Journal of Nonlinear and Convex Analysis
Volume 26, Number 6, 2025, 1759–1783

Yokohama Publishers
ISSN 1880-5221 ONLINE JOURNAL

ADAPTIVE NEURAL NETWORK OPTIMIZATION USING INFORMATION ENTROPY FOR DYNAMIC PARAMETER TUNING

QIN GAO, JINGSONG ZHANG, XUEDING TAO, WEISHANG GAO*, LIJIE SUN, YUE CHEN, AND PENG HE

Abstract. The adaptability of neural network tuning to specific problems and sample conditions is one of the key factors influencing regression performance. This study aimed to design an evolutionary optimization algorithm with a dynamic adjustment mechanism of population size and search area to address the tuning complexity of neural networks. The information entropy calculated using the relative magnitude of fitness and fitness increment adaptively carried out a trade-off between exploration and exploitation in the optimization algorithm. Entropy coefficients were modified in a timely manner when calculating the size and distribution area of the new generation. Consequently, the distribution trend and population density of the next generation were effectively controlled. In addition, a shrinking mechanism based on the shrinkage coefficient was suggested to redefine a smaller but more dominant region for the next generation. The performance of the proposed method compared with the existing techniques indicated that the new optimization algorithm was the first to achieve a more accurate solution. Thus, the structural parameters of the neural network could be adjusted efficiently using the improved optimization algorithm. Despite its high computational complexity and cost, the proposed approach could effectively improve algorithm performance. Experiments confirmed that the adjusted neural network achieved excellent results in the case of small-sample learning.

1. Introduction

Over the last few years, artificial intelligence [19,21] has proven to be a promising approach to solving complex engineering problems. Intelligent optimization [29,33] and pattern recognition technology [13, 32] have been widely used in industrial processing [4, 9], agricultural production [17, 27], transportation [3, 8], commercial operation [7,31], social services [14,15], and other fields, and are developing rapidly. Especially, artificial neural networks [25,28](ANN) and support vector machines [6, 16](SVM) have received extensive attention in data mining and modeling as classical pattern recognition algorithms. However, the parameters of pattern recognition algorithms need to be adjusted appropriately [22, 30] to adapt them to different complex problems, but a proper adjustment is difficult to obtain because of the

2020 *Mathematics Subject Classification.* 90C26, 68T07.

Key words and phrases. Entropy-based exploration, evolutionary optimization, machine learning, neural network, swarm intelligence.

This study was supported by the Taizhou Industrial Science and Technology Plan Project (Grant No. 21gyb23), the Taizhou Science and Technology Bureau Innovation Voucher Service Project (Grant No. CXQ-2021-023), the National College Student Innovation and Entrepreneurship Training Program (Grant No. 202310350043), and the Taizhou University Scientific Research Fund.

*Corresponding author.

complexity and variability of data features. Mostly, extensive adjustments need to be first estimated by professionals based on experience and knowledge and then used in a specific modeling process after several experiments and comparisons. The obtained parameters differ significantly from the optimal ones due to low efficiency and limited time for manual adjustment.

An optimization algorithm can be used to adjust the parameters by rapidly iterating through a wider range of feasible parameters to improve the efficiency and effect of parameter optimization [20, 34]. To date, swarm intelligence optimization algorithms [10, 12] have gained immense attention and have been applied widely because of their advantages in complex optimization [11, 24]. In most cases, the optimization model of the recognition algorithm is relatively complex, and using the mathematical analytical optimization method to solve the problem is difficult. Although traditional swarm intelligence optimization algorithms possess a unique solution for complex optimization problems [5, 26, 35], the trade-off between exploration and exploitation [1, 22] is a key and difficult issue for algorithms to adaptively adjust their performance. Therefore, a novel evolutionary optimization algorithm, the dynamic diversity evolution algorithm-2 (DDEA-2), characterized by strong adaptability, was proposed in this study. It was applied to optimize artificial neural network parameters.

Information entropy and its correlation coefficients on the basis of multiple agent evolution were introduced to make the proposed algorithm achieve more appropriate exploration and exploitation. The exploration–exploitation trade-off under the effect of entropy coefficients was adjusted according to the emergence of optima. As a result, the distribution of individuals increasingly evolved toward superior regions. Meanwhile, a shrinkage method for the search area was developed based on optimization sampling analysis to further improve the optimization efficiency. The shrinking mechanism led swarm sampling toward a better trend and facilitated the emergence of optima more efficiently during the optimization iteration process by redefining a smaller but more dominant region for the next generation.

The experimental results showed that DDEA-2 effectively solved benchmark optimization problems. The parameters of ANN were adjusted adaptively by combining the optimization algorithm with the ANN, and the recognition effect was improved gradually. The improved ANN algorithm not only obtained excellent results in small-sample recognition experiments but also demonstrated the effectiveness of DDEA-2. This study introduced DDEA-2, a novel optimization algorithm that combined dynamic population size and search area adjustment, information entropy, and a shrinking mechanism to achieve superior performance compared with traditional methods. We demonstrated its effectiveness by applying it to optimize the parameters of artificial neural networks, which led to significant improvements in recognition accuracy.

2. Related works

This study included two aspects: intelligent optimization algorithm design and artificial neural network parameter optimization. First, we proposed DDEA-2 as an optimization algorithm with better performance to solve complex optimization problems. Second, the optimization algorithm was combined with the parameter settings for ANN by constructing appropriate objective functions to obtain better recognition results.

2.1. **Optimization algorithms.** The particle swarm optimization (PSO) algorithm is a frequently used swarm intelligence optimization algorithm with fast convergence and more flexible changes [2,8,18]. The dynamic update mechanism of PSO is shown in Equation (2.1).

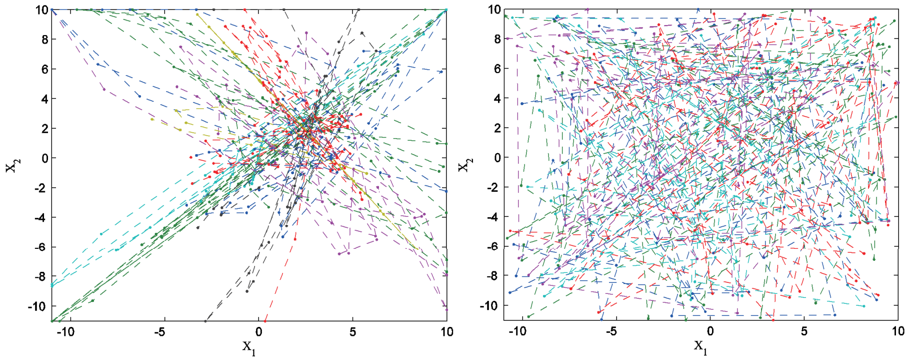
$$(2.1)\qquad \begin{cases} \overrightarrow{v}_{i+1} &= \omega\overrightarrow{v}_i + \overrightarrow{U}(0,\phi_1)\otimes(\overrightarrow{p}_i - \overrightarrow{x}_i) + \overrightarrow{U}(0,\phi_2)\otimes(\overrightarrow{p}_g - \overrightarrow{x}_i) \\ \overrightarrow{x}_{i+1} &= \overrightarrow{x}_i + \overrightarrow{v}_{i+1} \end{cases},$$

where $\overrightarrow{v}_{i+1}$ and $\overrightarrow{v}_i$ represent the steps of the current individual in the $(i{+}1)$th and ith generations [8], respectively; $\overrightarrow{x}_{i+1}$ and $\overrightarrow{x}_i$ represent the sampling positions of the current individual in the $(i{+}1)$th and ith generations, respectively [8]; ω represents the system inertia, particularly the degree to which the current individual maintains the original step size [8]; the function $\overrightarrow{U}(0,\phi)$ generates a random weight coefficient from *0* to ϕ, increasing the diversity of the swarm [8]; $\otimes$ represents the influence of the weight coefficient and is generally treated as multiplication [8]; and $\overrightarrow{p}_i$ and $\overrightarrow{p}_g$ represent the personal best positions of the current individual and the global best position of the current swarm, respectively [8]. Using Equation (2.1), individual sampling directions of particles can be updated one by one based on both personal best and global best information [8].

According to the dynamic Equation (2.1) of the PSO, the position evolution of each particle is attracted by both individual and global optima, making the population quickly converge to the current optimal region. The convergence of all individuals and mining of more precise optima in PSO results in strong exploitation . However, PSO tends to fall into local optima [8] due to overconvergence, which is a consequence of its strong exploitation. Fig. 1(A) shows that the particles represented as different colors [12] fall into a local optimum nearby point (0,0), which is the global optimum.

