq} - \lambda C + \frac{G}{V}} = \int_{t_0}^{t} dt \tag{3}$$

Finally, Eq (4) is calculated by Eq (3).

$$C = C_{eq} + (C_0 - C_{eq})F + \frac{G}{\lambda V}(1 - F) \tag{4}$$

where $F = exp[-\lambda(t - t_0)]$. C_0 and t_0 are the initial values of concentration and time. Eq (4) can either calculate the concentration in the V by a given turnover rate, or compute the average turnover rate with a simple linear regression using the given generation rate and other prerequisites.

2.2. **Mathematical Model of EO.** EO randomly initializes the positions of the population and a concentration represents the position of a particle. V is a unit volume. The position updating is defined as follows.

$$\overrightarrow{C_i}(n+1) = \overrightarrow{C_{eq}}(n) + (\overrightarrow{C_i}(n) - \overrightarrow{C_{eq}}(n))\overrightarrow{F}(n) + \frac{\overrightarrow{G}(n)}{\lambda}(1 - \overrightarrow{F}(n)) \tag{5}$$

C_{eq} is the equilibrium pool and is constructed by the positions of the first four optimal solutions and their average value. The algorithm randomly chooses one from C_{eq} for each run.

F is an exponential term, which is used to control the balance between exploration and exploitation of the algorithm.

$$t(n) = (1 - \frac{n}{Max\_iter})^{(2\frac{n}{Max\_iter})} \tag{6}$$

$$\overrightarrow{F}(n) = sign(\overrightarrow{r} - 0.5)[e^{-\overrightarrow{\lambda} t(n)} - 1] \tag{7}$$

where Max_iter means the maximum iteration. $\overrightarrow{r}$ and $\overrightarrow{\lambda}$ are two random vectors between [0,1]. $Sign$ is the signum function of Matlab. G assists the algorithm to acquire better performance and it is computed as follows.

$$GCP = \begin{cases} 0.5r_1 & if(r_2 \geq GP) \\ 0 & else \end{cases} \tag{8}$$

$$\overrightarrow{G_0}(n) = GCP * (\overrightarrow{X_{eq}}(n) - \overrightarrow{X_i}(n)) \tag{9}$$

$$\overrightarrow{G}(n) = \overrightarrow{G_0}(n) * \overrightarrow{F}(n) \tag{10}$$

where r_1 and r_2 are two random numbers between [0,1].

3. **Binary Equilibrium Optimizer (BEO).** In the EO, particles can be anywhere in the search space. While the position of a particle is encoded with a binary vector in the binary equilibrium optimizer (BEO), particles update their positions by changing 0 to 1, or 1 to 0. Their positions are very restricted and BEO cannot utilize the Eq (6) to perform the position updating. The difference between EO and BEO is the position updating mechanism. Consequently, several models of EO are modified to meet the needs of BEO.

In the EO, the position updating of a particle is mainly accomplished by the equilibrium pool, exponential term F and generation rate G. The equilibrium pool contains the first four optimal solutions and the average of them, but the value may not be 1 or 0 in the binary EO. For example, in the j^{th} dimension, $C^j_{eq,1} = 1$, $C^j_{eq,2} = 1$, $C^j_{eq,3} = 1$ and $C^j_{eq,4} = 0$, then $C^j_{eq,avg} = 0.75$, which contradicts the position requirement of BEO. So the updating of $C^j_{eq,avg}$ is redefined as following:

$$C^j_{eq,avg}(n) = \begin{cases} 1 & if(C^j_{eq,1}(n) + C^j_{eq,2}(n) + C^j_{eq,3}(n) \\ & \quad +C^j_{eq,4}(n)) \geq 2 \\ 0 & else \end{cases} \tag{11}$$

Eq (11) ensures that all solutions in the equilibrium pool are located at 0 or 1 of the hypercube. In the EO, only when the values of the four optimal solutions are all 1 or all 0 can the requirements of BEO be satisfied. From Eq (11), it is known that if half of the first four optimal values are 1, $C_{eq,avg}$ is 1. BEO has a high probability of acquiring 1 from the equilibrium pool.

F is used to control the exploration and exploitation of EO and G enhances the algorithm's global and local search. Both of them play an important role in the updating position of EO. In BEO, they need to be mapped to the [0, 1] space using a transfer function and then they are compared with a random number between [0,1]. BEO adopts the following transfer function and the curve of it is shown in the Figure 1:

$$V(x) = 1/(1 + exp(-10 * (x - 0.4))) \tag{12}$$

where $x = (C^j_i - C^j_{eq}) * F + (G/\lambda) * (1 - F)$.

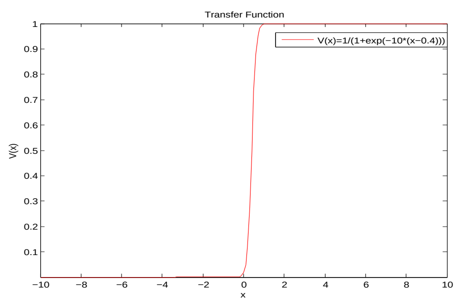


FIGURE 1. BEO's transfer function.

After calculating the value of transfer function, a new position updating equation is proposed.

$$C_i^j = \begin{cases} 1 - C_i^j & if(V(x) \geq rand) \\ C_i^j & else \end{cases} \tag{13}$$

Algorithm 1 is the pseudo code of BEO.

