Neural Computing and Applications
https://doi.org/10.1007/s00521-023-09299-x

S.I.: HYBRID APPROACHES TO NATURE-INSPIRED OPTIMIZATION ALGORITHMS AND THEIR APPLICATIONS

An evolutionary swarm intelligence optimizer based on probabilistic distribution

Yifei Yang[1] · Haichuan Yang[1] · Haotian Li[1] · Zheng Tang[1] · Shangce Gao[1]

Received: 13 October 2022 / Accepted: 20 October 2023

Abstract
In this study, we propose a novel approach to balance exploitation and exploration. The proposed approach is the Evolutionary Swarm Intelligence (ESI) optimizer, which combines an exploration-biased strategy with an exploitation-biased operator. The algorithm is built based on the collective behavior of biological groups, imitating their intelligence behavior. The biological evolutionary process, inspired by genetic algorithms, is applied to every individual in the algorithm. Both swarm intelligence and genetic algorithms have been widely used in practical problems, and their reliability has been proven. ESI is characterized by both spatial group intelligence behavior and temporal biological evolution. To test the performance of ESI, we used a classic test set from IEEE CEC2017 and 22 practical problems from IEEE CEC2011. The popular training tests of the dendritic neuron model were also included in the control trials. We compared ESI with some typical swarm intelligence algorithms and classic algorithms to evaluate its performance and ability to solve practical problems. The experimental results show that ESI outperforms other algorithms in terms of basic performance and the ability to solve practical problems.

Keywords Meta-heuristic algorithms · Swarm intelligence · Genetic algorithm · Dendritic neuron model · Exploitation and exploration

1 Introduction

Various algorithms have emerged over the past few decades, some based on physics phenomena, some on the summary and generalization of various sociological phenomena, and several on the study of biological aspects. These algorithms are collectively referred to as meta-heuristic algorithms (MHAs) [1]. In this study, we aim to conclude the mechanisms of the strategies of different algorithms by studying and researching some of the most classical algorithms. For instance, in particle swarm optimization (PSO) [2], the population is updated by changing the inertia coefficient and the step size of the vector. Meanwhile, the gravitational search algorithm (GSA) [3] updates the population by changing the gravitational constants, and the water wave optimization (WWO) [4] updates the population by changing the wavelength and wave-height. It is worth noting that the performances of these algorithms depend on the differences in their population update strategies. Therefore, operator is currently considered as the core of an algorithm [5]. Additionally, we investigated some improved algorithms based on MHAs and found that most of the improvements tend to add new mechanisms to the original algorithm's operator to improve the convergence of the algorithm [6–8]. Through

✉ Shangce Gao
gaosc@eng.u-toyama.ac.jp

Yifei Yang
yyf773906764@gmail.com

Haichuan Yang
yokaisen1994@gmail.com

Haotian Li
1591688699lht@gmail.com

Zheng Tang
ztang@eng.u-toyama.ac.jp

1 Faculty of Engineering, University of Toyama, Toyama 930-8555, Japan

perturbation, such as increasing the step length of the difference process, the purpose of jumping out of the local optimum can be achieved [9]. Furthermore, to enhance the exploration capability of the algorithm, researchers often add new mechanisms that favor exploration. In our study, we aim to simplify the algorithm and make it more rational and efficient by carrying out a few attempts. Additionally, we search for a moderate solution, which will be explained in the second part of this study.

Although the conclusions drawn in the previous section can be verified by comparing similar algorithms with different formulas, it does not imply that the rest of the algorithm is irrelevant. In addition to the three algorithms mentioned in the previous section, we also studied several algorithms based on swarm intelligence (SI) [10]. As we require a segmented structural design to adjust the exploitation-to-exploration ratio at a macro level, SI fits our idea perfectly. Both the Sparrow Search Algorithm (SSA) [11] and Artificial Bee Colony Algorithm (ABC) [12, 13] use similar elite strategies. We believe that dividing the algorithm into multiple parts that are independently responsible for exploitation and exploration [14], and maintaining an interchangeable relationship, is an effective strategy. In the second section of this study, we will also explain how we use this strategy. In the experimental section, the results can prove the effectiveness of using the elite strategy, as the convergence graph of the algorithm clearly shows the boundary between the exploration and exploitation processes. Therefore, we have both the basic structural part of the algorithm design and the iterative part of the algorithm design that implements the algorithm's functionality [15, 16].

We have studied several classical algorithms in search of the most suitable method among existing techniques. One such algorithm is GSA, which is a physics-inspired algorithm. In this algorithm, each individual represents a solution, and its quality represents the merit of the solution. The movement of the object is determined by gravity, with greater mass leading to greater gravitational force. The resolution of its equation shows that the better the position of the object, the slower its motion, enabling it to converge to the optimal position. ABC is a SI optimization algorithm that mimics the sociological mechanism of honeybees. It creates three types of bees: onlooker bees, scout bees, and worker bees, and evaluates location information by transforming the three bees into each other. The globally optimal solution is obtained using a greedy strategy. Cuckoo search (CS) [17] imitates the search of the cuckoo bird for a nest and sets a certain probability for the cuckoo to abandon the nest. It uses an operation that emphasizes utilization to move other individuals in the direction of better individuals to improve the population. Virtually every algorithm has a unique understanding of how to balance the combination of exploration and exploitation to ensure that we have difficulty finding a mechanism that can be called the real solution [18–20].

We have modified our perspective from finding a solution to proposing a new one. This change was influenced by research conducted by Li [21], which provided new insights into the understanding of exploration and exploitation. Building on the directions of exploitation and exploration highlighted in that study, we attempted to replicate two components that facilitate exploitation and exploration, respectively [22]. Based on the foregoing discussion, we hypothesize that an algorithm's exploitation capability is mainly dependent on its population update method, while the exploration capability is more reliant on the algorithm's overall structure. Therefore, in the design process, ESI adopted a structural design similar to that of SI [23, 24], which involves randomly moving individuals representing experts and laborers who follow the experts. ESI also endows experts and laborers with the ability to evolve based on biological evolution mechanisms [25]. The outcome of this evolution directly influences how experts move and how laborers follow.

During the experimental phase, we evaluated the performance of ESI by employing IEEE CEC2011 and CEC2017 tests. Additionally, we tested its efficacy in solving classification problems by training neurons. The results show that the ESI algorithm is competitive in terms of performance. Comparing to most algorithms with the same structure but have different population renewal method, ESI is superior in all test sets. When comparing algorithms with the similar iterative population renewal method but have different structures, it is still possible to maintain the advantage in high-dimensional problems. The main contributions of ESI can be summarized as follows: 1) In ESI, the exploration performance of SI techniques is amplified, and differential evolution is used to compensate for exploitation performance, which is theoretically a combination of the best of each [26, 27]. 2) The adaptive function with a certain probability distribution is used in the evolution to ensure that the range of exploration is increased, and the speed of exploitation is accelerated. 3) The performance of ESI was examined in two test sets of IEEE CEC and tested against neuron training. Various experiments have demonstrated the excellent performance and use of ESI for several problems.

We will present the proposed ESI in Sect. 2, provide the experimental results and discussion in Sect. 3, and conclude this study in the final section.

2 The proposed ESI

2.1 Description of ESI

Drawing inspiration from SI, we developed ESI, with an algorithmic structure that balances exploitation and exploration. ESI leverages one of the most explored mechanisms in SI, creating an exploration part, and using the most exploitable part of GA to renew the population structure [28–30]. During the exploration phase, we divided the population into two parts, experts and labor. The experts are responsible for finding new random positions to improve their ability to obtain the global optimum. When an expert finds a new location, it transforms into a labor and exploits the area. An overall evaluation of the location information is then made through exploitation by the labor to determine the exploitation value of the area. Only the exploitative part of the area is worth reserving, while the labor from other locations will transform into experts and repeat the above process. Figure 1 illustrates this process, where blue points represent individuals, and the purple depth in the top view corresponds to the position depth. The coordinate map shows the main view of the dashed position in the top view. The blue individual moves around the initial position each time the position information is updated. It is assumed that after one update, the individual's position changes to the style on the right side of Fig. 1. The red point represents the least adaptive individual, but we do not drop it immediately. Instead, the red individual is assigned a value C, representing the number of times it is allowed to be the worst individual. The worst individual will only be dropped when C satisfies the condition. The blue circle is safe because its fitness is not the worst [31].