Genetic algorithm (GA) is a swarm intelligence optimization algorithm with a wide search range and expansive population changes [8]. Its core dynamic updating mechanism is based on the crossover and mutation of chromosomes [8,23,36]. On the one hand, the crossover of excellent individuals preserves some superior features while avoiding excessive convergence in the dominant region [8]. On the other hand, individual random mutation preserves the global exploration mechanism throughout the evolution process [8]. These factors improve the optimization reliability of the GA. However, the evolutionary direction of the individuals after their crossover and mutation is not easy to control, introducing some difficulties that can affect the convergence of the algorithm [8]. Fig. 1(B) shows that the chromosomes represented as different colors [12] are scattered everywhere, and exploration rarely converges to optima. Compared with PSO, GA has slower optimization speed but higher reliability.

DDEA-2 was suggested as a swarm intelligence optimization algorithm with better comprehensive performance in this study. Dynamic transitions between exploration and exploitation were achieved by introducing diverse agents for multimodal



(A) Unconstrained behavior of PSO.

(B) Unconstrained behavior of GA.

FIGURE 1. Trajectories of swarm evolutionary behavior. (Trajectories in (A) and (B) are formed by PSO and GA, respectively, with benchmark function Schaffer [12]).

evolution during optimization sampling. Information entropy was proposed to adaptively perform a trade-off between exploration and exploitation based on the fitness value and growth in DDEA-2. Particularly, the entropy coefficient was modified in a timely manner when calculating the size and distribution area of the new generation, effectively controlling the distribution trend and density of the population in the next generation. Taking inspiration from the fast convergence of PSO, we introduced a shrinking mechanism with subregions to improve the speed of the optima ’ s emergence.

2.2. **Parameters setting.** The parameter setting for ANN includes two aspects: parameter selection and setting method. ANN has many adjustable parameters, and generally, only the main parameters with a significant impact on recognition performance are selected as the adjustment objects during the design process. Therefore, the appropriate selection of adjustment parameters directly influences the final design effect. Manual and automatic settings are currently the main parameter-setting methods, among which automatic setting is usually combined with optimization algorithms. The automatic setting of parameters requires a higher comprehensive ability of the optimization algorithm because of the different characteristics of the optimization objective function brought about by different parameter selections.

The frequently used adjustable parameters of ANN include the number of nodes in each layer, weights between nodes, learning rate in feedback, and so forth. Too many input layer (IL) nodes reduce the generalization ability of the ANN, whereas too few IL nodes tend to lose important information. In addition, the learning rate directly influences the training effect of the ANN. Thus, the number of nodes in the input layer and the learning rate in feedback were proposed as the adjustment parameters to be optimized in this study. We also transformed the problem of selecting the number of IL nodes into the problem of selecting the number of sampling rows for image recognition.

The efficiency and adaptability for parameter setting are improved by combining the setting process with an optimization algorithm. The automatic setting method is more suitable for combining with optimization algorithms because of the necessity to repeatedly test whether the fitness functions corresponding to various parameters are optimal. The speed, accuracy, and generalization ability of optimization algorithms directly affect the quality of parameter settings. In addition, constructing the objective function can indirectly influence the effectiveness of parameter settings. Thus, DDEA-2 with better comprehensive performance was introduced to optimize the parameter setting for ANN by constructing an appropriate objective function, achieving wonderful results.

2.3. **Comparative summary.** A comparative summary of previous studies is provided in Table 1. We compared previous studies from three aspects, efficiency, accuracy, and adaptability, and then roughly divided them into the following three relative levels to assist in explanation: high (H), medium (M), and low (L). The comparison results may not be comprehensive, but they can provide an overview of the general situation from various aspects.

Table 1. Comparative summary of previous studies

Previous studies	Efficiency	Accuracy	Adaptability
PSO	H	M	M
GA	L	M	M
ANN: manual tuning	L	M	H
ANN: optimized tuning	H	H	M

3. Optimization algorithm

The design of DDEA-2 includes three main aspects: evolutionary method, information entropy, and adaptive adjustment. We defined a single sample and other information derived from it as an agent during swarm optimization. Each agent contained not only the sampling coordinates and fitness of the current individual but also the size of the fitness change, the distribution parameters of its offspring, and so forth. The optimization task was achieved through the continuous evolution of these agent populations.

3.1. **Evolutionary method.** We introduced a three-layer agent mode, including partition management (PM), basic agents (BAs), and creation agents (CAs), into the sampling control developed in DDEA-2. Reasonable trade-off between global and local exploitation is the essence of improving the accuracy and rapidity of the swarm optimization algorithm. The optimization algorithm used uniform PM and randomly distributed BAs in the initial evolutionary stage to ensure the diversity of the population and avoid falling into local optima. CAs were generated around each BA as the next generation to carry out local exploitation. Then, the remaining CAs were converted into BAs in the next iteration through an elimination mechanism. After each iteration, the scope of the agents in the next generation shrank gradually to increase the degree of exploitation. The degree of global exploration

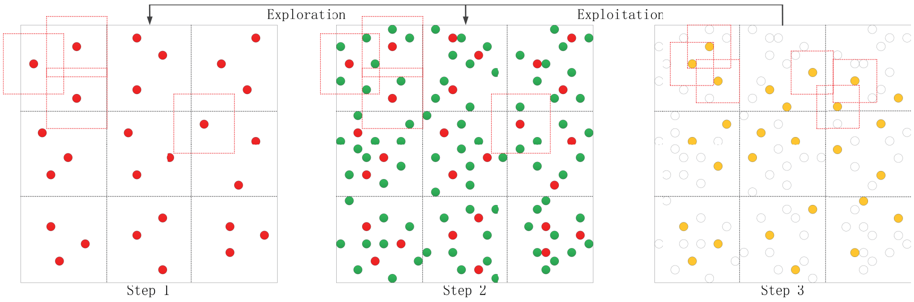


Figure 2. Evolutionary process of population distribution.

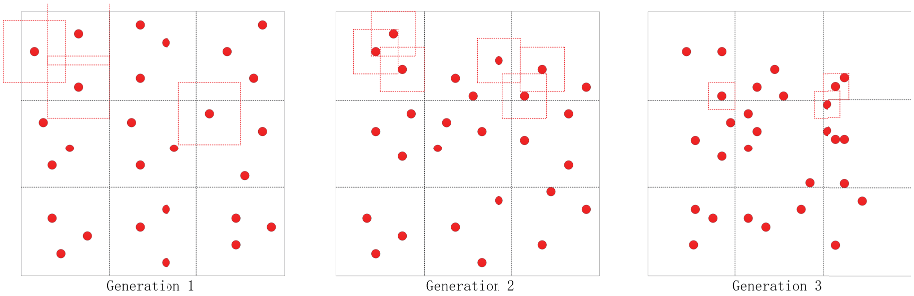


Figure 3. Convergence process of population distribution.

was controlled by adjusting the number and times of partition. The degree of local exploitation was controlled by setting the number of alternate iterations between BAs and CAs.

The evolutionary process of populations in an iteration cycle is presented in Fig. 2. First, the regions were divided by the black dotted lines, and BAs were randomly generated in each region, shown as red dots in step 1. The range and scale of agents in the next generation were calculated based on the fitness of these BAs. The red dotted boxes around some BAs shown in the figure are the ranges of agents born in the next generation. Second, CAs were randomly generated around each BA according to the range and scale of agents as shown in Fig. 2(step 2). The green individuals represent CAs generated as sub-generation agents around the red base–generation agents. The range and scale of agents that would be born in the next generation were calculated according to the fitness distribution characteristics of all the current agents, and the results were stored temporarily as the individual attributes of the current agents. Third, better individuals were selected and retained to enter the next iteration based on the survival of the fittest mechanism. The yellow individuals in Fig. 2(step 3) are the winning agents, and they enter the next cycle iteration as new BAs. The hollow dotted-line individuals are the eliminated agents, and they no longer enter the next cycle process. Then, the algorithm judged whether the region needed to be re-divided in the next iteration based on the current conditions. If necessary, it looped to step 1 to enter the next iteration. If not, it looped to step 2 to enter the next iteration. When the termination conditions were met, the algorithm exited the loop and ended the program.

Fig. 3 shows the changes in the distribution of BAs between various generations. Obviously, the scope of sub-agents continued to narrow as the iteration progressed, and the agent groups converged gradually to realize fine exploitation of the target in more optimal regions. On the contrary, region division limited the excessive convergence of some agents to the current optimum, maintaining a better individual diversity. Thus, the algorithm achieved extensive exploration in the global region and prevented the population from falling into a local optimum.

The effective acquisition of heuristic information is the key to improving the search ability of swarm intelligence optimization algorithm. The heuristic information mainly originates from the coordinates of individuals with better fitness, which is constantly revised and eventually converges to a better direction as the population evolves in iterations. In this study, we introduced an analysis of fitness improvement and considered using the coordinate region of the population with faster fitness improvement to create new heuristic factors, thereby guiding the evolution direction of the population jointly. The aforementioned two heuristic factors were generated from the mutual comparison of groups. The heuristic effects were recorded and used as different attributes of optimization individuals, updating continuously in the optimization iterations.