Algorithm 1: BEO

```
Initialize the related parameters of EO;
Randomly generate the positions of particles;
Compute the fitness of each particle;
for it = 1 : MAX_IT do
    C_eq,1 = the position of the first optimal solution;
    C_eq,2 = the position of the second optimal solution;
    C_eq,3 = the position of the third optimal solution;
    C_eq,4 = the position of the fourth optimal solution;
    for d = 1 : dim do
        Use Eq (11) to update C^d_eq,avg;
    C_pool = [C_eq,1; C_eq,2; C_eq,3; C_eq,4; C_eq,avg];
    t = (1 − it/MAX_IT)^(2*n/MAX_IT);
    for i = 1 : population_size do
        C_eq = C_pool(randi(size(C_pool, 1)), :);
        for d = 1 : dim do
            λ = rand() ;
            r = rand() ;
            F = sign(r − 0.5) * (exp(−λ * t) − 1);
            r1 = rand();
            r2 = rand();
            GCP = 0.5 * r1 * (r2 >= GP);
            G_0 = GCP * (C_eq(d) − λ * X(i, d));
            G = G_0. * F;
            tran_val = (C(i, d) − C_eq(d)) * F + (G/λ) * (1 − F);
            Use Eq (12) to acquire s by tran_val;
            Use Eq (13) to update C(i, d) by s;
        Update the fitness of C(i);
Output C_eq,1;
```

4. **Experimental Results and Analysis.** In this section, 29 benchmark functions examine the abilities of BEO, where the functions have been used by EO and many researchers. In order to validate the performance of BEO, it is compared with BBA, BDE, BGWO, BPSO and BPSOGSA. Table 1 describes the details of the benchmark functions. *Space* shows the boundary of search space; *Dim* denotes the dimension and f_{min} represents the optimum.

Benchmark functions include unimodal, multimodal, fixed-dimension and composite functions. Where f_1-f_7 are unimodal functions, f_8-f_{13} are multimodal functions, f_{14}-f_{23} are fixed-dimension functions, the rests are composite functions.

To make a fair comparison, all algorithms run 500 iterations and 30 times in every benchmark function, and their population size is 30. Table 2 lists the values of the key parameters of the compared algorithms. For the convenience of reading, all experimental data in this section and Section 5 are rounded to four decimal places. Wilcoxon's rank sum test is performed at a 5% significance level to judge whether the experimental results are statistically significant.

Table 3 shows the average (AVG) and standard deviation (STD) values of the compared algorithms in the benchmark functions. In Table 3, if the compared algorithms acquire the best result in the benchmark function, their data has red color. Table 4 is the results of Wilcoxon's rank-sum test based on BEO. "+" shows that the compared algorithm is superior to BEO, and "=" means that the algorithms have the same results. "-" represents that the algorithm is inferior to BEO.

4.1. **Analyse the Solution Quality.** From Table 3, it is concluded that BEO has the best performance. Except for f_{25}, it has the optimal results in the compared algorithms. BPSOGSA wins the second performance and performs well in 18 test functions, which is superior to the test results of BPSO. As can be seen from Table 4, BBA implements the worst and has 20 test functions that are inferior to BEO. BDE has 12 test functions worse than BEO, but only it has the only function better than BEO. BGWO is not as good as BEO in 16 test functions. BPSO and BPSOGSA have 7 and 6 functions that are worse than BEO. In terms of stability, BEO also executes perfectly. It is only slightly less stable in f_{25} (4.1422) and other test functions have no significant fluctuations. BPSOGSA has well stability in the multimodal and fixed-dimension functions. BPSO has good stability in the fixed-dimension functions. Regardless of the solution quality and stability, BEO is superior to other compared algorithms. This is an evidence that the parameters F and G well balance the local search and global search of the BEO, and have a forceful ability to find the optimal value.

BEO achieves satisfying results in the unimodal functions, except for f_5. The results acquired by other algorithms are not as good as BEO. BPSOGSA obtains the optimal solution in 2 test functions. BDE and BGWO acquire the optimal solution in 1 function. It demonstrates that BEO has high performance in seeking for the global solutions in the unimodal functions. The generation rate G helps the algorithm to exploit the optimal solution more carefully in the known space.

BEO again performs well in the multimodal functions, especially it earns the theoretically optimal value in f_9, f_{10}, f_{11} and f_{13}. BPOSGA achieves the optimal solution in f_9, f_{10} and f_{13}. BDE and BGWO acquire the optimal result in f_{13}. Multimodal function contains exponential local optima, so it is appropriate to judge whether the algorithm prevents falling into local minima. This shows that BEO effectively avoids local optimum and has the ability to jump out of local traps. The exponential term F makes BEO search for the optimal value in more space.

TABLE 1. The benchmark functions.