Mimicking the process of biological evolution plays a crucial role in exploiting this ability. To obtain a completely new gene sequence, differences between different individuals within a population are taken, and the gene nodes that need evolutionary changes in the individuals themselves are found. This process is illustrated in Fig. 2. Two random individuals, *r1* and *r2*, are selected from the population through random selection, and a set of significant difference features *Difference* is obtained by subtracting the two. Since organisms do not evolve traits beyond those possessed by the species in a single genetic evolutionary process, *Difference* is used as the sequence of genes that may be generated as a variant in a single evolution. The yellow part of the figure represents genes with significant differences, and the gray part represents genes that are not significantly different, similar to recessive genes. i is a child and its genes for both colors are taken from its parent and subsequently mutated by *Difference* to obtain a new gene sequence [32]. We believe that such a design is more in line with the reality of biological evolution than the mutation strategy of the most traditional genetic algorithms. The purple color of the new genes obtained represents the mutated genes, while the orange and blue colors are the genes acquired from the parent.

The entire cycle is a repetitive process of searching for adaptive positions and mutating to genes that are adapted to the current position. The reason why the stage of genetic variation is not designed to make C increase is that we do not artificially decide whether the evolution of an organism is worth preserving or not. C increases only when the current evolution cannot adapt to the current environment to follow the intelligent factors demonstrated in natural evolution. When an individual can fully adapt to its current environment, then it can be located at a locally optimal location in the solution space. The adaptation value at this

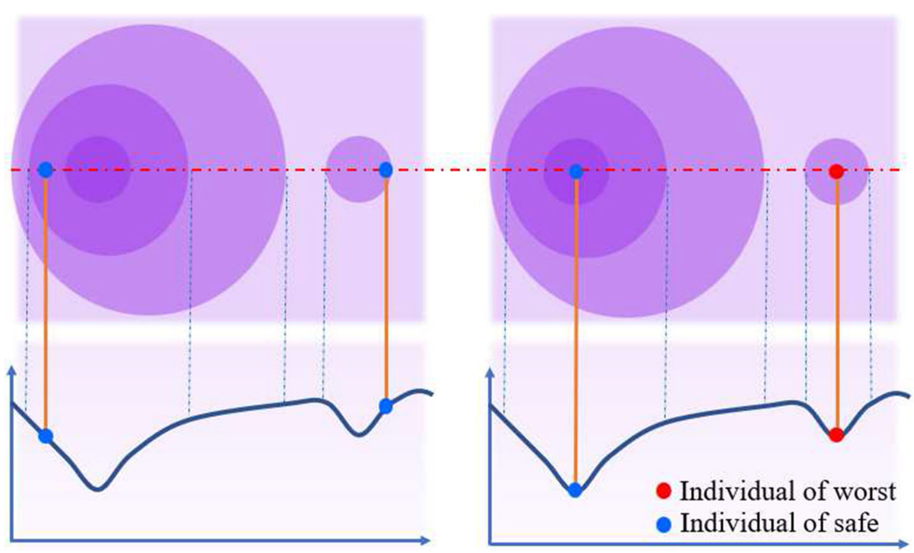

Fig. 1 Movement method display chart

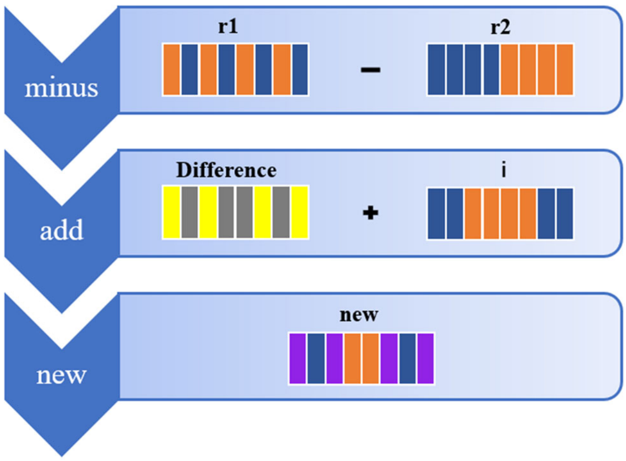

Fig. 2 Variation method display chart

location will be saved as the historical best, and each subsequent update of it will increase C until it satisfies the condition of being discarded since this individual cannot evolve further. Each historical best information is recorded and compared to select the most outstanding individual among all local optima [33]. This individual can be considered as the global optimal individual in the search space.

2.2 Iteration expression of ESI

In the exploration section, each individual in the population moves a random distance to find some locations of value. The movement process mimics the behavior pattern of a creature with intelligence, the behavior pattern of approaching a nearby friendly party. The equation is expressed as:

$$X_i = X_i + a \cdot (X_i - X_k), \tag{1}$$

where X_i is the location information of current population, a is a random number between 0 to 1, X_k is the number k location information of current population. The number k is got from a random number between 1 to maximum population size without number i. Then, according to the fitness of X_i, a roulette wheel was designed [34]. This wheel was designed as the smaller the ratio of the fitness of the current individual to the average fitness of the current population. The higher the value of the region, the higher the probability of being selected. The worse the rating of the current individual, the more difficult it is to be inherited. When the current individual is not selected, then it will make the bad number of the current individual plus one. The bad number is expressed as C in our algorithm.

$$p_i = \frac{nfit_i}{\sum_{n-1}^{S} nfit}, \tag{2}$$

where p_i is a probability distribution obtained using Eq. 2 and is used to act as a roulette wheel. The current better individual has a higher probability of being selected. $nfit$ is the fitness value obtained after evaluating the individuals updated in Eq. 1. S is the size of the population and n is the order of the ith individual.

$$\begin{aligned} X_i &= X_i + a \cdot (X_r - X_k), \\ C_i &= C_i + 1. \end{aligned} \tag{3}$$

where X_r means the random selected individual. We used Eqs. 1 and 3 to ensure whether globally superior locations exist in nearby regions for randomly generated populations. We also set a value C when the fitness of Eq. 3 is bad, C adds 1. A new individual will be randomly generated and will replace the old one when C is larger than a value that varies with dimension and population size. At this stage, the individual will prefer a more valuable position to continue the exploration. It will choose the direction of the exploration on its own based on the information from the previous step. This is because although organisms are convergent, creatures tend to choose to approach the better individuals. Therefore, during the movement phase, the whole population has a higher probability to move closer to the current better individuals in addition to each other.

It should converges as fast as possible during the exploitation process. Therefore, the exploitation strategy adopted in ESI is based on a process of rapid genetic evolution. The specific implementation is the adaptive differential operation [35]. We defined the genome that can realize a feature as a dimension, and the 30-dimensional problem can be understood as every individual possessing 30 features expressed by the genome. Subsequently, during the evolutionary process, each genome in the organism evolves based on the environment. Evolution takes the form of pruning or mutation of the genome [36]. The population will be retained in a more relaxed manner to ensure that most variants will be validated for fitness and to be more adapted to the environment. Therefore, in each generation, the evolutionary process will contain two stages. The first stage is the mutation stage, which serves as the evolutionary direction for new individuals through the genetic differences of two randomly different individuals in the whole population [37]. After the population has mutated and evolved, the mutated individuals are retained with a certain probability, while some of the original individuals can also be retained. we magnified the probability of retention to quickly obtain the final individuals obtained during the evolutionary process, making the whole population evolve faster. The equation of exploitation is expressed as:

$$CR = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$
$$F = \frac{1}{\pi\gamma\left[1 + \left(\frac{x-x_0}{\gamma}\right)^2\right]}, \tag{4}$$

where we set $\mu = 0.5$, $\sigma = 0.1$, $x_0 = 0.5$, and $\gamma = 0.1$. We used the normal and Cauchy distributions obtained under this set of parameters to generate the values of CR and F, as shown in Eq. 4. It conforms to 0.5 when CR or F exceeds 1.

$$V = X_b + F \cdot (X_{r1} - X_{r2}), \tag{5}$$

where X_b is the individual in which its fitness has been sorted from smallest to largest, F is a random number that follows the Cauchy distribution [38], and X_{r1} and X_{r2} are random populations got from X.

$$U_i(D) = \begin{cases} t_1 \cdot V_i(D) + t_2 \cdot X_i(D), & \text{if } rand > CR_i \\ X_i(D), & \text{if } rand \le CR_i. \end{cases} \tag{6}$$

where U_i is the location we must obtained, X_i is a number i individual in the current population, D is the dimension, t_1 is a random number between 0 to 1 and should be less than another random number CR that follows normal distribution [39], and t_2 is the subtraction value with 1 and t_1. This is a standard adaptive differential process. We used it for faster convergence [40].