To sum up, the adaptive adjustment in DDEA-2 was formed by the automatic adjustment in the distribution trend of the optimization group guided by multiple heuristic factors. The heuristic intensity was quantified using the information entropy coefficients, including fitness entropy coefficient (f_{ec}) and fitness growth entropy coefficient (g_{ec}). Then, the derivative range and scale of the next generations were adjusted using f_{ec} and g_{ec}, respectively, playing an important role in characterizing the trend of population distribution. Thus, the adaptive regulator was established based on information entropy.

3.2. **Information entropy.** Information entropy is defined here as a measure of fitness distribution, that is, the degree to which a better target might appear. The physical meaning of the analogy “entropy” is a measure of the degree of confusion in a system. Analogously, this study introduced the concept of information entropy to reversely characterize the degree of clarity of the optimal solution. Considering the involvement of several heuristic factors in the emergence of a better target, the information entropy is also composed of a variety of coefficients. This study suggested the fitness entropy coefficient (f_{ec}) and fitness growth entropy coefficient (g_{ec}) to describe the information entropy in current optimization sampling.

The fitness entropy coefficient is defined as Equation (3.1),

$$f_{ec} = e^{\frac{2(f-f_{mid})}{f_{max}-f_{min}}} \tag{3.1}$$

where f_{max}, f_{min}, and f_{mid} represent the current maximum fitness, minimum fitness, and median fitness, respectively. The fitness entropy coefficient indicates the optimal degree of the current agent in its group. The higher the f_{ec} the more dominant the region where the current agent is located. The optimization sampling range is narrowed in dominant regions to enable targeted small-scale exploitation, reducing invalid sampling in inferior regions and improving the utilization of agent sampling resources.

The growth entropy coefficient is defined as Equation (3.2),

$$g_{ec} = e^{\frac{2(g-g_{mid})}{g_{max}}} \quad if\ g>0 \tag{3.2}$$

where g_{max} and g_{mid} represent the current maximum fitness growth and growth mid-value, respectively. The growth entropy coefficient indicates the complexity of fitness change in the region where the current agent is located. The higher the g_{ec} is, the greater the increment speed and increment space of the region where the current agent is located. The number of sub-agents in the complex regions needs to be increased to improve the precision of the search, preventing the optimization process from missing potential advantage regions and improving the reliability of the distribution of agent sampling resources. The generation scale on which sub-agents are generated is set to a minimum if $g \leq 0$.

3.3. **Adaptive adjustment.** The derivative range and scale of agents in the next generation were adjusted according to Equations (3.3) and (3.4),

$$r_{i+1} = \beta \times \left[(1-\alpha) + \frac{\alpha}{f_{ec}}\right] \times r_i \tag{3.3}$$

$$s_{i+1} = [(1-\alpha) + \alpha \times g_{ec}] \times s_i \tag{3.4}$$

where r_i and r_{i+1} represent the range of agents in current and next generations, respectively; β is a specified shrinkage coefficient set to 2/3 in this study; α is a learning rate affecting the adjustment speed of dynamic parameters; and s_i and s_{i+1} represent the scale of agents in current and next generations, respectively.

The range of sub-agents produced in the next generation by an agent decreases with the improvement in the fitness of the current agent. The scale of sub-agents produced in the next generation by an agent mainly increases or decreases according to the fitness growth of the current agent. The optimization sampling in the dominant region should be more emphasized, although both the rate of increase and decrease in fitness can indicate the complexity of fitness change. Therefore, the scale of sub-agents around an agent with a fitness growth of less than zero was set to a minimum in this study. The range and scale of sub-agents worked together to dynamically adjust the sampling density.

A priority feasible region was suggested to deal with the optimization problems, characterized by a fitness distribution similar to that of convex optimization. The fitness distribution is generally centered on the optimum and shows a trend of deterioration to the surroundings in most optimization problems. The difference is that the fitness fluctuates to varying degrees during the intermediate transition. Therefore, the orientation of the optimal region can be generally analyzed after a period of optimization sampling. If the optimization range is appropriately narrowed to an optimal region, the emergence speed of the best solution can be effectively improved. Fig. 4 shows the emergence of a priority feasible region by adjusting the range and scale of offspring.

When the algorithm progressed to re-partition regions, a cutoff fitness value was inferred from the current agent population and the agents with fitness greater than this value were screened. Then, a subregion was redefined as a priority feasible

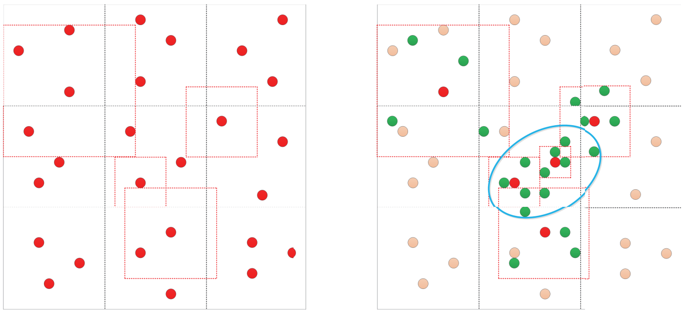


FIGURE 4. Priority feasible region caused by derivative range and scale.

region based on the maximum and minimum coordinates of the retained advantageous individuals. As the iteration progressed in DDEA-2, the priority feasible region shrunk gradually to the optimal region during optimization.

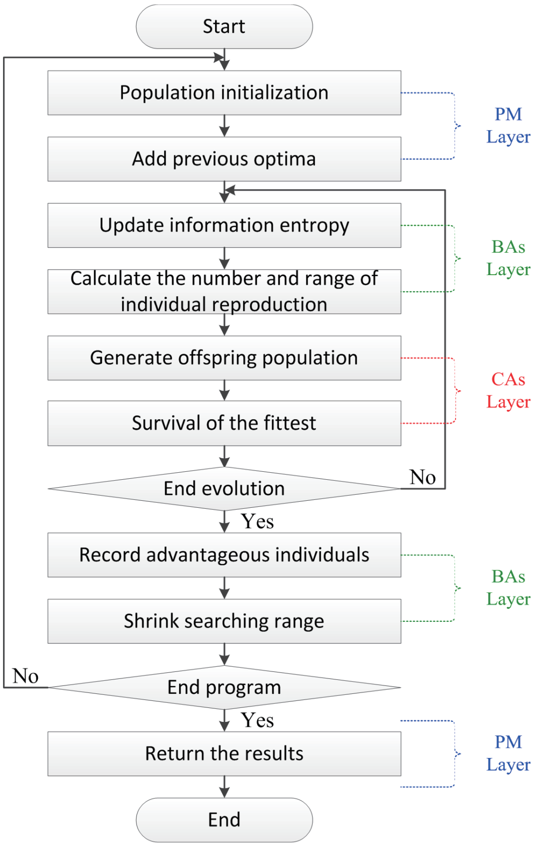

Start
Population initialization
Add previous optima
PM Layer
Update information entropy
Calculate the number and range of individual reproduction
BAs Layer
Generate offspring population
Survival of the fittest
CAs Layer
End evolution
No
Yes
Record advantageous individuals
Shrink searching range
BAs Layer
End program
No
Yes
Return the results
PM Layer
End

FIGURE 5. Flowchart of DDEA-2.

The flowchart of DDEA-2 is shown in Fig. 5. First, the algorithm sequentially enters the PM, BA, and CA layers from top to bottom. Second, the conversion from CAs to BAs forms an inner loop, and the conversion from the BAs to the PM forms an outer loop. Finally, the algorithm returns and ends the process after passing through the PM layer. Overall, the algorithm divides the process into three levels, from the outside to the inside, by integrating a three-layer agent model. The outer layer focuses on global exploration, whereas the inner layer focuses on local exploitation. Combined with information entropy, this approach led to a dynamic adjustment and alternation between exploration and exploitation. The shrinking mechanism was arranged in the outer loop and guided by the advantageous BAs in the current iteration, aiming to converge the next round of population to a dominant region.

4. ANN optimization

4.1. **Structural parameters.** The artificial neural network and training model to be optimized in this study are shown in Fig. 6. A given image is sampled intermittently and expanded into a one-dimensional vector as the IL nodes of the neural network, as shown in the left part of Fig. 6. Each IL node outputs information to the hidden layer (HL) nodes after weighting. The HL nodes are weighted and output to output layer (OL) nodes. The output and true values are comprehensively compared and processed, and the error is fed back to correct the weights between each layer. The degree of feedback correction is controlled by a learning rate (*eta*), making the current output converge to the true value.