Function	Space	Dim	f_{min}
$f_1(x) = \sum_{i=1}^{n} x_i^2$	[-100, 100]	30	0
$f_2(x) = \sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$	[-10, 10]	30	0
$f_3(x) = \sum_{i=1}^{n}(\sum_{j-1}^{i} x_j)^2$	[-100, 100]	30	0
$f_4(x) = max_i\{\lvert x_i\rvert, 1 \leq i \leq n\}$	[-100, 100]	30	0
$f_5(x) = \sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]$	30	0
$f_6(x) = \sum_{i=1}^{n}([x_i + 0.5])^2$	[-100, 100]	30	0
$f_7(x) = \sum_{i=1}^{n} ix_i^4 + random[0, 1)$	[-1.28, 1.28]	30	0
$f_8(x) = \sum_{i=1}^{n} -x_i sin(\sqrt{\lvert x_i\rvert})$	[-500,500]	30	-12569
$f_9(x) = \sum_{i=1}^{n}[x_i^2 - 10cos(2\pi x_i) + 10]$	[-5.12,5.12]	30	0
$f_{10}(x) = -20exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - exp(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)) + 20 + e$	[-32,32]	30	0
$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600,600]	30	0
$f_{12}(x) = \frac{\pi}{n}\{10sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4) y_i = 1 + \frac{x_i + 1}{4} u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	[-50,50]	30	0
$f_{13}(x) = 0.1\{sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$	[-50,50]	30	0
$f_{14}(x) = (\frac{1}{500}\sum_{j=1}^{25}\frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6})^{-1}$	[-65, 65]	2	1
$f_{15}(x) = \sum_{i=1}^{11}[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$	[-5, 5]	4	0.00030
$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	[-5, 5]	2	-1.0316
$f_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})cosx_1 + 10$	[-5, 5]	2	0.398
$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	[-2, 2]	2	3
$f_{19}(x) = -\sum_{i=1}^{4} c_i exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$	[1, 3]	3	-3.86
$f_{20}(x) = -\sum_{i=1}^{4} c_i exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	[0, 1]	6	-3.32
$f_{21}(x) = -\sum_{i=1}^{5}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0, 10]	4	-10.1532
$f_{22}(x) = -\sum_{i=1}^{7}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0, 10]	4	-10.4028
$f_{23}(x) = -\sum_{i=1}^{10}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0, 10]	4	-10.5363
f_{24} $f_1, f_2, f_3, ..., f_{10} = Sphere\ \ Function$ $[\sigma_1, \sigma_2, ..., \sigma_{10}] = [1, 1, 1, ..., 1]$ $[\lambda_1, \lambda_2, ..., \lambda_{10}] = [5/100, 5/100, 5/100, ..., 5/100]$	[-5, 5]	30	0
f_{25} $f_1, f_2, f_3, ..., f_{10} = Griewank's\ \ Function$ $[\sigma_1, \sigma_2, ..., \sigma_{10}] = [1, 1, 1, ..., 1]$ $[\lambda_1, \lambda_2, ..., \lambda_{10}] = [5/100, 5/100, 5/100, ..., 5/100]$	[-5, 5]	30	0
f_{26} $f_1, f_2, f_3, ..., f_{10} = Griewank's\ \ Function$ $[\sigma_1, \sigma_2, ..., \sigma_{10}] = [1, 1, 1, ..., 1]$ $[\lambda_1, \lambda_2, ..., \lambda_{10}] = [1, 1, 1, ..., 1]$	[-5, 5]	30	0
f_{27} $f_1, f_2 = Ackley's Function, f_3, f_4 = Rastrigin's\ \ Function,$ $f_5, f_6 = Weierstrass Function, f_7, f_8 = Griewank's Function$ $f_9, f_{10} = Sphere\ \ Function$ $[\sigma_1, \sigma_2, ..., \sigma_{10}] = [1, 1, 1, ..., 1]$ $[\lambda_1, \lambda_2, ..., \lambda_{10}] = [5/32, 5/32, 1, 1, 5/0.5, 5/0.5, 5/100, 5/100, 5/100, 5/100]$	[-5, 5]	30	0
f_{28} $f_1, f_2 = Rastrigin's\ \ Function, f_3, f_4 = Weierstrass\ \ Function,$ $f_5, f_6 = Griewank's Function, f_7, f_8 = Ackley's\ \ Function$ $f_9, f_{10} = Sphere\ \ Function$ $[\sigma_1, \sigma_2, ..., \sigma_{10}] = [1, 1, 1, ..., 1]$ $[\lambda_1, \lambda_2, ..., \lambda_{10}] = [1/5, 1/5, 5/0.5, 5/0.5, 5/100, 5/100, 5/32, 5/32, 5/100, 5/100]$	[-5, 5]	30	0
f_{29} $f_1, f_2 = Rastrigin's\ \ Function, f_3, f_4 = Weierstrass\ \ Function,$ $f_5, f_6 = Griewank's\ \ Function, f_7, f_8 = Ackley's\ \ Function$ $f_9, f_{10} = Sphere\ \ Function$ $[\sigma_1, \sigma_2, ..., \sigma_{10}] = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]$ $[\lambda_1, \lambda_2, ..., \lambda_{10}] = [0.1 * 1/5, 0.2 * 1/5, 0.3 * 5/0.5, 0.4 * 5/0.5, 0.5 * 5/100, 0.6 * 5/100, 0.7 * 5/32, 0.8 * 5/32, 0.9 * 5/100, 1 * 5/100]$	[-5, 5]	30	0

TABLE 2. The details of the compared algorithms.

Algorithm	Parameters
BBA	A=0.9 r=0.9 F_{max}=2 F_{min}=0
BDE	cr=0.9
BGWO	a=2
BPSO	c_1=2 c_2=2 w=2 w_{max}=0.9 w_{min}=0.4 V_{max}=6
BPSOGSA	α=20 G_0=100 $Rnorm$=2 $Rpower$=1

TABLE 3. The statistical results of the compared algorithms.