Algorithm 1 The main procedure of ESI.

```
Result: The best individual and its fitness.
Initialization

while nFES ≤ FES do
    Calling the operation of ESI
    /*Updating the location of the population and record the number of C */
    Implement Eq. (1) and Eq. (3);
    return X and C ;
    /*Initialize a new individual */
    /*Updating the genes of the population */
    Implement Eq. (6);
    /*Update U */
    while U_best better than X_best do
        X_best = U_best
    end while
end while
```

3 Experimental results and discussion

3.1 Experiment setup

The maximum number of evaluations used in all test sets of IEEE CEC was set to $FES = D \cdot 10^4$. MATLAB was used to conduct all tests on a PC with an Intel(R) Core (TM) i7–9700 rated frequency in 3.00 GHz and 8 GB of RAM. The experiment was carried out 51 times to ensure the accuracy of the experimental results. In the dendritic neuron training, all parameters of ESI were set the same as $M = 5$, $k = 5$, and $qs = 0.5$. where M is the number of dendrites in the DNM, and k and qs are mutually a set of parameters that affect the weights. All parameters of others were set to that in their studies. Note that in the dendritic neuron training, the original data must first be normalized before it can be used. In high-dimensional problems, a certain degree of signal amplification is required to avoid the situation where all output values are normalized to zero. The maximum number of evaluations was set to $3 \cdot 10^4$.

In this section, we compare ESI with various algorithms. First, the test set of three different dimensions of IEEE CEC2017 is compared. Second, ESI is tested on 22 practical problems, and the role of using ESI to train dendritic neuron model is investigated. Finally, a discussion is made for four different adaptive schemes.

3.2 Experiment results analysis

In the comparison, under the IEEE CEC test set, we chose the Wilcoxon rank-sum statistical test to perform the evaluation [41]. Table 1 presents the full contrast of the comparison of various algorithms [42–46]. 'mean' is the average of the result of each problem running for 51 times, 'std' is the standard deviation of these 51 results, and 'W/T/L' is the number of wins, ties, and losses in all problems when ESI compares to the others. Among the algorithms chosen for comparison, to fully validate ESI, we chose an improved algorithm of each classical algorithm for comparison, which is also a part of interesting and unique algorithms. DE is one of the most successful improvement algorithms in the GA series, while DNLGSA and GLPSO were used as comparison objects in the GSA and PSO series. PSO-sono is the latest PSO-improved algorithm pitched in 2022 and is also used for comparison as one of the strongest algorithms in the SI series. Spatial information sampling (SIS) is a unique search method that differs from other classical algorithms; thus, it was chosen for the comparison. LSE is an improved algorithm of spherical evolution (SE) using the gradient descent method, and CWFS is an improved algorithm of Wingsuit flying search (WFS) using chaotic mapping [47–54]. From the

Table 1 Experiment results

IEEE CEC2017								
ESI VS	GA	DNLGSA	GLPSO	DE	PSO_sono	SIS	LSE	CWFS
D=30	30/0/0	30/0/0	30/0/0	17/3/10	20/2/8	27/0/3	25/1/4	29/1/0
D=50	30/0/0	30/0/0	29/1/0	20/2/8	18/2/10	24/1/5	20/6/4	19/3/8
D=100	30/0/0	28/2/0	24/0/6	20/5/5	14/7/9	21/4/5	20/4/6	20/2/8
ESI VS	CSO	DEPSO	GWO	IPA	PSO	SCA	SE	SSA
D=30	23/3/4	29/0/1	30/0/0	27/1/2	30/0/0	30/0/0	20/4/6	29/0/1
D=50	24/0/6	29/0/1	27/1/2	24/3/3	30/0/0	30/0/0	16/5/9	28/2/0
D=100	22/2/6	29/1/0	24/3/3	22/3/5	29/1/0	29/1/0	12/10/8	27/3/0

Table 2 Experiment results of IEEE CEC2017 on D=30

	ESI		GA		DNLGSA		GLPSO		DE	
	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
F1	3.617E-12	9.232E-12	1.083E+10	1.314E+09	3.001E+07	2.121E+08	9.855E+04	4.741E+05	**7.523E-15**	7.164E-15
F2	**2.862E+02**	1.720E+03	7.390E+33	2.105E+34	1.000E+30	1.000E+30	3.432E+24	1.436E+25	9.684E+07	6.691E+08
F3	**1.897E-10**	3.853E-10	3.334E+04	3.576E+03	1.731E+04	1.567E+04	2.191E+04	5.145E+03	3.272E+01	3.176E+01
F4	**2.037E+01**	2.725E+01	2.230E+03	2.661E+02	2.796E+02	1.355E+02	2.914E+02	9.243E+01	3.750E+01	2.831E+01
F5	**4.807E+01**	1.220E+01	2.070E+02	2.160E+01	1.486E+02	3.165E+01	1.761E+02	1.916E+01	1.790E+02	1.025E+01
F6	4.129E-05	2.480E-04	5.681E+01	5.067E+00	4.119E+01	7.302E+00	5.087E+00	2.062E+00	**2.204E-08**	2.571E-08
F7	**7.323E+01**	1.038E+01	3.609E+02	5.452E+01	2.940E+02	7.471E+01	1.620E+02	5.406E+01	2.097E+02	1.011E+01
F8	**4.310E+01**	1.130E+01	1.420E+02	2.280E+01	1.195E+02	2.962E+01	1.535E+02	3.818E+01	1.794E+02	8.088E+00
F9	1.024E+00	1.586E+00	3.416E+03	4.639E+02	3.128E+03	1.347E+03	1.409E+01	9.247E+00	**0.000E+00**	0.000E+00
F10	**2.927E+03**	4.416E+02	5.140E+03	1.032E+03	4.177E+03	6.503E+02	6.542E+03	3.351E+02	6.938E+03	2.802E+02
F11	**3.918E+01**	2.203E+01	7.041E+02	7.847E+01	3.888E+02	2.910E+02	1.322E+02	6.007E+01	5.405E+01	1.843E+01
F12	1.139E+04	8.522E+03	1.593E+09	2.770E+08	1.593E+08	1.906E+08	7.840E+06	1.328E+07	**6.397E+03**	3.928E+03
F13	2.872E+02	2.573E+02	2.085E+08	1.136E+08	1.967E+04	1.500E+04	5.504E+04	2.297E+05	**7.962E+01**	9.105E+00
F14	**4.084E+01**	1.228E+01	1.620E+05	7.924E+04	8.620E+04	1.822E+05	3.526E+04	8.101E+04	6.285E+01	4.996E+00
F15	5.106E+01	2.788E+01	1.372E+04	5.093E+03	1.019E+04	9.933E+03	8.491E+03	8.313E+03	**3.709E+01**	7.034E+00
F16	**4.361E+02**	1.884E+02	2.376E+03	3.144E+02	1.251E+03	3.277E+02	1.359E+03	2.055E+02	1.100E+03	3.959E+02
F17	**5.322E+01**	3.802E+01	7.327E+02	1.907E+02	5.910E+02	2.364E+02	2.777E+02	1.646E+02	7.627E+01	1.791E+01
F18	6.732E+01	3.526E+01	5.081E+05	2.608E+05	2.344E+05	3.939E+05	6.947E+05	7.499E+05	**3.533E+01**	4.304E+00
F19	2.657E+01	1.344E+01	1.401E+06	8.631E+05	8.822E+03	1.795E+04	9.548E+03	1.393E+04	**1.707E+01**	5.130E+00
F20	6.757E+01	7.059E+01	5.104E+02	1.273E+02	6.114E+02	2.040E+02	2.793E+02	1.377E+02	**3.935E+01**	2.282E+01
F21	**2.451E+02**	1.184E+01	4.496E+02	2.848E+01	3.237E+02	2.992E+01	3.743E+02	2.345E+01	3.707E+02	1.197E+01
F22	1.000E+02	1.366E-13	2.656E+03	6.246E+02	2.451E+03	2.217E+03	1.021E+02	2.319E+00	**1.000E+02**	6.390E-14
F23	**3.973E+02**	1.255E+01	9.362E+02	6.880E+01	6.849E+02	8.187E+01	5.930E+02	2.099E+01	5.195E+02	1.501E+01
F24	**4.654E+02**	1.272E+01	1.068E+03	6.533E+01	7.783E+02	9.283E+01	6.565E+02	2.180E+01	5.881E+02	1.201E+01
F25	3.869E+02	7.599E-01	6.718E+02	1.879E+01	5.058E+02	4.957E+01	4.330E+02	2.126E+01	**3.867E+02**	2.976E-02
F26	**1.191E+03**	5.584E+02	5.238E+03	8.234E+02	3.475E+03	1.125E+03	2.945E+03	9.361E+02	2.284E+03	3.946E+02
F27	5.147E+02	8.142E+00	1.188E+03	1.040E+02	7.673E+02	1.462E+02	6.666E+02	2.153E+01	**4.760E+02**	8.654E+00
F28	3.272E+02	4.712E+01	1.228E+03	6.816E+01	6.282E+02	1.183E+02	5.465E+02	7.722E+01	**3.132E+02**	3.653E+01
F29	**5.301E+02**	9.321E+01	2.225E+03	3.241E+02	1.355E+03	3.182E+02	8.682E+02	1.783E+02	6.386E+02	1.626E+02
F30	2.962E+03	6.911E+02	1.393E+07	5.163E+06	2.564E+06	4.197E+06	9.170E+04	1.539E+05	**2.141E+03**	8.762E+01
W/T/L	–/–/–		30/0/0		30/0/0		30/0/0		16/3/11	

comparison results data, ESI can easily win when comparing some variants of the classical algorithm. It also shows advantages when compared with variants of novel algorithms and performs well in high-dimensional tests when competing with powerful DE algorithms. Based on the comparison of algorithms with greater exploratory