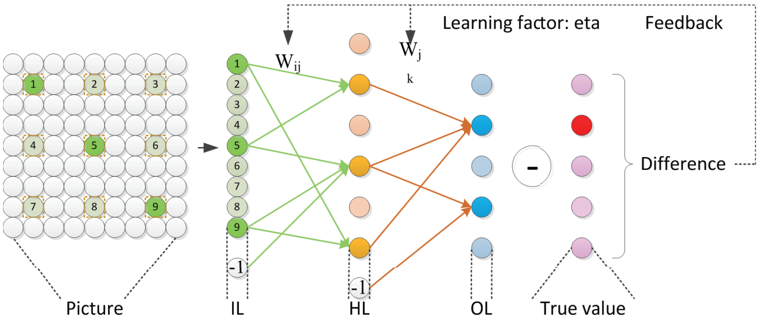


Figure 6. Proposed ANN structure and training model.

In this model, the sparsity of image sampling in the input phase can be controlled by setting the number of rows and columns (*Ir* and *Ic*) for sampling. Too many sampling rows and columns reduce the generalization ability of the model, while too few sampling rows and columns lead to the loss of important information. This indicates that inappropriate *Ir* and *Ic* reduce the accuracy of the model in identifying new samples. In addition, the learning rate (*eta*) also directly affects the training effect of the entire model during the training of the neural network. Therefore, this study took *Ir*, *Ic*, and *eta* as the optimization parameters of the neural network. The number of sampling rows and columns was set to the same value in this study to reduce the amount of optimization computation. Thus, the original optimization problem was reduced to a two-dimensional optimization problem by determining *Ir* and *eta* as optimization parameters.

4.2. **Fitness function.** For combining the optimization problem of ANN parameters with the optimization algorithm, the optimization parameters should be used as independent variables to design a fitness function that reflects the training quality of the neural network. Generally, samples are divided into training and testing sample sets to complete the training and testing of the ANN model, respectively. During ANN reinforcement learning, different selections of sampling row and column numbers and learning rates produce different training effects. The training effect is reflected in the recognition rate of not only the current training sample set but also the test samples, which is called the generalization ability of a pattern recognition algorithm for the samples that have not participated in the training. Therefore, the fitness function proposed in this study was mainly designed based on test samples, as shown in Equation (4.1).

$$\begin{aligned} F\left(Ir, eta\right) &= E\left\{P\left[T\left(Ir, eta, Sxy\right), TSx\right], TSy\right\} \\ &= E\left\{P\left[\omega_{ij}, \omega_{jk}, TSx\right], TSy\right\} \\ &= E\left\{Py, TSy\right\} \end{aligned} \tag{4.1}$$

The construction process of the fitness function is shown in Equation (4.1). First, function T was used in the training process to obtain the network weights, ω_{ij} and ω_{jk}, with "*Sxy*" as the training samples. *Ir* and *eta* were used as training parameters of ANN and decision variables of DDEA-2 in this study. Second, function P was used by the trained ANN in prediction to obtain the recognition result of " *Py* " based on the test samples " *TSx* ". Third, the difference between the recognition result "*Py*" and the real test sample result "*TSy*" was used to calculate the current fitness. In this study, a column vector result " py_n " was obtained for recognizing the nth test sample, and the modulus of the difference between the column vector and the actual vectors " tsy_n " was used as the recognition error for the current sample. The fitness of decision variables was defined based on the recognition errors for test samples as shown in Equation (4.2).

$$f\left(Ir, eta\right) = \sqrt{\frac{\sum_{n=1}^{N} \left|py_n - tsy_n\right|^2}{N}} \tag{4.2}$$

On the contrary, not all test samples were used for the design of the fitness function in this study. The parameters of the neural network were adaptively adjusted through the optimization algorithm; that is, the test samples were used to indirectly guide the design of the neural network model. In practical applications, an ANN is generally used to recognize new targets whose features may not be identical to the samples in previous training and testing. Only the model with good generalization ability can extend the good recognition effect to the unknown target recognition. Therefore, some samples should be retained for verifying the actual generalization ability of a new ANN. These samples do not participate in either neural network training or optimization of neural network parameters.

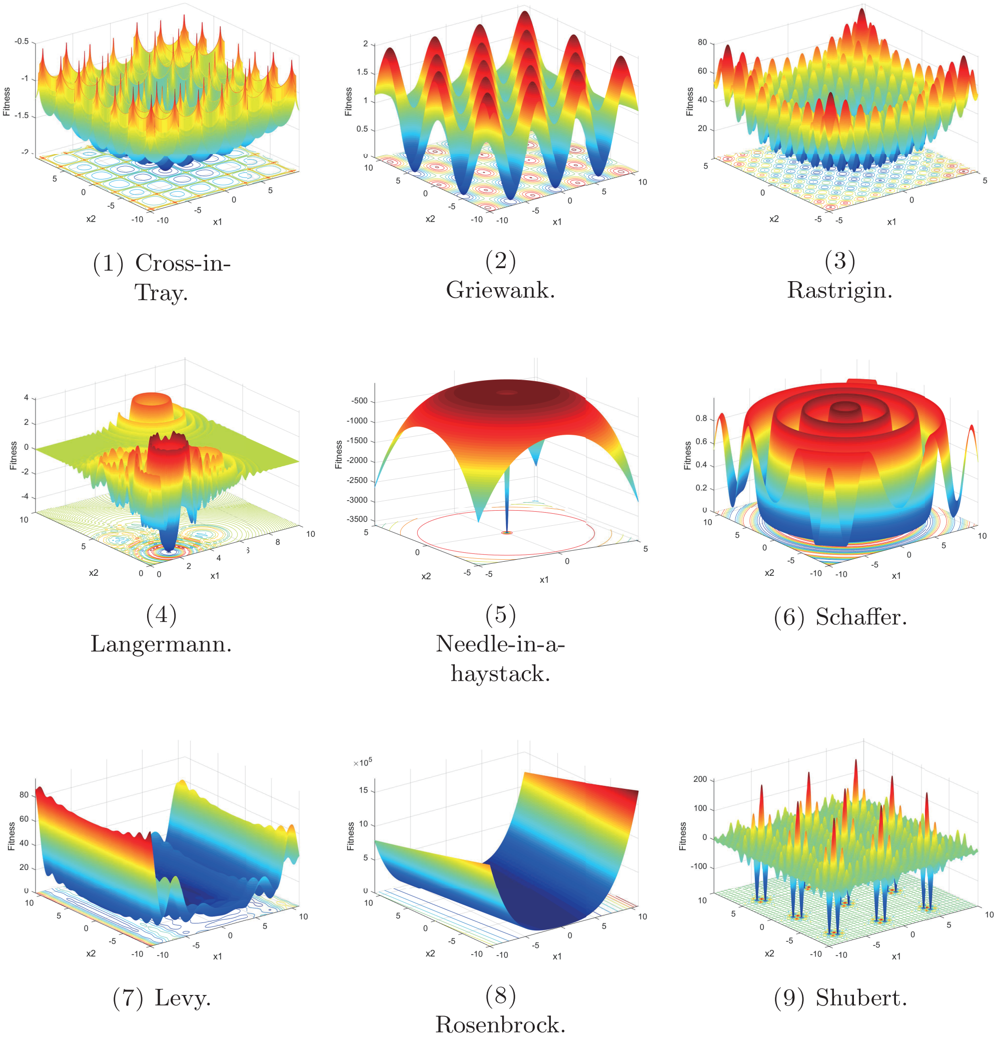
4.3. **Discrete seeking.** The decision variables processed using the DDEA-2 in this study used values in continuous space; however, the row parameter of the neural network required integer data. A method of discrete seeking was introduced into the fitness calculation of the original optimization algorithm in this study to extend the application scope of DDEA-2 from continuous optimization to discrete optimization problems. When the new location coordinates of the optimization agents were determined, the discrete parameters were rounded, and the agents with the same location appeared. Before the fitness calculation, whether DDEA-2 had ever sought the position of each agent was judged. If information about the same position was obtained in seeking history, the fitness in the history was directly assigned to the current agent. If not, the fitness was calculated according to Equation (3.2), and the position and the fitness were added to the seeking history. Thus, computing resources were saved, and computing efficiency was improved to a certain extent.

5. Experiments and analysis

Swarm intelligence optimization algorithms are generally tested using benchmark functions [8] to demonstrate the advantage of the new algorithm and thus objectively analyze the effectiveness of the algorithm [8]. We validated the superior performance [8] of DDEA-2 using nine well-known benchmark functions. We compared it with five state-of-the-art algorithms: PSO, GA, variable particle swarm optimization (VPSO), flexible grid optimization (FG), and rain forest algorithm (RFA). Statistical data related to the optimization results were analyzed to demonstrate the reliability of the proposed approach. Then, we verified the actual application effectiveness of DDEA-2 in terms of solving regression optimization problems for ANN in the learning and recognition of digital samples.