Function	BBA		BDE		BEO		BGWO		BPSO		BPSOGSA	
	AVG	STD	AVG	STD	AVG	STD	AVG	STD	AVG	STD	AVG	STD
f_1	1.0667	0.9803	2.9333	1.6595	0	0	5.2333	1.2507	0.0333	0.1826	0	0
f_2	1.0667	1.0483	3.0667	1.388	0	0	4.9667	1.3257	0.0333	0.1826	0	0
f_3	26.2333	39.9704	148.5	117.0967	0	0	321.6333	166.2421	0.1	0.3051	0.0333	0.1826
f_4	0.8333	0.379	1	0	0	0	1	0	1	0	1	0
f_5	290.4333	159.6282	0	0	0	0	0	0	116.5333	95.5369	198.9333	120.9357
f_6	9.5667	2.2581	13.7	2.7965	7.5	0	18.6333	3.3086	7.5667	0.3651	7.5	0
f_7	25.5569	22.7144	50.7334	22.9601	0.0001	0.0001	72.6001	16.2896	0.431	0.6022	0.0668	0.2537
f_8	-17.3624	0.8114	-25.2441	0	-25.2441	0	-25.2441	0	-25.2441	0	-25.2441	0
f_9	1.0667	1.1427	3.7333	1.4126	0	0	4.8667	1.4077	0.0667	0.2537	0	0
f_{10}	0.5521	0.4578	1.2127	0.2755	0	0	1.606	0.1836	0	0	0	0
f_{11}	0.0372	0.0376	0.1624	0.0587	0	0	0.2331	0.0603	0.0017	0.0052	0	0
f_{12}	1.8514	0.1334	2.0876	0.1971	1.669	0	2.5868	0.2465	1.6707	0.0096	1.669	0
f_{13}	0.9567	0.1135	0	0	0	0	0	0	0.0033	0.0183	0	0
f_{14}	12.6705	0	12.6705	0	12.6705	0	12.6705	0	12.6705	0	12.6705	0
f_{15}	0.1484	0	0.1484	0	0.1484	0	0.1484	0	0.1484	0	0.1484	0
f_{16}	0	0	0	0	0	0	0	0	0	0	0	0
f_{17}	27.7029	0	27.7029	0	27.7029	0	27.7029	0	27.7029	0	27.7029	0
f_{18}	600	0	600	0	600	0	600	0	600	0	600	0
f_{19}	-0.3348	0	-0.3348	0	-0.3348	0	-0.3325	0.0087	-0.3348	0	-0.3348	0
f_{20}	-0.1507	0.0306	-0.1602	0.0143	-0.1657	0	-0.1415	0.0478	-0.1657	0	-0.1657	0
f_{21}	-4.6379	1.2734	-5.0552	0	-5.0552	0	-5.0552	0	-5.0552	0	-5.0552	0
f_{22}	-4.8092	1.0597	-5.0877	0	-5.0877	0	-5.0877	0	-5.0877	0	-5.0877	0
f_{23}	-4.7105	1.2754	-5.1285	0	-5.1285	0	-5.1285	0	-5.1285	0	-5.1285	0
f_{24}	887.8665	4.72	900	0	857.243	0	870.6608	6.3999	857.3351	0.1694	857.3345	0.433
f_{25}	918.036	12.7047	900	0	915.2382	4.1422	924.2313	3.5271	916.3785	0.1103	916.4276	0.2109
f_{26}	948.7455	63.8031	900	0	900	0	1147.0084	28.1442	906.4274	11.1142	903.1327	8.1404
f_{27}	934.2393	22.9375	900	0	900	0	1012.6667	8.2132	900.9881	5.412	900	0
f_{28}	955.805	41.2041	900	0	900	0	1064.43	9.0621	904.4227	13.4979	907.5154	17.0998
f_{29}	908.6275	7.757	900	0	900	0	926.5543	2.7821	903.8725	4.1036	908.0299	6.4098

Fixed-dimension function only has a few local optimal solutions and the dimension is small. BEO, BPSO and BPSOGSA have exactly the same results. BGWO shows not well in f_{20} and BGWO performs worse in f_{19} and f_{20}. BBA executes the worst. Composite multimodal function has exceptionally complex structures with several randomly located deep local optima and many randomly located global optimum. The compared algorithms don't achieve good results. BDE and BEO complete relatively well. Although BEO does not perform as well as BDE in f_{25}, it is better than BBA, BGWO, BPSO and BPSOGSA.

4.2. **Analyse the Time Complexity and Convergence.** The time complexity of BBA, BDE, BEO, BGWO and BPSOGSA is $O(T{*}P{*}f{+}T{*}P{*}D{*}ft)$, while BPSOGSA is $O(T{*}P{*}f{+}T{*}P{*}P{*}D{*}ft)$. Where T is the iterations, P means the population size, f represents the computational time of the fitness function, ft denotes the running time of transfer function. Table 5 is the average time of the compared algorithms running once in the benchmark function. It can be seen that BPSO has the least time complexity because of its few parameters and simple updating equations. Although the solution quality of BPSOGSA has been greatly improved compared with BPSO, it utilizes the calculation equations of BPSO and BGSA, and has a large running time. The transfer function of BBA is more complicated than BDE, BEO, BGWO and BPSO, so the time is large. BEO is the second only to BPSO, which shows that its equations are not too complicated. The compared algorithms have the lowest time complexity in f_{15}-f_{23} and have little difference for running time in f_1-f_{12}, which indicates that the dimension is a significant element

TABLE 4. Wilcoxon's rank-sum test for the optimal results on BEO.

Function	BBA	BDE	BGWO	BPSO	BPSOGSA
f_1	-	-	-	=	=
f_2	-	-	-	=	=
f_3	-	-	-	=	=
f_4	-	-	-	-	-
f_5	-	=	=	-	-
f_6	-	-	-	=	=
f_7	-	-	-	-	=
f_8	-	=	=	=	=
f_9	-	-	-	=	=
f_{10}	-	-	-	=	=
f_{11}	-	-	-	=	=
f_{12}	-	-	-	=	=
f_{13}	-	=	=	=	=
f_{14}	=	=	=	=	=
f_{15}	=	=	=	=	=
f_{16}	=	=	=	=	=
f_{17}	=	=	=	=	=
f_{18}	=	=	=	=	=
f_{19}	=	=	=	=	=
f_{20}	-	-	-	=	=
f_{21}	=	=	=	=	=
f_{22}	=	=	=	=	=
f_{23}	=	=	=	=	=
f_{24}	-	-	-	-	=
f_{25}	-	+	-	-	-
f_{26}	-	=	-	-	-
f_{27}	-	=	-	=	=
f_{28}	-	=	-	=	-
f_{29}	-	=	-	-	-