Table 3 Experiment results of IEEE CEC2017 on D=30

	ESI		PSO-sono		SIS		LSE		CWFS	
	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
F1	**3.617E-12**	9.232E-12	2.237E+03	2.323E+03	4.006E+03	4.457E+03	1.712E+03	1.889E+03	4.194E+03	4.692E+03
F2	2.862E+02	1.720E+03	2.670E+16	1.366E+17	**1.049E-04**	5.491E-05	1.241E+17	2.025E+17	1.015E+08	5.666E+08
F3	**1.897E-10**	3.853E-10	3.911E-02	1.024E-01	2.973E+03	1.693E+04	5.469E+04	8.818E+03	6.862E-01	1.087E+00
F4	**2.037E+01**	2.725E+01	1.227E+02	3.887E+01	7.895E+01	3.875E+01	1.039E+02	8.301E+00	8.752E+01	1.281E+01
F5	4.807E+01	1.220E+01	**3.481E+01**	1.081E+01	1.000E+02	2.696E+01	6.486E+01	6.911E+00	6.548E+01	1.331E+01
F6	4.129E-05	2.480E-04	2.838E-01	4.869E-01	2.548E+00	2.507E+00	**1.204E-13**	2.702E-14	2.736E+00	1.957E+00
F7	7.323E+01	1.038E+01	**6.297E+01**	1.176E+01	1.374E+02	2.497E+01	1.049E+02	8.527E+00	9.886E+01	1.372E+01
F8	4.310E+01	1.130E+01	**3.304E+01**	1.128E+01	9.834E+01	2.217E+01	6.849E+01	7.629E+00	6.348E+01	1.344E+01
F9	1.024E+00	1.586E+00	1.920E+00	3.617E+00	2.761E+01	6.264E+01	**4.308E-05**	2.894E-04	1.369E+01	1.221E+01
F10	2.927E+03	4.416E+02	3.279E+03	6.237E+02	3.249E+03	6.333E+02	3.447E+03	3.169E+02	**2.417E+03**	3.761E+02
F11	**3.918E+01**	2.203E+01	1.562E+02	5.143E+01	1.462E+02	5.379E+01	8.684E+01	2.016E+01	1.063E+02	4.113E+01
F12	**1.139E+04**	8.522E+03	8.860E+04	1.747E+05	7.465E+05	7.161E+05	1.512E+06	5.168E+05	1.156E+06	1.153E+06
F13	**2.872E+02**	2.573E+02	1.470E+04	1.802E+04	8.048E+04	5.943E+04	1.972E+04	7.823E+03	4.012E+04	2.257E+04
F14	**4.084E+01**	1.228E+01	1.174E+03	4.015E+03	4.630E+03	4.918E+03	4.094E+04	2.498E+04	1.325E+03	1.951E+03
F15	**5.106E+01**	2.788E+01	5.647E+03	5.359E+03	4.189E+06	2.952E+07	6.485E+03	4.195E+03	2.750E+04	1.775E+04
F16	**4.361E+02**	1.884E+02	5.754E+02	2.333E+02	7.098E+02	2.971E+02	5.097E+02	1.150E+02	5.046E+02	1.677E+02
F17	**5.322E+01**	3.802E+01	2.000E+02	1.240E+02	2.344E+02	1.084E+02	8.949E+01	2.716E+01	1.514E+02	8.154E+01
F18	**6.732E+01**	3.526E+01	5.036E+04	6.606E+04	1.444E+05	9.140E+04	2.195E+05	9.164E+04	7.159E+04	3.975E+04
F19	**2.657E+01**	1.344E+01	6.372E+03	8.016E+03	1.307E+05	6.491E+04	9.153E+03	5.086E+03	8.852E+04	9.108E+04
F20	**6.757E+01**	7.059E+01	2.130E+02	8.639E+01	4.455E+02	1.828E+02	1.177E+02	6.079E+01	2.470E+02	8.115E+01
F21	2.451E+02	1.184E+01	**2.355E+02**	1.003E+01	2.948E+02	2.174E+01	2.594E+02	3.066E+01	2.620E+02	1.374E+01
F22	**1.000E+02**	1.366E-13	1.001E+02	6.703E-01	1.004E+02	1.014E+00	1.134E+02	4.413E+00	1.000E+02	4.060E-02
F23	3.973E+02	1.255E+01	**3.916E+02**	1.527E+01	4.431E+02	2.524E+01	4.142E+02	8.591E+00	4.113E+02	1.536E+01
F24	4.654E+02	1.272E+01	**4.541E+02**	1.058E+01	5.074E+02	2.449E+01	4.996E+02	2.283E+01	4.738E+02	1.458E+01
F25	**3.869E+02**	7.599E-01	3.970E+02	1.266E+01	3.875E+02	5.082E+00	3.871E+02	2.320E-01	3.871E+02	2.262E+00
F26	**1.191E+03**	5.584E+02	1.218E+03	4.123E+02	1.659E+03	6.587E+02	1.221E+03	3.953E+02	1.365E+03	5.930E+02
F27	**5.147E+02**	8.142E+00	5.345E+02	2.381E+01	5.199E+02	1.646E+01	5.166E+02	2.832E+00	5.218E+02	1.440E+01
F28	**3.272E+02**	4.712E+01	4.083E+02	5.780E+01	3.553E+02	5.973E+01	4.132E+02	3.300E+00	3.871E+02	4.326E+01
F29	**5.301E+02**	9.321E+01	8.008E+02	1.579E+02	7.706E+02	1.609E+02	5.597E+02	5.340E+01	6.539E+02	9.960E+01
F30	**2.962E+03**	6.911E+02	1.592E+04	8.804E+03	5.221E+05	3.810E+05	1.693E+04	5.578E+03	5.325E+05	5.060E+05
W/T/L	–/–/–		21/2/7		28/0/2		27/1/2		27/1/2	

Table 4 Experiment results of IEEE CEC2011

IEEE CEC2011				
ESI VS	GA	DNLGSA	GLPSO	DE
W/T/L	17/1/4	20/1/1	18/3/0	11/5/6
ESI VS	PSO-sono	SIS	LSE	CWFS
W/T/L	11/3/8	13/1/8	7/3/12	10/5/7

power, ESI gradually loses as the dimension rises. This is because the operation of differential selection is influenced by the size of the dimension (i.e., the higher dimension, the easier it is to fall into a local optimum). For example, the CWFS algorithm, which is a new and improved algorithm based on WFS, has a complex design of chaotic systems. Therefore, ESI is challenging when compared comparing with it. Moreover, DE is a very popular algorithm in the field of algorithm improvement, and it is not an easy task to challenge it. The results also show that ESI did not easily win. Tables 2 and 3 list the data of ESI with each algorithm at 30 dimensions for 30 problems. The bolded data represent them as individuals with the best mean performance in the current problem.