5.1. **Experimental design.**

We set up nine benchmark optimization problems and one ANN parameter optimization problem in the following experiments. The first nine experiments were divided into three groups, each with a different distribution in fitness, to fully verify the comprehensive performance of DDEA-2 by comparing it with other state-of-the-art algorithms. The fitness distribution of Cross-in-Tray, Griewank, and Rastrigin is shown in Fig. 7(1-3) for the first three test functions, respectively. Their global optimal coordinate was (± 1.3491, ± 1.3491), (0,0), and (0,0), respectively. The corresponding optimal fitness was -2.06261, 0, and 0. These functions presented many local minima, resulting in many challenges for optimization algorithms to avoid falling into local optima. The fitness distribution of Langermann, Needle-in-Haystack, and Schaffer is shown in Fig. 7(4-6) for the middle three test functions, respectively. Their global optimal coordinate were (2.00299219, 1.006096), (0,0), and (0,0), respectively. The corresponding optimal fitness was -5.1621259, -3600, and 0. The global optimization of these functions was extremely concealed, leading to strong challenges for optimization algorithms to avoid premature convergence and conduct global exploration. The fitness distribution of Levy, Rosenbrock, and Shubert is shown in Fig. 7(7-9) for the middle three test functions, respectively. Their global optimal coordinate was (1, 1), (1,1), and multiple solutions, respectively. The corresponding optimal fitness was 0, 0, and -186.7309. These functions had a relatively flat distribution around their global or multiple optima over a large range, challenging fast convergence and local exploitation.



(1) Cross-in-Tray.

(2) Griewank.

(3) Rastrigin.

(4) Langermann.

(5) Needle-in-a-haystack.

(6) Schaffer.

(7) Levy.

(8) Rosenbrock.

(9) Shubert.

Figure 7. Fitness distribution of three benchmark functions.

The distribution characteristics of these benchmark functions in each group were similar but not identical. Therefore, while reflecting the performance of the optimization algorithms in terms of speed and accuracy, these tests also reflected the adaptive ability and universality of the optimization algorithms in practical applications to a certain extent.

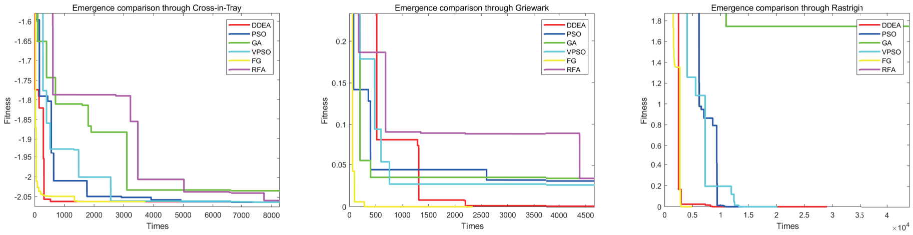
The latter experiment was to verify the effectiveness of DDEA-2 in improving the recognition performance of ANN in application. The number of sample rows and the learning rate for the ANN were selected as the parameters to be optimized. Each agent ’ s coordinate generated during the iterative optimization process of the DDEA-2 represented a combination of sample rows and learning rate for the ANN. These two parameters were used in the training on ANN, and the comprehensive recognition error in testing was calculated as the fitness of the current agent.

In these tests, we compared and analyzed DDEA-2 with five state-of-the-art algorithms: PSO, GA, VPSO, FG, and RFA. The dimension of decision variables was selected as 2 in each objective function; that is, two independent variables ($x1$, $x2$) existed in each optimization problem. For DDEA-2, the initial population size of the algorithm of DDEA-2 was set to 16 (4 × 4), the learning factor was set to 0.5, the number of external cycles was set to 6, the number of internal cycles was set to 4, the truncation rate was set to 0.3, the lower limit of population size was set to 4 (2 × 2), and the upper limit was set to 25 (5 × 5). For PSO, the initial population size was set to 100, the individual optimal weight coefficient was set to 1.49445, the global optimal weight coefficient was set to 1.49445, and the maximum number of iterations was set to 1000. For GA, the initial population size of the GA algorithm was set to 100, the chromosome coding length was set to 15, the mutation probability was set to 0.05, and the maximum number of iterations was set to 1000. The initial VPSO population was set to 40 random individuals, the interval for population variation was set to 51, and the settings of the remaining parameters were the same as for PSO. The initial FG population was set to 49 (7 × 7) even individuals, the learning factor was set to 0.38, the number of external cycles was set to 50, the number of internal cycles was set to 20, the lower limit of the population size was set to 25 (5 × 5), and the upper limit was set to 81 (9 × 9). The initial RFA population was set to 36 (6 × 6) even individuals, the learning factors were set to 0.5, the number of external cycles was set to 4, the number of internal cycles was set to 6, and the lower limit of population size was set to 9 (3 × 3), with no maximum population limit.

5.2. **Results and analysis.**

5.2.1. *Optimization effect.* The results of the comparison between DDEA-2, PSO, GA, VPSO, FG, and RFA using the nine benchmark functions are shown in Fig. 8–10. The red, blue, green, cyan, yellow, and magenta solid lines in these figures represent the optimization curves generated by DDEA-2, PSO, GA, VPSO, FG, and RFA, respectively. Each optimization algorithm displayed different performances, which were attributed to the various distribution characteristics of the objective functions. DDEA-2 did not always show the fastest and most accurate emergences in all tests, but it performed relatively well in most of them. Both DDEA-2 and PSO explored the optimal region and exploited the optimal value more quickly and stably in most cases; DDEA-2 performed better in terms of the rapidity and accuracy of optimization. These multiple tests confirmed that DDEA-2 had better adaptive ability, laying a strong foundation for the application of DDEA-2 in ANN optimization.

DDEA-2 enhanced its exploration due to the low information entropy in the early stage of optimization for Cross-in-Tray and Rastrigin, thus capturing the optimal region earlier in the first 500 and 3000 samples; also, the optimal result emerged more quickly. When the information entropy was high in the later stage, DDEA-2 enhanced its exploitation, thus accelerating the emergence of optima in the middle and later stages and approaching the global optimum before 2500 samples for Griewank. The shrinkage mechanisms were also added to improve the optimization efficiency and accuracy.

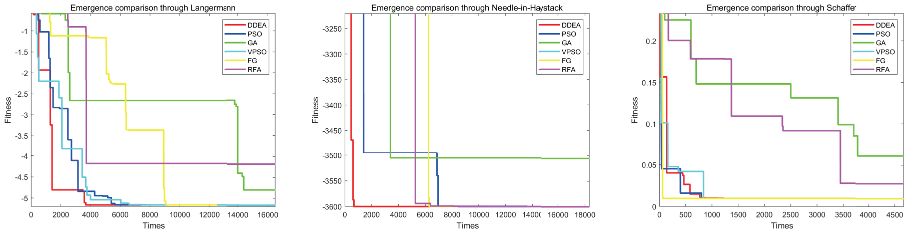


(1) Emergences at Cross-in-Tray.

(2) Emergences at Griewank.

(3) Emergences at Rastrigin.

FIGURE 8. Optimization emergence compared with that in group 1.



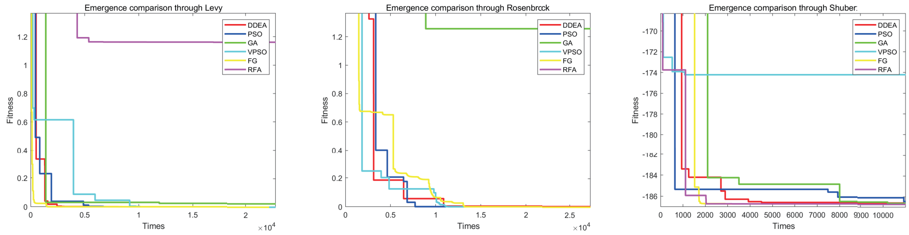
(1) Emergences at Langermann.

(2) Emergences at Needle-in-Haystack.

(3) Emergences at Schaffer.

FIGURE 9. Optimization emergence compared with that in group 2.

The low information entropy in the early stage of optimization for Langermann and Needle-in-Haystack also made DDEA-2 enhance its exploration, thus capturing the optimal region earlier in the first 2000 samples; also, the optimal result emerged more quickly. This accurate exploration for the global optimal region directly accelerated the exploitation effect, approaching the global optimum earlier before 1000 samples for Schaffer.



(1) Emergences at Levy.

(2) Emergences at Rosenbrock.

(3) Emergences at Shubert.

FIGURE 10. Optimization emergence compared with that in group 3.