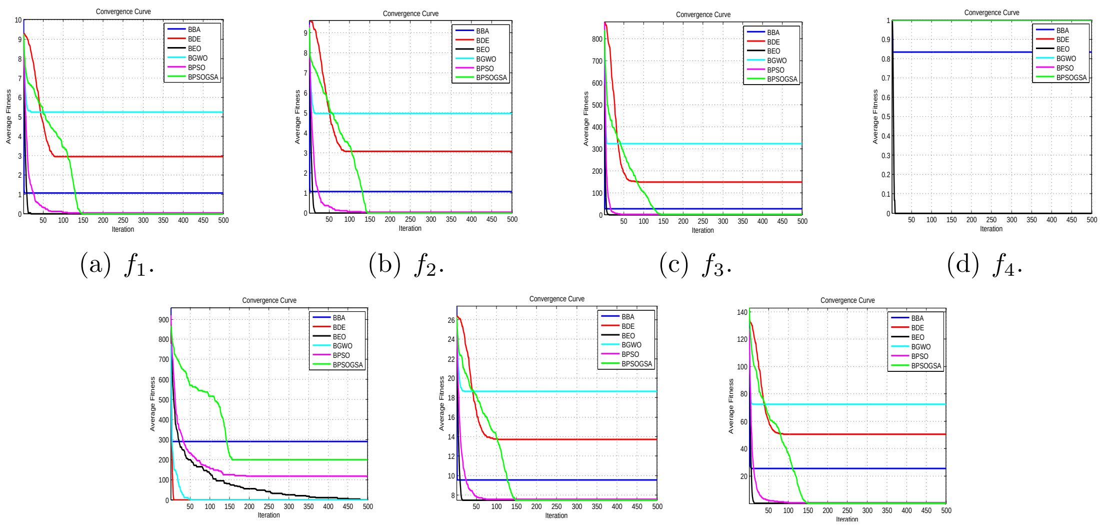
influencing the time complexity of algorithm and it is not related to whether the test function is an unimodal or a multimodal function. While they have large computational time in f_{24}-f_{29}, which proves that the structure of the benchmark function affects the time complexity of algorithm.

Unimodal function includes merely a global optimal solution and it has no local trap. Consequently, it is useful to examine the convergence rates of the algorithms. Figure 2 displays the curves of the compared algorithms in the unimodal functions. In f_1, f_2, f_3, f_6 and f_7, BBA has a faster convergence speed at the beginning of iteration, but the final results are worse than BEO due to its stagnation. While the convergence rates and final solutions of BEO are better than BDE, BGWO, BPSO and BPSOGSA. In f_4, BDE, BGWO, BPSO and BPSOGSA have the completely consistent performance, therefore just the convergence curve of BPSOGSA is seen in the figure. In f_5, BBA, BGWO and BPSOG converge faster than BEO, and merely the solution of BGWO is the same as BEO. BEO not only has the ability of fast convergence, but also finds the optimal solution.

From above discussion, the optimization ability, time complexity and convergence are mentioned. BEO inherits the low calculation time of EO and has strong abilities of finding the optimal solution and fast convergence through the new transfer function. Because the positions of binary algorithms only take 0 and 1, it must be pointed out that these algorithms converge quickly. But once they get caught in a local optimum, the algorithms are easily being stagnation.

Table 5. The average running time of the compared algorithms.

Function	BBA	BDE	BEO	BGWO	BPSO	BPSOGSA
f_1	1.5155	0.3468	0.325	9.5608	0.1337	1.4695
f_2	1.5239	0.3619	0.3412	9.5345	0.1505	1.488
f_3	2.1954	1.0487	1.0267	10.2841	0.8462	2.2584
f_4	1.667	0.4118	0.3846	9.5601	0.1914	1.5397
f_5	1.5974	0.4454	0.4187	9.5892	0.2258	1.5473
f_6	1.5966	0.439	0.4031	9.5737	0.2227	1.5462
f_7	1.6356	0.4471	0.4399	9.6186	0.243	1.5365
f_8	1.6221	0.4757	0.4347	9.6094	0.249	1.559
f_9	1.6474	0.4683	0.4432	9.659	0.239	1.5572
f_{10}	1.6462	0.4805	0.4433	9.6401	0.2604	1.5743
f_{11}	1.6727	0.5114	0.4737	9.6846	0.2846	1.5975
f_{12}	1.9866	0.8043	0.7861	9.9524	0.5924	1.9086
f_{13}	1.959	0.8013	0.7831	9.9202	0.574	1.8799
f_{14}	1.5353	1.6827	1.5475	1.949	1.4293	2.2927
f_{15}	0.3936	0.4406	0.32	1.4238	0.1774	1.1339
f_{16}	0.221	0.3739	0.2434	0.7266	0.1078	1.0193
f_{17}	0.2118	0.3657	0.2405	0.7218	0.1006	1.0113
f_{18}	0.227	0.3753	0.2483	0.7295	0.1111	1.022
f_{19}	0.4146	0.5181	0.3974	1.1804	0.2505	1.1806
f_{20}	0.56	0.5197	0.413	2.1291	0.2595	1.2422
f_{21}	0.4933	0.5501	0.4369	1.5414	0.2933	1.231
f_{22}	0.5477	0.6014	0.492	1.5811	0.3392	1.2797
f_{23}	0.6262	0.6776	0.5802	1.663	0.4162	1.3554
f_{24}	73.7952	73.4408	68.9057	78.7799	69.4932	70.49
f_{25}	78.097	74.1121	73.4106	82.7751	73.8172	74.8666
f_{26}	75.3479	71.4213	69.9987	79.9712	70.2471	71.4914
f_{27}	100.8156	96.3498	96.6701	107.2902	95.3859	97.3836
f_{28}	100.8077	96.6088	96.0334	105.1755	94.5929	97.2641
f_{29}	102.2629	97.157	97.0355	104.8673	94.8697	98.5768



(a) f_1. (b) f_2. (c) f_3. (d) f_4.