In the comparison, under the practical problems test, we selected 22 problems from IEEE CEC2011 to comprehensively test the performance of ESI on the actual problems. Table 4 lists the specific comparison information. The results show that ESI does not perform well as it did in the IEEE CEC2017 test set in terms of actual problems. This is because most of the IEEE CEC2011 real-world problem test sets are biased toward problems that focus on exploration. Additionally, ESI, which uses adaptive

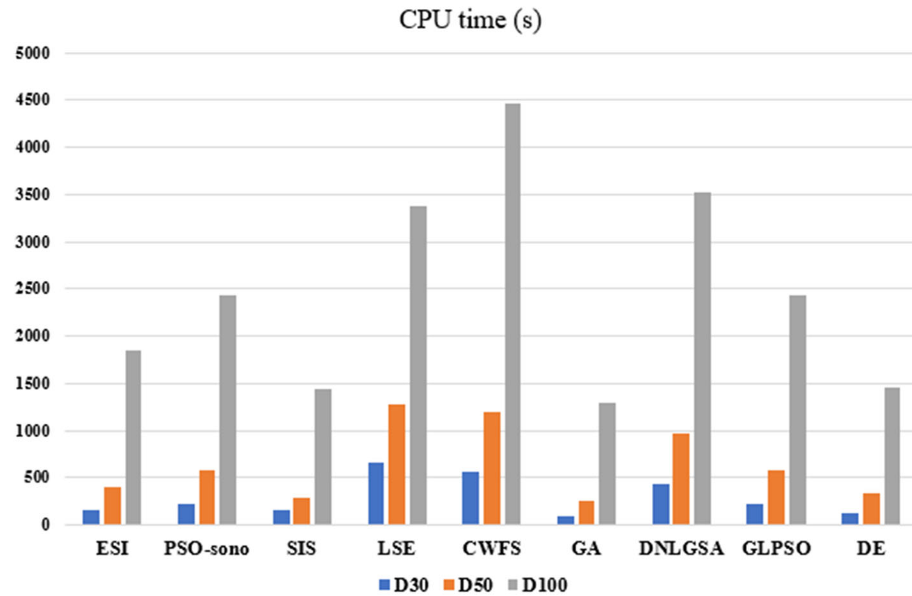

Fig. 3 Bar graph of CPU running time consumed by all tested algorithms on IEEE CEC2017 functions with 30, 50, 100 dimensions

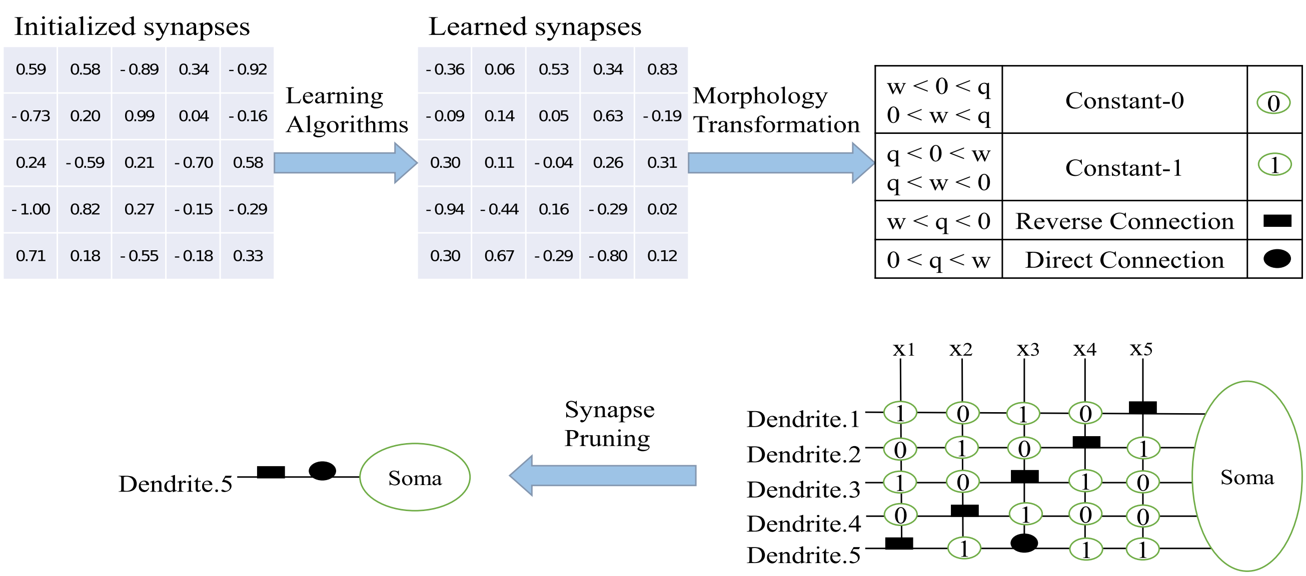

Fig. 4 Training process of DNM

differential evolution, has exploited in half of the evaluation counts, making it perform less than that of exploration-focused algorithms on problems that require more exploration. ESI has two advantages in the overall set of real-world problems tested. First, the SI structure is suitable for solving real-world problems or for dealing with complex multi-peaked problems. Second, the design of ESI effectively alleviates a series of problems caused by a large problem dimension in differential evolution.

Figure 3 shows the time required to run each algorithm on IEEE CEC2017 for the comparison. The data are the average results obtained once for each algorithm run thirty times on the same device. The time consumption of ESI is higher than that of GA because of the use of the SI mechanism, while the time consumption of ESI is lower than that of other SI algorithms because of the simplified way of moving in the SI algorithm.

3.3 Experimental data and comparison results on dendritic neuron model training

The dendritic neuron model (DNM) is a single neuron model with synapses and dendrites [55]. It has been proven valid for solution of classification problems. DNM is

Table 5 Experimental results of accuracy in test suit

	ESI		SEDE		CJADE		SCJADE	
	Mean	std	Mean	std	Mean	std	Mean	std
Tic-tac-toe	**6.72E-01**	3.15E-02	5.674E-01	1.297E-01	5.92E-01	1.08E-01	6.191E-01	9.074E-02
Heart	**8.07E-01**	1.25E-02	6.732E-01	1.163E-01	7.77E-01	7.13E-02	7.732E-01	7.575E-02
Australia	**8.39E-01**	1.45E-02	5.923E-01	1.779E-01	6.84E-01	1.77E-01	6.519E-01	1.459E-01
Congress	**9.17E-01**	1.89E-02	8.085E-01	1.349E-01	7.93E-01	1.46E-01	8.215E-01	1.440E-01
Vote	**8.94E-01**	2.76E-02	7.831E-01	1.437E-01	7.74E-01	1.26E-01	8.211E-01	1.148E-01
Spect	7.38E-01	5.57E-02	8.184E-01	4.471E-02	8.42E-01	7.23E-02	**8.911E-01**	4.811E-02
German	**7.10E-01**	2.10E-02	4.333E-01	1.623E-01	3.51E-01	9.82E-02	3.582E-01	1.111E-01
Breast	**9.20E-01**	1.62E-02	6.587E-01	5.611E-02	6.35E-01	1.13E-16	6.400E-01	2.578E-02
Ionosphere	**8.22E-01**	6.36E-02	7.020E-01	2.347E-01	1.79E-01	2.82E-17	1.788E-01	2.823E-17
KrVsKpEW	**7.46E-01**	3.01E-02	6.626E-01	1.462E-01	4.79E-01	2.87E-02	4.834E-01	4.230E-02
W/T/L	–/–/–		9/0/1		9/0/1		9/0/1	

composed of synaptic, dendritic, membrane, and soma layers. Figure 4 shows its structure and method of action in detail. In terms of functional implementation, DNM uses a sigmoid function at the synaptic layer to process external signals, generating only one of results that inhibition or excitation. The sigmoid function is expressed in Eq. 7, where $X_{i,j}$ is the result of inhibition or excitation (close to 0 or 1), $-k$ is a magnification, $w_{i,j}$ and $q_{i,j}$ are the matrices obtained from population in the algorithm, and T_i is the matrix of m dimensions reconstructed from the training data [56].

$$X_{i,j} = \frac{1}{1 + e^{-k\cdot\left(w_{i,j}\cdot T_i - q_{i,j}\right)}}, \quad i = [1, 2, ..., D], \quad j = [1, 2, ..., m]. \tag{7}$$

where w and q are the half selection from the population in the algorithm. The block means a result between w and 0. w is lower than 0 when the block is black; otherwise, it means that w is bigger than 0. After the algorithm trains the DNM, a new group of w and q will burn. DNM employs Eq. 7 to obtain eigenvalues, and the number of X is determined by M. During morphology transformation, the dendrite, which has one individual of zero, will be pruned. The reserved dendrite will be summed into soma.

The GA-inspired DE is a powerful algorithm; however, we did not compare the improved DE algorithm in the test set. Therefore, we chose three popular DE improvement algorithms for neuron training, including SEDE, CJADE, and SCJADE. In this study, the problem sets used to validate the DNM training results are binary classification problems. Based on the data sheet , as shown in Table 5, ESI achieved an overwhelming advantage. We believe that the reason why ESI performed weaker than the other three algorithms on the 'Spect' problem is that the problem itself is more suitable for algorithms with strong exploitation capabilities. Since ESI is an algorithm that balances exploitation and exploration, it does not outperform SCJADE and other DE-improved algorithms in terms of exploitation capability [57–60].