The low information entropy in the early stage of optimization for Levy and Rosenbrock DDEA-2 enhance its exploration, thus capturing the optimal region comparable to PSO, VPSO, and FG; also, the optimal result emerged quickly. Although DDEA-2 did not perform extremely well for Shubert, its adaptive adjustments in the exploration–exploitation network made its performance relatively stable. The high information entropy in the middle and later stages made DDEA-2 enhance its exploitation and approach the global optimum before 5000 samples for Shubert.

Fig.8–10 show all random test results. This study involved 100 tests for each algorithm and benchmark function above, totaling 100 × 6 × 9 experiments, to further demonstrate the reliability of DDEA-2. The mean and standard deviation (SD) of the optimal results obtained using each optimization algorithm are shown in Table 2–4.

Table 2. Statistical data for reliability experiments in Group 1

Algorithm	Cross-in-Tray		Griewank		Rastrigin	
	Mean	SD	Mean	SD	Mean	SD
DDEA-2	-2.0626	0.0000	0.0003	0.0013	0.0585	0.2204
PSO	-2.0626	0.0000	0.0014	0.0030	0.0497	0.2179
GA	-2.0626	0.0000	0.0028	0.0026	0.7990	0.5012
VPSO	-2.0626	0.0000	0.0012	0.0028	0.0199	0.1400
FG	-2.0626	0.0000	0.0031	0.0040	1.0646	0.6205
RFA	-2.0618	0.0004	0.0761	0.0250	2.0619	0.0808

In the first set of experiments, DDEA-2 achieved the minimum mean and SD in tests for Cross-in-Tray and Griewank, as shown in Table 2. The mean minimum achieved by DDEA-2 for Rastrigin was slightly larger than that achieved by PSO and VPSO. However, the SD achieved by DDEA-2 for Rastrigin was below 0.3, indicating that the stability and reliability of DDEA-2 were comparable to that of some state-of-the-art algorithms.

In the second set of experiments, DDEA-2 achieved the minimum mean in tests for Needle-in-Haystack and Schaffer, as shown in Table 3. The SD achieved by DDEA-2 for Needle-in-Haystack was also minimal. The mean and SD achieved by DDEA-2 for Langermann were larger, indicating that Langermann presented strong challenges for DDEA-2.

In the third set of experiments, the means and standard deviations achieved by DDEA-2 were all close to the minimums, as shown in Table 4. The algorithms that performed the best were different from each other in optimizing these three benchmark functions, indicating that stability and reliability were challenges for each algorithm.

5.2.2. *Recognition effect.* The regression and recognition of a recognition algorithm can be adjusted adaptively based on the specific situation of various problems by

TABLE 3. Statistical data for reliability experiments in Group 2

Algorithm	Langermann		Needle-in-Haystack		Schaffer	
	Mean	SD	Mean	SD	Mean	SD
DDEA-2	-4.0969	1.5184	-3600.0	0.0319	0.0013	0.0087
PSO	-5.1405	0.2162	-3102.5	446.7192	0.0042	0.0048
GA	-5.0797	0.0715	-3595.6	4.5957	0.0074	0.0032
VPSO	-5.1621	0.0000	-3353.8	387.2647	0.0022	0.0041
FG	-4.7834	0.8628	-3592.6	73.8779	0.0092	0.0021
RFA	-1.8010	0.6461	-3483.1	19.5322	0.0024	0.0014

TABLE 4. Statistical data for reliability experiments in Group 3

Algorithm	Levy		Rosenbrock		Shubert	
	Mean	SD	Mean	SD	Mean	SD
DDEA-2	0.0000	0.0000	0.0005	0.0014	-186.6953	0.0405
PSO	0	0	0	0	-183.0419	5.6780
GA	0.0040	0.0047	0.1954	0.2141	-186.7023	0.0312
VPSO	0	0	0.1864	1.4944	-182.4767	5.9570
FG	0.0000	0.0000	0.0517	0.0957	-186.7309	0.0000
RFA	0.0059	0.0006	2.6842	0.5504	-185.6250	0.1709

combining its parameters with an optimization algorithm. Combining DDEA-2 with ANN improved the structural parameters of ANN and verified the practical application ability of DDEA-2 in this study. The number of sample rows and the learning rate for the ANN were selected as the parameters to be optimized. The coordinates of each agent generated during the iterative optimization process of the DDEA-2 represented a combination of sample rows and learning rate for the ANN. These two parameters were used in the training on ANN, and the comprehensive recognition error in testing was calculated as the fitness of the current agent. After optimization iteration, better parameters gradually emerged from the evolution of the population, and the ANN achieved satisfactory regression and recognition effects using these optimal parameters.

Three different feasible ranges were used to conduct the tests. The result curves are shown in Fig. 11, 12, and 13, respectively. The red, blue, green, cyan, orange, and magenta lines in these figures represent the optimization curves generated by DDEA-2, PSO, GA, VPSO, FG, and RFA, respectively. The " Times " in the horizontal axis refers to the sum of the samples distributed until the current generation, that is, the total number of fitness calculations in the decision space through the previous optimization process. The recognition error of the ANN optimized by DDEA-2 in each test dropped below 0.2. The optimal parameters obtained by DDEA-2 in the tests were relatively consistent because of the adaptive adjustment. DDEA-2 showed a more sustained and robust emergence process in each test.

PSO displayed a fast convergence speed in these tests, but the emergence of optima in the later stage was affected by the dropping into local optima. GA showed relatively stable optimization characteristics in these tests, which benefited from its higher exploration ability. However, the emergence speed of optima in the later stage was affected by its lower exploitation ability. Overall, DDEA-2 performed better on the trade-off between exploration and exploitation and showed flexible adaptation under different optimization conditions.

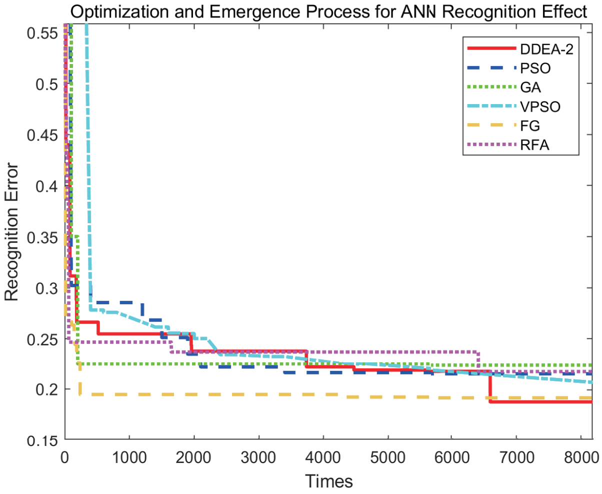


FIGURE 11. Optimization and emergence process in Test 1.

In Test 1, the feasible range of the decision variable *Ir* was set between 3 and 40, and the feasible range of the decision variable *eta* was set between 0.1 and 10. Most optimization algorithms quickly reduced the error to below 0.3; especially FG and GA showed faster emergence in the beginning. GA used its stronger exploration capabilities to obtain a smaller error below 0.25 before 300 samples, which was similar to the effect of FG with stronger exploitation capabilities. This result indicated that the dispersed search brought by exploration could sometimes assist in improving the convergence progress of exploitation toward the global optimum. Hence, DDEA-2 exceeded FG before the 7000th sampling, thus achieving a smaller error.

In Test 2, the feasible range of the decision variable *Ir* was set between 3 and 15, and the feasible range of the decision variable *eta* was set between 0.1 and 10. All six optimization algorithms quickly reduced the error to about 0.25 in the beginning in the presence of a reduced search range. When most algorithms fell into local optima, DDEA-2 was the first to jump out of the local optimum before 7000 samples to achieve a smaller error.

In Test 3, the feasible range of the decision variable *Ir* was set between 4 and 10, and the feasible range of the decision variable *eta* was set between 0.01 and

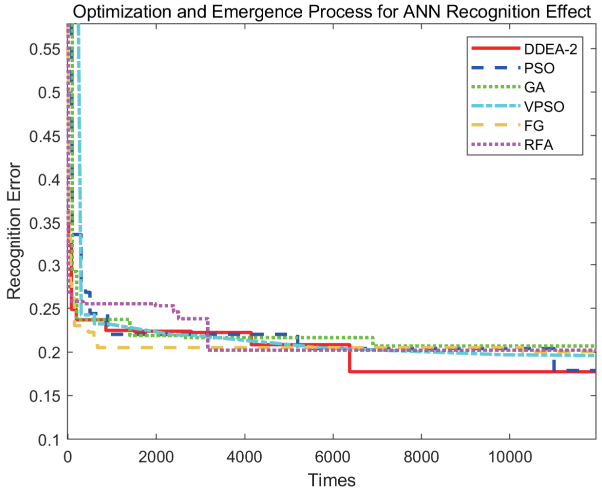


Figure 12. Optimization and emergence process in Test 2.