(e) f_5. (f) f_6. (g) f_7.

Figure 2. Convergence curves of the unimodal benchmark functions.

5. **Application for Feature Selection.** This section will use the wrapper method to implement feature selection in the UCI datasets. It adopts KNN and K-fold cross validation as the classification algorithms, the commonly used in the data mining [45, 46].

TABLE 6. The details of the simulation data sets.

Data set	Name	Instances	Attributes
Adult	Adult	48842	14
Breast Cancer Wisconsin (Diagnostic)	CancerWD	569	32
Breast Cancer Wisconsin (Prognostic)	CancerWP	198	34
Breast Cancer	Cancer	286	9
Car Evaluation	Car	1728	6
Chess (King-Rook vs. King-Pawn)	Chess	3196	36
Congressiona Voting Records	Voting	435	16
Credit Approval	Credit	690	15
Dermatology	Dermatology	366	33
Statlog (Heart)	Heart	270	13
Lymphography	Lymphography	148	18
SPECT Heart	SPECT	267	22
Waveform (Version 2)	Waveform	5000	40

TABLE 7. The errors and numbers of the compared algorithms on KNN.

Data set	BBA		BDE		BEO		BGWO		BPSO		BPSOGSA	
	Error	Number	Error	Number	Error	Number	Error	Number	Error	Number	Error	Number
Adult	0.2467	5	0.257	5.4	0.2376	1	0.2629	5.3	0.2378	1	0.2394	1.9
CancerWD	0.0605	9.2	0.0544	20.25	0.0488	9.9	0.0569	19.5	0.0498	13.15	0.0491	14
CancerWP	0.2123	13.85	0.1998	22	0.1858	11.75	0.2108	21.45	0.1955	15.55	0.1893	16.05
Cancer	0.2467	3	0.2306	4.55	0.2256	2.8	0.2334	4.85	0.2282	3.35	0.2278	3.35
Car	0.1773	1.15	0.1009	5.8	0.0904	6	0.0899	6	0.0919	6	0.0907	6
Chess	0.1174	8	0.0748	27.45	0.0374	16.35	0.0736	26.5	0.0516	19.15	0.0532	18.1
Voting	0.17	6.35	0.155	9.85	0.1506	6.15	0.1633	10.95	0.1548	6.8	0.1542	7.55
Credit	0.1928	4.45	0.1835	7.6	0.127	4.15	0.1639	7.25	0.1316	4.5	0.1326	4.45
Dermatology	0.0548	4.7	0.0215	26.7	0.0176	16.6	0.0208	25.5	0.022	19.15	0.0208	19.35
Heart	0.418	4.05	0.393	6.95	0.3762	3.75	0.3925	6.9	0.3854	4.3	0.3883	4.8
Lymphography	0.1974	4.85	0.1464	11.8	0.1325	8	0.1547	11.8	0.1405	9	0.1418	8.95
SPECT	0.2747	8.1	0.257	16.05	0.2376	7.25	0.2573	15.05	0.2526	9.55	0.2503	9.9
Waveform	0.209	9.2	0.1766	33.2	0.1602	17.1	0.1741	31.65	0.1731	22.25	0.1666	22.35

5.1. **Simulation Results.** 13 data sets are employed to validate the performance of BEO. These data sets are from UCI machine learning repository [47] and Table 6 describes the details of the data sets. They run 100 iterations and 20 times on each data set, and every algorithm has 10 particles.

5.2. **KNN Simulation Analysis.** Table 7 is the data acquired by KNN. The *error* indicates the classification error of the algorithm and the *number* is the number of features obtained by the algorithm. The red font is the lowest error and the smallest number of features in the corresponding data set. It can be drawn that BEO achieves the least classification errors in the CancerWD, CancerWP, Cancer, Chess, Voting, Credit, Dermatology, Heart, Lymphography, SPECT and Waveform. BGWO implements the optimal results in the Adult and Car. BBA and BEO excel in the selected sub-features, and they get the minimum values in 6 and 7 data sets respectively. While BPSO only performs well in the Adult. BEO accomplishes the minimum errors and features in the CancerWD, Cancer, Voting, Credit, Heart and Spect.

The compared algorithms have small classification errors in the CancerWD and Dermatology, and have large errors in the Heart. This is because the features of CancerWD and Dermatology are single and it is easy for them to establish classification models. Although there are only two types of Heart, its data structure is complex. Adult has lots of instances, but its missing data is also the reason for the high classification error.

6. **Conclusions.** Although Equilibrium optimizer has just been proposed, it has shown great performance in the benchmark functions and engineering applications. To apply EO to feature selection, this paper brings a binary version, named BEO. It retains most

of the operations of EO, so it well balances exploration and exploration. The transfer function converts the operational result of EO to [0,1] interval, therefore it has a crucial impact on the performance of BEO. This paper proposes a new transfer function to complete the value representation. The experiments show that BEO exceeds BBA, BDE, BGWO, BPSO and BPSOGSA. At the end of the paper, these algorithms complete feature selection in the UCI and BEO shows excellent performance.

For further studies, we conceive applying BEO to practical applications such as power system, text classification and so on. It is also worth studying the variant of BEO to get ideal result in a particular area.

Acknowledgment. This work is supported by the National Natural Science Foundation of China (61872085), the Natural Science Foundation of Fujian Province (2018J01638) and the Fujian Provincial Department of Science and Technology (2018Y3001).