3.4 Image analysis

In Fig. 5, we averaged three selected problems from IEEE CEC2017 test set. ESI greatly performs in single-peak problems. This is because it generated by its part of imitating GA [61]. However, the broader global search structure makes it difficult to converge to 0 for simple problems. In the case of multi-peak problems, ESI performs well because of the structure designed with SI technology [62]. ESI has higher stability relative to most algorithms in 51 experiments because of the higher convergence ability of differential evolution. The most intuitive manifestation of this is that in the box plot graphs, ESI has fewer extreme values, and the mean is closer to the median. When comparing the convergence plots, ESI continues to converge on problem 3 and the final value is smaller compared to other algorithms. Therefore, ESI has a sufficiently strong exploitation capability. In the box plot graphs in the same problem, in logarithmic coordinates, the results of ESI show a good stability. Even the occasional extreme values produced are much lower than those of other algorithms. The excellent exploration ability of the ESI was demonstrated in problem 30. The stable and strong performance of ESI stands out when some algorithms have difficulty locating the global optimum. It can be shown that ESI is an algorithm with both sufficient exploitation and exploration capabilities.

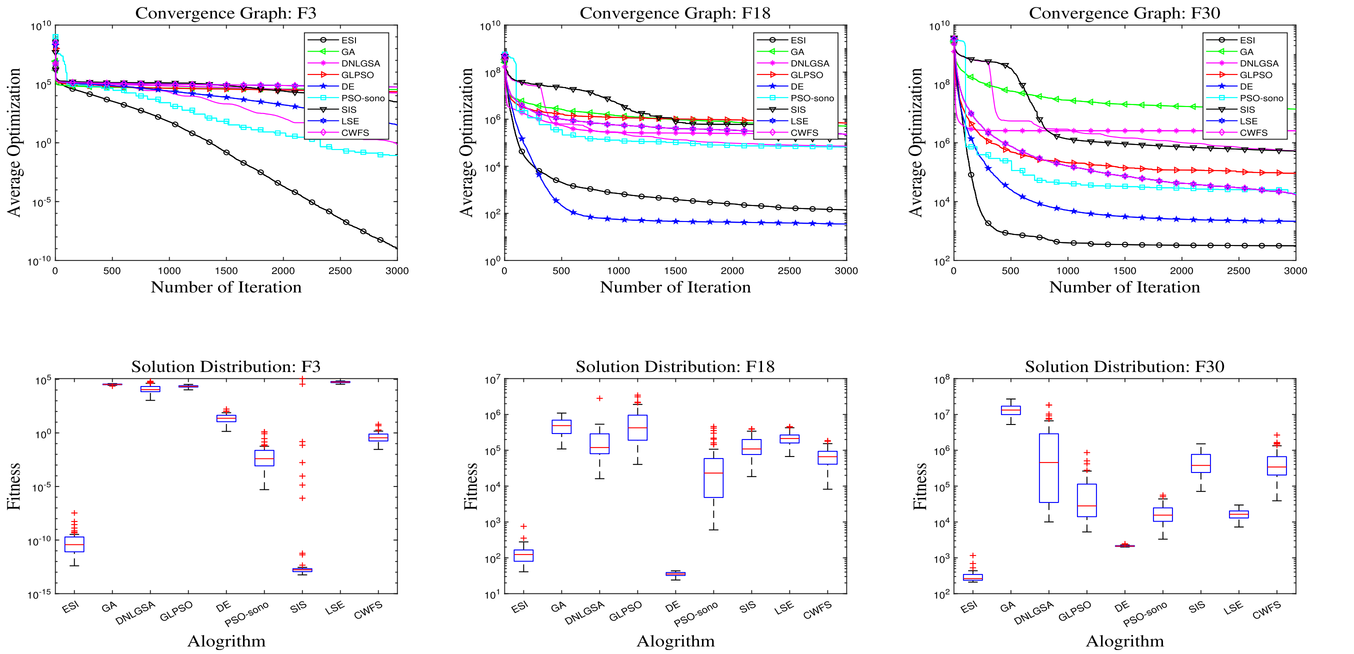

Fig. 5 Convergence graphs and Boxplot graphs of IEEE CEC2017 on D=30

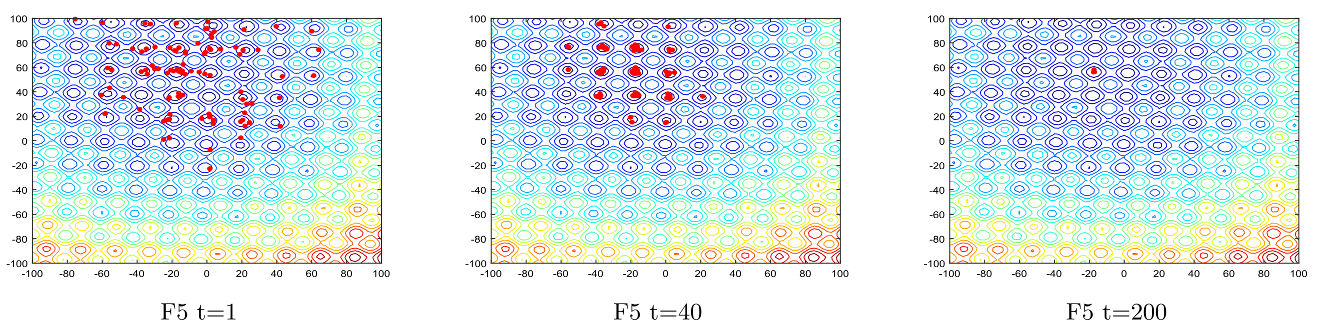

Fig. 6 Search history of individuals of ESI in 2 dimensions in IEEE CEC2017

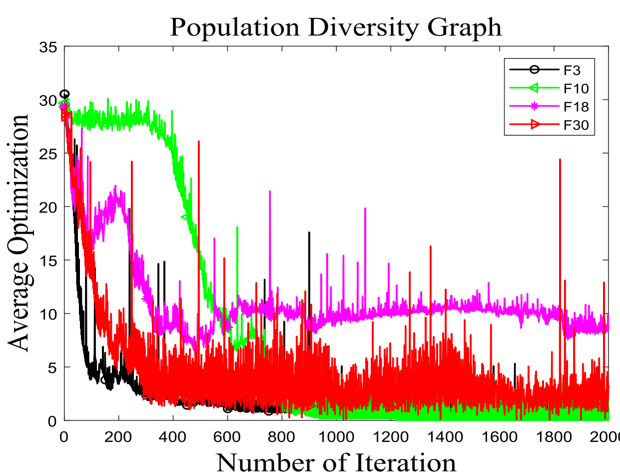

Fig. 7 Population diversity of ESI in 30 dimensions in IEEE CEC2017

Table 6 Discussion of adaptive

	ESI	ESI(F1)	ESI(F2)	ESI(F3)
W/T/L	–/–/–	16/14/0	7/20/3	20/5/5

In Fig. 6, we select one problem from IEEE CEC2017 test set to display the two-dimensional landscape search trajectory, where t is the number of iterations. Each individual of the population will be randomly distributed at positions within the upper and lower bounds because of the initialization. In problems 5 of the multi-peaked problem, the individuals are randomly distributed among each local optimum when t = 1. When t = 40, Since each individual can only move within a sufficiently small range, individuals are evolving to better fit their current position. This is also consistent with the description of ESI gene evolution.

When t = 200, ESI can find the global optimal position from multiple local optima. Thus, while the individual cannot evolve to be more adapted to the current environment, it moves in a manner that makes it difficult to find a more promising location. Therefore, we can assume that ESI has a good balance with exploitation and exploration and find the global optimal position when the number of iterations is sufficient.

In Fig. 7, the effect of the interconversion of exploration and exploitation in ESI can be visualized. First, note that the number of iterations is less than most algorithms because of the special mechanism of ESI. For example, in 300,000 evaluations, the population iteration of ESI is only 2,000 generations at the same number of evaluations. Based on the figure, the initial phase of ESI maintains a good exploratory power, which is similar to the values of most algorithms. In the face of exploitation-oriented problems, the algorithm actively boosts the exploitation part but also maintains a certain degree of exploration. In ESI, the SI mechanism works well for problems that focus more on exploration capabilities. In problem 10, this is a problem that requires more exploratory skills; thus, ESI maintains a high diversity in this problem for a longer period. In problem 18, ESI maintains a more stable moderate amount of diversity, indicating the desire to explore and exploit this problem. The other two problems are more closely related to a general continuous problem of small-scale exploration followed by continuous exploitation. Therefore, ESI behaves as what the problem needs [63].