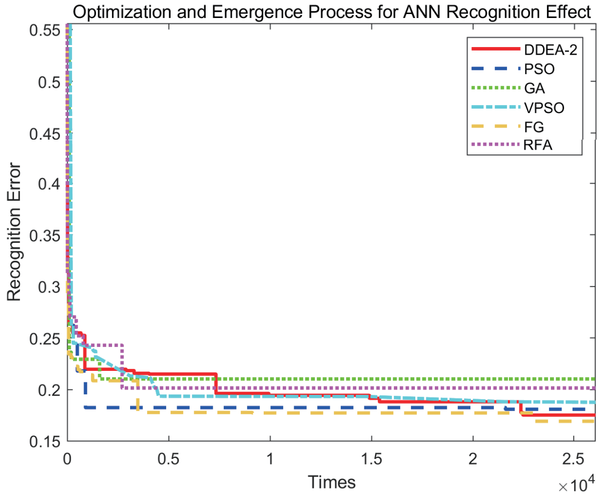


Figure 13. Optimization and emergence process in Test 3.

7. All six optimization algorithms quickly reduced the error to below 0.25 in the beginning, with a further narrowing of the search range. In a smaller search range, PSO and FG with stronger exploitation showed the advantage of rapid convergence, that is, faster emergence and better optimizing results. DDEA-2 also achieved an optimization result comparable to that of FG and PSO in the later stage.

The emergence processes conducted by DDEA-2 were more sustained and robust in the aforementioned three tests. First, it was because DDEA-2 enhanced its exploration when the information entropy was low in the early stage, making its emergence processes comparable to those of PSO and FG, whose optimization was faster. Second, information entropy continued to increase in the middle and later stages. Therefore, the DDEA dynamically adjusted exploration–exploitation network not only avoids falling into local optima but also makes the emergence of optimal values more efficient. Overall, DDEA-2 performed better on the trade-off between exploration and exploitation and showed flexible adaptation under different optimization conditions.

Table 5 shows the results and statistics for ANN optimization by DDEA-2 with different ranges of decision variables. The results showed that the smaller the feasible range we set, the smaller the recognition error achieved for ANN . This can be attributed to the generation of more exploitation samples in the smaller feasible range for the emergence of optima. The initial population was generated randomly in each test, but the optimal *Ir* and eta obtained in the end were similar. Among the three tests, all the optimal results of *Ir* were between 6 and 7, and the better results of eta were between 3 and 5. Similar results indicated that DDEA-2 had strong adaptability to various feasible regions. The three smaller SDs of errors also indicated the high stability and reliability of DDEA-2 in ANN optimization. The difference in minimum errors indicated that the proposed approach was susceptible to the range setting, requiring further improvement in future research.

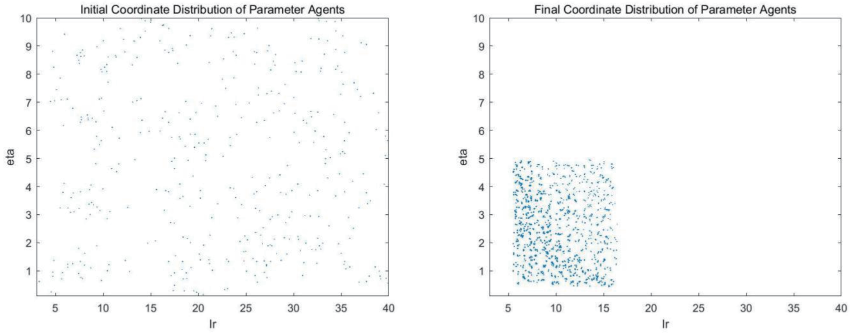
Table 5. Results and statistics for ANN optimization

Test	Ir range	eta range	Min error	Mean error	SD of errors	Ir	eta
1	3–40	0.1–10	0.1802	0.1922	0.0092	6–7	3.2518–5.2328
2	3–15	0.1–10	0.1712	0.1879	0.0114	6–7	2.9113–5.1107
3	4–10	0.01–7	0.1699	0.1854	0.0111	6–7	3.3787–5.3391

Fig. 14–16 show the state change of the population distribution converging to better regions before and after the optimization in the three tests. The results indicated that the population distribution shrank to a similar area in these tests, with more individuals approximately near point (7, 2) where the fitness was higher. Multiple sets of experiments comprehensively showed that DDEA-2 had an excellent adaptive adjustment for complex optimization problems of ANN.

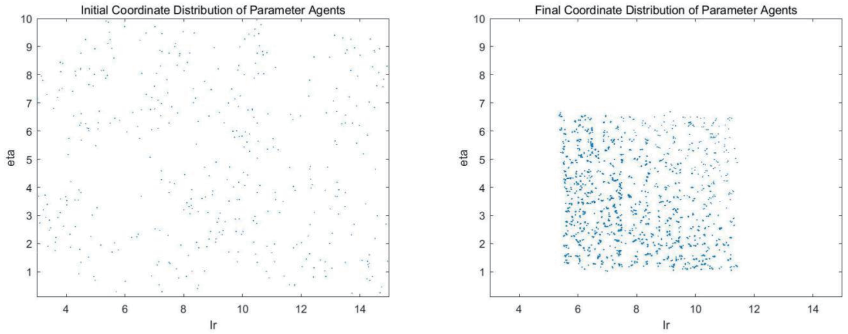
The initial scattered sampling distribution resulted from enhanced exploration by DDEA-2 because of lower information entropy. When the optimal orientation was unclear in the early stages, DDEA enhanced the dispersion of sampling points to more fully explore the optimal region globally. In the later stage, the information entropy continued to increase with an increase in sampling and the gradual appearance of the optimal orientation. The DDEA sampling population dynamically adjusted to form a trend of convergence toward the global optimum, aiming to accelerate the exploitation.

5.2.3. *Generalization capability.* The generalization capability discussed in this study included the ability to expand the application for both the optimization algorithm and modeling recognition algorithm. For DDEA-2, the optimization results were obtained more accurately by adaptive adjustment of population distribution, irrespective of whether the optimizations were different types of a nominal problem or



(1) Initial distribution. (2) Final distribution.

Figure 14. Evolutionary comparison of agent distribution in Test 1.



(1) Initial distribution. (2) Final distribution.

Figure 15. Evolutionary comparison of agent distribution in Test 2.

Figure 16. Evolutionary comparison of agent distribution in Test 3.

a specific application on neural network parameter tuning. For ANN, the recognition effect for learning samples was extended successfully to that for new samples using the optimization algorithm while adjusting the parameters in the recognition models.

6. Conclusions

This study introduced DDEA-2, a novel optimization algorithm that combined dynamic population size and search area adjustment, information entropy, and a shrinking mechanism to achieve superior performance compared with traditional methods. We demonstrated its effectiveness through its application to optimize the parameters of ANN, leading to significant improvements in recognition accuracy.

The statistical analysis indicated that the performance of DDEA-2 was relatively stable and reliable.

DDEA-2, featuring a dynamic adjustment mechanism of population size and search area, was introduced in this study. This optimization algorithm was also combined with ANN to adjust parameters for achieving excellent study and recognition effects.

Information entropy was proposed and adaptive trade-off was performed between exploration and exploitation according to the fitness value and growth. The entropy coefficient was determined to calculate the size and distribution area of the new generation, and the distribution trend and density of the next-generation population were effectively controlled. During this process, the sampled information was fully used, and the subsequent population evolution mode was effectively inspired and adjusted based on the actual situation of the fitness distribution. Thus, the adaptive adjustments of exploration and exploitation were realized with the information entropy. The population distribution density after the search iterations exhibited obvious optimization characteristics, as demonstrated in the experimental results. In addition, a shrinking mechanism for feasible ranges was suggested to improve the speed and accuracy of the emergence of optimal parameters. Remarkable rapidity and accuracy were obtained by DDAE-2 in the optimization results for three different nominal problems.

The structural parameters of an ANN were adjusted efficiently using the DDEA-2 in this study. A specialized objective function based on the recognition error for new samples was designed to help the ANN achieve better generalization capability in the case of small-sample learning. DDEA-2 was combined with ANN through the specialized objective function, and the tuning direction was more focused on the generalization recognition for new samples. Experiments confirmed that the combination was effective and achieved remarkable results in the case of small-sample learning.

We aim to further discuss the scalability of the proposed approach in the future through the following aspects: (1) testing the universality of DDEA-2 using more complex standard optimization problems; (2) selecting or adding other adjustable parameters of ANN as optimization decision variables to improve recognition performance; and (3) combining DDEA-2 with other modeling methods to address potential limitations of the approach and expand its variability to various regression applications.