REFERENCES

[1] A.-Q. Tian, S.-C. Chu, J.-S. Pan, H. Cui, and W.-M. Zheng, A compact pigeon-inspired optimization for maximum short-term generation mode in cascade hydroelectric power station, *Sustainability*, vol. 12, no. 3, 767, 2020.

[2] Z.-G. Du, J.-S. Pan, S.-C. Chu, H.-J. Luo, and P. Hu, Quasi-affine transformation evolutionary algorithm with communication schemes for application of rssi in wireless sensor networks, *IEEE Access*, vol. 8, pp. 8583–8594, 2020.

[3] J. M.-T. Wu, J. Zhan, and J. C.-W. Lin, An aco-based approach to mine high-utility itemsets, *Knowledge-Based Systems*, vol. 116, pp. 102–113, 2017.

[4] T.-Y. Wu, C.-M. Chen, K.-H. Wang, and J. M.-T. Wu, Security analysis and enhancement of a certificateless searchable public key encryption scheme for iiot environments, *IEEE Access*, vol. 7, pp. 49232–49239, 2019.

[5] J. C.-W. Lin, Y. Zhang, C.-H. Chen, J. M.-T. Wu, C.-M. Chen, and T.-P. Hong, A multiple objective pso-based approach for data sanitization, in *2018 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, pp. 148–151, IEEE, 2018.

[6] T.-Y. Wu, C.-M. Chen, K.-H. Wang, C. Meng, and E. K. Wang, A provably secure certificateless public key encryption with keyword search, *Journal of the Chinese Institute of Engineers*, vol. 42, no. 1, pp. 20–28, 2019.

[7] J. M.-T. Wu, J. C.-W. Lin, Y. Djenouri, P. Fournier-Viger, and Y. Zhang, A swarm-based data sanitization algorithm in privacy-preserving data mining, in *2019 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1461–1467, IEEE, Wellington, New Zealand, 10-13 June 2019.

[8] L. Zhao, M. Gai, and Y. Jia, Classification of multiple power quality disturbances based on pso-svm of hybrid kernel function, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 10, no. 1, pp. 138–146, 2019.

[9] J. Wang, B.-W. Pan, Q.-R. Wang, and Q. Ding, A chaotic key expansion algorithm based on genetic algorithm, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 10, no. 2, pp. 289–299, 2019.

[10] T.-Y. Wu, J. C.-W. Lin, Y. Zhang, and C.-H. Chen, A grid-based swarm intelligence algorithm for privacy-preserving data mining, *Applied Sciences*, vol. 9, no. 4, 774, 2019.

[11] M. Xu, Y.-P. Feng, and Z.-M. Lu, Fast feature extraction based on multi-feature classification for color image, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 10, no. 2, pp. 338–345, 2019.

[12] L. Kang, R.-S. Chen, N. Xiong, Y.-C. Chen, Y.-X. Hu, and C.-M. Chen, Selecting hyper-parameters of gaussian process regression based on non-inertial particle swarm optimization in internet of things, *IEEE Access*, vol. 7, pp. 59504–59513, 2019.

[13] H. Belhadi, K. Akli-Astouati, Y. Djenouri, J. C.-W. Lin, and J. M.-T. Wu, Gfsom: Genetic feature selection for ontology matching, in *International Conference on Genetic and Evolutionary Computing*, pp. 655–660, Springer, Changzhou, China, 14-17 December 2018.

[14] F. Zhang, T.-Y. Wu, Y. Wang, R. Xiong, G. Ding, P. Mei, and L. Liu, Application of quantum genetic optimization of lvq neural network in smart city traffic network prediction, *IEEE Access*, vol. 8, pp. 104555–104564, 2020.

[15] W. Song and C. Huang, Mining high average-utility itemsets based on particle swarm optimization, *Data Science and Pattern Recognition*, vol. 4, no. 2, pp. 19–32, 2020.

[16] T. T. Zin, P. Tin, and H. Hama, Characterizing reliability measure for internet of things by markov queue, *Data Science and Pattern Recognition*, vol. 2, no. 2, pp. 1–10, 2018.

[17] L. Kong, C.-M. Chen, H.-C. Shih, C.-W. Lin, B.-Z. He, and J.-S. Pan, An energy-aware routing protocol using cat swarm optimization for wireless sensor networks, in *Advanced Technologies, Embedded and Multimedia for Human-Centric Computing*, pp. 311–318, Springer, 2014.

[18] P. Fournier-Viger, J. C.-W. Lin, R. U. Kiran, Y. S. Koh, and R. Thomas, A survey of sequential pattern mining, *Data Science and Pattern Recognition*, vol. 1, no. 1, pp. 54–77, 2017.

[19] X. Xue and Y. Wang, Optimizing ontology alignments through a memetic algorithm using both matchfmeasure and unanimous improvement ratio, *Artificial Intelligence*, vol. 223, pp. 65–81, 2015.

[20] X. Xue and J. Liu, Collaborative ontology matching based on compact interactive evolutionary algorithm, *Knowledge-Based Systems*, vol. 137, pp. 94–103, 2017.

[21] Z.-G. Du, J.-S. Pan, S.-C. Chu, and Y.-J. Chiu, Improved binary symbiotic organism search algorithm with transfer functions for feature selection, *IEEE Access*, vol. 8, pp. 225730–225744, 2020.

[22] X. Xue and J. Chen, Using compact evolutionary tabu search algorithm for matching sensor ontologies, *Swarm and Evolutionary Computation*, vol. 48, pp. 25–30, 2019.

[23] X. Xue, Y. Wang, and W. Hao, Optimizing ontology alignments by using nsga-ii, *International Arab Journal of Information Technology*, vol. 12, no. 2, pp. 176–182, 2015.