3.5 Discussion

Table 6 presents the comparison of the values under three adaptive designs, all using the IEEE CEC2017 test set of dimensions 30. We adjusted the upper limit of C such that the effect of adaptation was amplified to make the test results more clearly reflect the differences between the adaptive schemes. ESI refers to the version we adopted. ESI (F1) and ESI (F2) are the results obtained under F from the standard normal distribution and then by $F \cdot 0.5$ and $F \cdot 1.5$. And ESI (F3) is the result of replacing the Cauchy distribution using the power-law distribution. It is clear from the results that the choice of a classical mathematical approach is superior rather than using only a linear function with accelerated convergence. For the aspect of probabilistic selection, it is not a good strategy to use a greater probability of retaining the variant individuals in terms of results. Therefore, we chose to use 0.5 as the normal distribution of the location information.

4 Conclusion

In this study, the new approach that combines the structure and iterations of different algorithms has yielded good results in practice. Using ESI, our choices and innovations are more balanced, and not biased toward exploration or exploitation. Similarly, if the exploration-biased structure, such as the SI structure, was combined with the exploration-biased iteration, then it could be argued that this would give it a significant advantage in solving discrete optimization problems. With the performance in the IEEE CEC test set, it can be concluded that ESI is inferior to PSO-sono in terms of exploration capability but is superior to it in terms of exploitation capability. Moreover, ESI is inferior to DE in exploitation capability, but superior to it in exploration capability. ESI is a more balanced algorithm according to the concept of no free lunch theorems [64]. Although this does not solve the NP-hard problem, it provides a solution in another direction by selecting the algorithm suitable for solving the problem through the difference in exploitation and exploration capabilities of different algorithms. Therefore, the innovation presented in this study lies not only in proposing a new and more balanced algorithm but also in providing a new idea to improve the algorithm. For ESI, there is still plenty of room for improvement in the way it is updated behaviorally and in the selection of individual evolution. Further accuracy improvements can also be made using the improved ESI in terms of solutions for practical problems. It can have a certain effect on neural network training since ESI can train DNM [65]. Finally, it would be a good research direction to combine ESI with neural networks.

Author Contributions Yifei Yang: Conceptualization, Methodology, Software, Investigation, Validation, Writing - original draft, Writing - review & editing. Haichuan Yang: Visualization, Investigation, Validation. Haotian Li: Conceptualization, Writing - review & editing. Zheng Tang: Methodology, Software, Supervision, Writing - review & editing. Shangce Gao: Methodology, Software, Supervision, Writing - review & editing. All authors read and approved the final manuscript.

Funding This research was partially supported by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant JP22H03643, Japan Science and Technology Agency (JST) Support for Pioneering Research Initiated by the Next Generation (SPRING) under Grant JPMJSP2145, and JST through the Establishment of University Fellowships towards the Creation of Science Technology Innovation under Grant JPMJFS2115.

Availability of data and material Related data and material can be found at https://toyamaailab.github.io.

Declarations

Ethics approval and consent to participate Not applicable

Consent for publication Not applicable

Conflict of interest The authors declare no conflict of interest.

References

1. Abdel-Basset M, Abdel-Fatah L, Sangaiah AK (2018) Metaheuristic algorithms: a comprehensive review. In: Computational intelligence for multimedia big data on the cloud with engineering applications, pp 185–231. https://doi.org/10.1016/B978-0-12-813314-9.00010-4
2. Poli R, Kennedy J, Blackwell T (2007) Particle swarm optimization. Swarm Intell 1(1):33–57. https://doi.org/10.1007/s11721-007-0002-0
3. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179(13):2232–2248. https://doi.org/10.1016/j.ins.2009.03.004
4. Zheng Y-J (2015) Water wave optimization: a new nature-inspired metaheuristic. Comput Oper Res 55:1–11. https://doi.org/10.1016/j.cor.2014.10.008
5. Xu Z, Gao S, Yang H, Lei Z (2021) SCJADE: yet another state-of-the-art differential evolution algorithm. IEEJ Trans Electr Electron Eng 16(4):644–646. https://doi.org/10.1002/tee.23340
6. Li X, Yang H, Li J, Wang Y, Gao S (2021) A novel distributed gravitational search algorithm with multi-layered information interaction. IEEE Access 9:166552–166565. https://doi.org/10.1109/ACCESS.2021.3136239
7. Azzougui Y, Recioui A, Mansouri A (2019) Pmu optimal placement in wide area monitoring systems using grey wolf optimization technique. Alger J Signals Syst 4(1):1–7
8. Azzougui Y, Recioui A (2020) Application of the moth flame optimisation to the selective harmonic elimination in multilevel converters. Int J Smart Grid Green Commun 2(1):1–18
9. Li J, Yang L, Yi J, Yang H, Todo Y, Gao S (2022) A simple but efficient ranking-based differential evolution. IEICE Trans Inf Syst 105(1):189–192. https://doi.org/10.1587/transinf.2021EDL8053
10. Karaboga D, Akay B (2009) A survey: algorithms simulating bee swarm intelligence. Artif Intell Rev 31(1):61–85. https://doi.org/10.1007/s10462-009-9127-4
11. Xue J, Shen B (2020) A novel swarm intelligence optimization approach: sparrow search algorithm. Syst Sci Control Eng 8(1):22–34. https://doi.org/10.1080/21642583.2019.1708830
12. Karaboga D (2010) Artificial bee colony algorithm. Scholarpedia 5(3):6915. https://doi.org/10.4249/scholarpedia.6915
13. Kaya E, Gorkemli B, Akay B, Karaboga D (2022) A review on the studies employing artificial bee colony algorithm to solve combinatorial optimization problems. Eng Appl Artif Intell 115:105311. https://doi.org/10.1016/j.engappai.2022.105311
14. Črepinšek M, Liu S-H, Mernik M (2013) Exploration and exploitation in evolutionary algorithms: a survey. ACM Comput Surv 45(3):1–33. https://doi.org/10.1145/2480741.2480752
15. Tang D (2019) Spherical evolution for solving continuous optimization problems. Appl Soft Comput 81:105499. https://doi.org/10.1016/j.asoc.2019.105499
16. Al-Betar MA, Alyasseri ZAA, Awadallah MA, Abu Doush I (2021) Coronavirus herd immunity optimizer (chio). Neural Comput Appl 33:5011–5042. https://doi.org/10.1007/s00521-020-05296-6
17. Yang X-S, Deb S (2014) Cuckoo search: recent advances and applications. Neural Comput Appl 24(1):169–174. https://doi.org/10.1007/s00521-013-1367-1
18. Yang X-S, Deb S (2009) Cuckoo Search via Lévy flights. In: 2009 world congress on nature & biologically inspired computing (NaBIC), pp 210–214. https://doi.org/10.1109/NABIC.2009.5393690. IEEE
19. Akay B, Karaboga D (2012) A modified Artificial Bee Colony algorithm for real-parameter optimization. Inf Sci 192:120–142. https://doi.org/10.1016/j.ins.2010.07.015
20. Recioui A (2020) Capacity optimization of mimo systems involving conformal antenna arrays using a search group algorithm. Alger J Signals Syst 5(4):209–214
21. Li X, Li J, Yang H, Wang Y, Gao S (2022) Population interaction network in representative differential evolution algorithms: Power-law outperforms poisson distribution. Physica A 603:127764. https://doi.org/10.1016/j.physa.2022.127764
22. Luger J, Raisch S, Schimmer M (2018) Dynamic balancing of exploration and exploitation: the contingent benefits of ambidexterity. Organ Sci 29(3):449–470. https://doi.org/10.1287/orsc.2017.1189
23. Oliveto PS, Witt C (2015) Improved time complexity analysis of the Simple Genetic Algorithm. Theoret Comput Sci 605:21–41. https://doi.org/10.1016/j.tcs.2015.01.002
24. Desale S, Rasool A, Andhale S, Rane P (2015) Heuristic and Meta-heuristic algorithms and their relevance to the real world: a survey. Int J Comput Eng Res Trends 351(5):2349–7084. https://doi.org/10.22362/ijcert
25. Pagel M (1999) Inferring the historical patterns of biological evolution. Nature 401(6756):877–884. https://doi.org/10.1038/44766
26. Zhang J, Sanderson AC (2009) Jade: Adaptive differential evolution with optional external archive. IEEE Trans Evol Comput 13(5):945–958. https://doi.org/10.1109/TEVC.2009.2014613
27. Morales-Castañeda B, Zaldivar D, Cuevas E, Fausto F, Rodríguez A (2020) A better balance in metaheuristic algorithms: does it exist? Swarm Evol Comput 54:100671. https://doi.org/10.1016/j.swevo.2020.100671
28. Lawson DJ, Davies NM, Haworth S, Ashraf B, Howe L, Crawford A, Hemani G, Davey Smith G, Timpson NJ (2020) Is population structure in the genetic biobank era irrelevant, a challenge, or an opportunity? Hum Genet 139(1):23–41. https://doi.org/10.1007/s00439-019-02014-8
29. Wang Z, Gao S, Zhou M, Sato S, Cheng J, Wang J (2022) Information-theory-based nondominated sorting ant colony optimization for multiobjective feature selection in classification. IEEE Trans Cybern. https://doi.org/10.1109/TCYB.2022.3185554
30. Recioui A, Merdj M, Kamli A (2021) Optimal placement of fault passage indicators in distribution networks using genetic algorithms. Alger J Signals Syst 6(4):181–191
31. Xue Y, Jiang J, Zhao B, Ma T (2018) A self-adaptive artificial bee colony algorithm based on global best for global optimization. Soft Comput 22(9):2935–2952. https://doi.org/10.1007/s00500-017-2547-1
32. Wegener I (2001) Theoretical aspects of evolutionary algorithms. In: International colloquium on automata, languages, and programming, pp 64–78. https://doi.org/10.1007/3-540-48224-5_6. Springer
33. Zhou Z, Li F, Zhu H, Xie H, Abawajy JH, Chowdhury MU (2020) An improved genetic algorithm using greedy strategy toward task scheduling optimization in cloud environments. Neural Comput Appl 32(6):1531–1541. https://doi.org/10.1007/s00521-019-04119-7
34. Qian W, Chai J, Xu Z, Zhang Z (2018) Differential evolution algorithm with multiple mutation strategies based on roulette