References

[1] M. M. Al-Rifaie, *Exploration and exploitation zones in a minimalist swarm optimiser*, Entropy **23** (2021): 977.

[2] Ankita and S. K. Sahana, *Ba-PSO: A balanced PSO to solve multi-objective grid scheduling problem*, Appl. Intell. **52** (2022), 4015–4027.

[3] N. Chang, P. Cheng, C. Feng, L. Wang and Y. Li, *Research progress of off-road path optimization algorithm for equipment*, Optics and Precision Engineering **31** (2023), 776–792.

[4] G. Chen, T. Zhang, T. Fu, L. Wang and X. Song, *Multi-objective optimal mining trajectory planning for intelligent electric shovel based on NSGA-II*, Modern Manufacturing Engineering **2** (2024), 142–149.

[5] Z. Chen, J. Cao, F. Zhao and J. Zhang, *A grouping cooperative differential evolution algorithm for solving partially separable complex optimization problems*, Cognitive Computation **15** (2023), 956–975.

[6] S. Ding, Y. An, X. Zhang, F. Wu and Y. Xue, *Wavelet twin support vector machines based on glowworm swarm optimization*, Neurocomputing **225** (2017), 157–163.

[7] J. Gao and W. Liao, *The impact mechanism of industrial intelligence upgrading on consumer upgrading and retail enterprise efficiency*, Journal of Commercial Economics **11** (2022), 25–28.

[8] W. Gao, Q. Gao, L. Sun and Y. Chen, *Design of a novel multimodal optimization algorithm and its application in logistics optimization*, Electronic Research Archive **32** (2024), 1946–1972.

[9] W. Gao and C. Shao, *Iterative dynamic diversity evolutionary algorithm for constrained optimization*, Acta Automatica Sinica **40** (2014), 2469–2479.

[10] W. Gao, C. Shao and Y. An, *Bidirectional dynamic diversity evolutionary algorithm for constrained optimization*, Neurocomputing **2013** (2013): 762372.

[11] W. Gao, C. Shao and Q. Gao, *An optimization algorithm with novel flexible grid: applications to parameter decision in LS-SVM*, Journal of Computing Science and Engineering **9** (2015), 39–50.

[12] W. Gao, C. Shao and Q. Gao, *Pseudo-collision in swarm optimization algorithm and solution: rain forest algorithm*, Acta Phys. Sin. **62** (2013): 190202.

[13] Y. Han, S. Xiang, T. Zhang, Y. Zhang, X. Guo and Y. Shi, *Conversion of a single-layer ANN to photonic SNN for pattern recognition*, Science China(Information Sciences) **67** (2024), 261–270.

[14] Y. Han, X. Wang, Z. Geng, Q. Zhu, S. Bi and H. Zhang, *Optimal scheduling for regional integrated energy system operation based on the AMOWOA*, Acta Automatica Sinica **50** (2024), 576–588.

[15] J. Hong, B. Gao, S. Dong, Y. Cheng, Y. Wang and H. Chen, *Key problems and research progress of energy saving optimization for intelligent connected vehicles*, China Journal of Highway and Transport **34** (2021), 306–334.

[16] H. Huang, Y. Wang and H. Zong, *Support vector machine classification over encrypted data*, Applied Intelligence **52** (2022), 5938–5948.

[17] L. Jin, W. Li and Q. Zhu, *Research on simulation and optimization of processing technology of gear parts of agricultural machinery based on ant colony algorithm*, Journal of Agricultural Mechanization Research **44** (2022), 41–44.

[18] P. Kumari and S. K. Sahana, *PSO-DQ: An improved routing protocol based on PSO using dynamic queue mechanism for MANETs*, J. Inf. Sci. Eng. **38** (2022), 41–56.

[19] T. Y. T. Lam, R. S. Y. Tang, J. Y. L. Ching, S. C. Ng, M. H. Kyaw and J. J. Y. Sung, *Artificial intelligence-assisted colonoscopy improves adenoma detection in screening colonoscopy. A large-scale multi-center randomized controlled study*, Journal of Gastroenterology and Hepatology **36** (2021), 65–65.

[20] R. Maithani, R. Agarwal, A. Kumar and S. Sharma, *Parametric optimization of impinging air jet on hemispherical protrusion of a solar thermal collector*, Experimental Heat Transfer **36** (2023), 786–807.

[21] M. Nandipati, O. Fatoki and S. Desai, *Bridging nanomanufacturing and artificial intelligence-a comprehensive review*, Materials **17** (2024), 1621–1621.

[22] D. Pelusi, R. Mascella, L. Tallini, J. Nayak, B. Naik and A. Abraham, *Neural network and fuzzy system for the tuning of Gravitational Search Algorithm parameters*, Expert Systems with Applications **102** (2018), 234–244.

[23] R. Qi, Z. Wang and S. Li, *A parallel genetic algorithm based on spark for pairwise test suite generation*, J. Comput. Sci. Technol **31** (2016), 417-427.

[24] J. Tang, G. Liu and Q. Pan, *A review on representative swarm intelligence algorithms for solving optimization problems:applications and trends*, IEEE/CAA Journal of Automatica Sinica **8** (2021), 1627–1643.

[25] V. Teslyuk, A. Kazarian, N. Kryvinska and I. Tsmots, *Optimal artificial neural network type selection method for usage in smart house systems*, Sensors **21** (2021): 47.

[26] O. E. Turgut, M. S. Turgut and E. Kirtepe, *A systematic review of the emerging metaheuristic algorithms on solving complex optimization problems*, Neural Computing & Applications **35** (2023), 14275–14378.

[27] W. Wang, *Optimization design of intelligent module of CNC lathe for agricultural machinery processing based on case-based reasoning and artificial intelligence*, Journal of Agricultural Mechanization Research **42** (2020), 204–208.

[28] Y. Wu and J. Feng, *Development and application of artificial neural network*, Wireless Personal Communications **102** (2018), 1645–1656.

[29] J. Xia, S. S. Dou, Z. Yang, Q. Li and J. Xia, *Intelligent optimization algorithm and the application in mechanical design*, Journal of Chemical and Pharmaceutical Research **6** (2014), 2830–2843.

[30] Y. Xu, Y. Zhong and Z. Huang, *An improved blind recognition method of the convolutional interleaver parameters in a noisy channel*, IEEE Access **7** (2019): 101775–101784.

[31] L. Xue, J. Wu, W. Song, J. Wang, H. Qi and Y. Wang, *Preferential decision making and pricing optimization model for commercial operation of power towers*, Price:Theory & Practice **4** (2023), 170–174.

[32] L. Yang, B. Yang and X. Gu, *Adversarial reconstruction CNN for illumination-robust frontal face image recovery and recognition*, International Journal of Cognitive Informatics and Natural Intelligence **15** (2021), 18–33.

[33] Y. Zeng, B. Liao, Z. Li, C. Hua and S. Li, *A comprehensive review of recent advances on intelligence algorithms and information engineering applications*, IEEE Access **12** (2024), 135886–135912.

[34] X. Zhang, Z. Zhao, Z. Guo and W. Zhao, *Research on machining parameter optimization in finishing milling with multiple constraints*, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture **236** (2022), 968–980.

[35] X. Zhao, X. Jia, T. Zhang, T. Liu and Y. Cao, *A supervised surrogate-assisted evolutionary algorithm for complex optimization problems*, IEEE Transactions on Instrumentation and Measurement **72** (2023), DOI:10.1109/TIM.2023.3261905.

[36] W. Zhou, S. Li, G. Ma, X. Chang, X. Ma and C. Zhang, *Parameters inversion of high central core rockfill dams based on a novel genetic algorithm*, Sci. China Technol. Sci. **59** (2016), 783–794.

Manuscript received April 16, 2024
revised October 12, 2024

Q. Gao
School of Intelligent Manufacturing, Taizhou University, Taizhou, China
E-mail address: gaows@tzc.edu.cn; wei_qin@foxmail.com

J. Zhang
School of Intelligent Manufacturing, Taizhou University, Taizhou, China
E-mail address: 1165623887@qq.com

X. Tao
Zhejiang Ruiling Enterprise Consulting Co., Ltd, Taizhou, China
E-mail address: xueding-t@chinaruiling.cn

W. Gao
School of Intelligent Manufacturing, Taizhou University, Taizhou, China
E-mail address: gao.weishang@hotmail.com

L. Sun
School of Electronics and Information Engineering, Taizhou University, Taizhou, China
E-mail address: lijiesun@tzc.edu.cn

Y. Chen
School of Intelligent Manufacturing, Taizhou University, Taizhou, China
E-mail address: chenyue@tzc.edu.cn

P. He
School of Electronics and Information Engineering, Taizhou University, Taizhou, China
E-mail address: hphgy@tzc.edu.cn