[24] Z. Meng, J.-S. Pan, and L. Kong, Parameters with adaptive learning mechanism (palm) for the enhancement of differential evolution, *Knowledge-Based Systems*, vol. 141, pp. 92–112, 2018.

[25] S.-C. Chu, H.-C. Huang, J. F. Roddick, and J.-S. Pan, Overview of algorithms for swarm intelligence, in *International Conference on Computational Collective Intelligence*, pp. 28–41, Springer, Gdynia, Poland, 21-23 September 2011.

[26] H.-C. Huang, S.-C. Chu, J.-S. Pan, C.-Y. Huang, and B.-Y. Liao, Tabu search based multi-watermarks embedding algorithm with multiple description coding, *Information Sciences*, vol. 181, no. 16, pp. 3379–3396, 2011.

[27] S. Qin, C. Sun, Y. Jin, Y. Tan, and J. Fieldsend, Large-scale evolutionary multi-objective optimization assisted by directed sampling, *IEEE Transactions on Evolutionary Computation*, doi:10.1109/TEVC.2021.3063606, 2021.

[28] J.-S. Pan, L. Kong, T.-W. Sung, P.-W. Tsai, and V. Snášel, A clustering scheme for wireless sensor networks based on genetic algorithm and dominating set, *Journal of Internet Technology*, vol. 19, no. 4, pp. 1111–1118, 2018.

[29] T.-K. Dao, T.-S. Pan, and J.-S. Pan, A multi-objective optimal mobile robot path planning based on whale optimization algorithm, in *2016 IEEE 13th International Conference on Signal Processing (ICSP)*, pp. 337–342, IEEE, Chengdu, China, 6-10 November 2016.

[30] G. Fu, C. Sun, Y. Tan, G. Zhang, and Y. Jin, A surrogate-assisted evolutionary algorithm with random feature selection for large-scale expensive problems, in *International Conference on Parallel Problem Solving from Nature*, pp. 125–139, Springer, Leiden, Netherlands, 5-9 September 2020.

[31] X. Kou and J. Feng, Matching ontologies through compact monarch butterfly algorithm, *Journal of Network Intelligence*, vol. 5, no. 4, pp. 191–197, 2020.

[32] N. Neggaz, A. A. Ewees, M. Abd Elaziz, and M. Mafarja, Boosting salp swarm algorithm by sine cosine algorithm and disrupt operator for feature selection, *Expert Systems with Applications*, vol. 145, 103103, 2020.

[33] D. O'Neill, A. Lensen, B. Xue, and M. Zhang, Particle swarm optimisation for feature selection and weighting in high-dimensional clustering, in *2018 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1–8, IEEE, Rio de Janeiro, Brazil, July 8-13 2018.

[34] M. Mafarja, A. Qasem, A. A. Heidari, I. Aljarah, H. Faris, and S. Mirjalili, Efficient hybrid nature-inspired binary optimizers for feature selection, *Cognitive Computation*, vol. 12, no. 1, pp. 150–175, 2020.

[35] S. Arora and P. Anand, Binary butterfly optimization approaches for feature selection, *Expert Systems with Applications*, vol. 116, pp. 147–160, 2019.

[36] H. Faris, M. M. Mafarja, A. A. Heidari, I. Aljarah, A.-Z. Ala'M, S. Mirjalili, and H. Fujita, An efficient binary salp swarm algorithm with crossover scheme for feature selection problems, *Knowledge-Based Systems*, vol. 154, pp. 43–67, 2018.

[37] P. Hu, J.-S. Pan, and S.-C. Chu, improved binary grey wolf optimizer and its application for feature selection, *Knowledge-Based Systems*, vol. 195, 105746, 2020.

[38] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, Equilibrium optimizer: A novel optimization algorithm, *Knowledge-Based Systems*, vol. 191, 105190, 2020.
[39] X.-W. Xu, T.-S. Pan, P.-C. Song, C.-C. Hu, and S.-C. Chu, Multi-cluster based equilibrium optimizer algorithm with compact approach for power system network, *Journal of Network Intelligence*, vol. 6, no. 1, pp. 117–142, 2021.
[40] G. Pampara, A. P. Engelbrecht, and N. Franken, Binary differential evolution, in *2006 IEEE International Conference on Evolutionary Computation*, pp. 1873–1879, IEEE, Vancouver, BC, Canada, 16-21 July 2006.
[41] J. Kennedy and R. C. Eberhart, A discrete binary version of the particle swarm algorithm, in *1997 IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation*, vol. 5, pp. 4104–4108, IEEE, Orlando, FL, USA, 12-15 October 1997.
[42] S. Mirjalili, S. M. Mirjalili, and X.-S. Yang, Binary bat algorithm, *Neural Computing and Applications*, vol. 25, no. 3-4, pp. 663–681, 2014.
[43] E. Emary, H. M. Zawbaa, and A. E. Hassanien, Binary grey wolf optimization approaches for feature selection, *Neurocomputing*, vol. 172, pp. 371–381, 2016.
[44] S. Mirjalili, G.-G. Wang, and L. d. S. Coelho, Binary optimization using hybrid particle swarm optimization and gravitational search algorithm, *Neural Computing & Applications*, vol. 25, no. 6, pp. 1423–1435, 2014.
[45] J. Deshmukh and U. Bhosle, A study of mammogram classification using adaboost with decision tree, knn, svm and hybrid svm-knn as component classifiers, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 9, no. 3, pp. 548–557, 2018.
[46] Q. Song and T. Qin, Load forecasting based on wavelet submodel and ls-svm, *Journal of Network Intelligence*, vol. 5, no. 3, pp. 102–112, 2020.
[47] A. Asuncion and D. Newman, Uci machine learning repository, 2007.