wheel selection. Appl Intell 48(10):3612–3629. https://doi.org/10.1007/s10489-018-1153-y
35. Al-Dabbagh RD, Neri F, Idris N, Baba MS (2018) Algorithmic design issues in adaptive differential evolution schemes: review and taxonomy. Swarm Evol Comput 43:284–311. https://doi.org/10.1016/j.swevo.2018.03.008
36. Yang C, An Z, Li C, Diao B, Xu Y (2019) Multi-objective pruning for CNNs using genetic algorithm. In: International conference on artificial neural networks, pp. 299–305. https://doi.org/10.1007/978-3-030-30484-3_25. Springer
37. Yu Y, Gao S, Wang Y, Todo Y (2019) Global optimum-based search differential evolution. IEEE/CAA J Automatica Sinica 6(2):379–394. https://doi.org/10.1109/JAS.2019.1911378
38. Arnold BC, Beaver RJ (2000) The skew-Cauchy distribution. Stat Probabil Lett 49(3):285–290. https://doi.org/10.1016/S0167-7152(00)00059-6
39. Ahsanullah M, Kibria B, Shakil M (2014) Normal Distribution. In: Normal and student st. distributions and their applications, pp 7–50. Atlantis Press, Paris. https://doi.org/10.2991/978-94-6239-061-4_2
40. Xue Y, Xue B, Zhang M (2019) Self-adaptive particle swarm optimization for large-scale feature selection in classification. ACM Trans Knowl Discov Data 13(5):1–27. https://doi.org/10.1145/3340848
41. Woolson RF (2007) Wilcoxon signed-rank test. Wiley Encyclopedia of Clinical Trials, 1–3. https://doi.org/10.1002/9780471462422.eoct979
42. Bahrami M, Bozorg-Haddad O, Chu X (2018) Cat swarm optimization (cso) algorithm. Adv Optim Nat Inspired Algorithm. https://doi.org/10.1007/978-981-10-5221-7_2
43. Zhang W-J, Xie X-F (2003) Depso: hybrid particle swarm with differential evolution operator. In: SMC'03 conference proceedings. 2003 IEEE International conference on systems, man and cybernetics. conference theme - system security and assurance (Cat. No.03CH37483), vol 4, pp 3816–38214. https://doi.org/10.1109/ICSMC.2003.1244483
44. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
45. Aslan S, Demirci S (2020) Immune plasma algorithm: a novel meta-heuristic for optimization problems. IEEE Access 8:220227–220245. https://doi.org/10.1109/ACCESS.2020.3043174
46. Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. Knowl-Based Syst 96:120–133. https://doi.org/10.1016/j.knosys.2015.12.022
47. Yang H, Tao S, Zhang Z, Cai Z, Gao S (2022) Spatial information sampling: another feedback mechanism of realising adaptive parameter control in meta-heuristic algorithms. Int J Bio-Inspired Comput 19(1):48–58. https://doi.org/10.1504/IJBIC.2022.120751
48. Zhang A, Sun G, Ren J, Li X, Wang Z, Jia X (2018) A dynamic neighborhood learning-based gravitational search algorithm. IEEE Trans Cybern 48(1):436–447. https://doi.org/10.1109/TCYB.2016.2641986
49. Gong Y-J, Li J-J, Zhou Y, Li Y, Chung HS-H, Shi Y-H, Zhang J (2016) Genetic learning particle swarm optimization. IEEE Trans Cybern 46(10):2277–2290. https://doi.org/10.1109/TCYB.2015.2475174
50. Liu J, Lampinen J (2005) A fuzzy adaptive differential evolution algorithm. Soft Comput 9(6):448–462. https://doi.org/10.1007/s00500-004-0363-x
51. Meng Z, Zhong Y, Mao G, Liang Y (2022) PSO-sono: a novel PSO variant for single-objective numerical optimization. Inf Sci 586:176–191. https://doi.org/10.1016/j.ins.2021.11.076
52. Yang H, Yu Y, Cheng J, Lei Z, Cai Z, Zhang Z, Gao S (2022) An intelligent metaphor-free spatial information sampling algorithm for balancing exploitation and exploration. Knowle-Based Syst. https://doi.org/10.1016/j.knosys.2022.109081
53. Yang H, Gao S, Wang R-L, Todo Y (2021) A ladder spherical evolution search algorithm. IEICE Trans Inf Syst 104(3):461–464. https://doi.org/10.1587/transinf.2020EDL8102
54. Covic N, Lacevic B (2020) Wingsuit flying search-a novel global optimization algorithm. IEEE Access 8:53883–53900. https://doi.org/10.1109/ACCESS.2020.2981196
55. Luo X, Wen X, Zhou M, Abusorrah A, Huang L (2022) Decision-tree-initialized dendritic neuron model for fast and accurate data classification. IEEE Trans Neural Netw Learn Syst 33(9):4173–4183. https://doi.org/10.1109/TNNLS.2021.3055991
56. Lan G (2020) First-order and stochastic optimization methods for machine learning. Springer, Paris. https://doi.org/10.1007/978-3-030-39568-1
57. Gao S, Zhou M, Wang Y, Cheng J, Yachi H, Wang J (2019) Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction. IEEE Trans Neural Netw Learn Syst 30(2):601–614. https://doi.org/10.1109/TNNLS.2018.2846646
58. He H, Gao S, Jin T, Sato S, Zhang X (2021) A seasonal-trend decomposition-based dendritic neuron model for financial time series prediction. Appl Soft Comput 108:107488. https://doi.org/10.1016/j.asoc.2021.107488
59. Zhou T, Gao S, Wang J, Chu C, Todo Y, Tang Z (2016) Financial time series prediction using a dendritic neuron model. Knowl-Based Syst 105:214–224. https://doi.org/10.1016/j.knosys.2016.05.031
60. Wang S, Yu Y, Zou L, Li S, Yu H, Todo Y, Gao S (2020) A novel median dendritic neuron model for prediction. IEEE Access 8:192339–192351. https://doi.org/10.1109/ACCESS.2020.3032948
61. Keane AJ (1995) Genetic algorithm optimization of multi-peak problems: studies in convergence and robustness. Artif Intell Eng 9(2):75–83. https://doi.org/10.1016/0954-1810(95)95751-Q
62. Rana S, Jasola S, Kumar R (2011) A review on particle swarm optimization algorithms and their applications to data clustering. Artif Intell Rev 35(3):211–222. https://doi.org/10.1007/s10462-010-9191-9
63. Zhu KQ, Liu Z (2004) Population diversity in permutation-based genetic algorithm. In: European conference on machine learning, pp 537–547. https://doi.org/10.1007/978-3-540-30115-8_49. Springer
64. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82. https://doi.org/10.1109/4235.585893
65. Wang Z, Gao S, Wang J, Yang H, Todo Y (2020) A dendritic neuron model with adaptive synapses trained by differential evolution algorithm. Comput Intell Neurosci. https://doi.org/10.1155/2020/2710561