

# Research on Personalized Training Model of College Vocal Singing Skills Based on Fuzzy Control Algorithm

Ge Zhang \*

School of Arts, Zhaotong University, Zhaotong, Yunnan, 657000, China; Zhang9654411@126.com

**Abstract:** This paper first identifies three commonly used personalized learning parameters—“learning objectives, learning styles, and cognitive levels”—based on learners' needs. Subsequently, it analyzes the recommendation of personalized learning paths from two dimensions: learners and learning resources, and constructs a personalized learning path recommendation model for vocal singing techniques in higher education institutions. Combining three strategies—division of labor strategy, parameter adaptive adjustment strategy, and simulated annealing strategy incorporating distance factors—to enhance the algorithm's ability to escape local optima, this study proposes a division-of-labor and fuzzy control-based particle swarm optimization algorithm (LDFSPSO) for model solution. Results indicate that the LDFSPSO algorithm can rapidly identify the global optimum of single-peak functions and demonstrates significant advantages when handling multi-peak functions with multiple local extrema points of similar magnitude. The proposed method achieves greater diversity in the resource recommendation sequences for learners and better recommendation effectiveness. Case analysis results also indicate that the LDFSPSO algorithm is more effective in identifying resources that align with learners' needs, and can recommend resources with higher matching degrees for learners with different characteristics.

**Keywords:** personalized learning parameters; fuzzy control; particle swarm optimization; vocal singing techniques

## 1. Introduction

For vocal performance, in addition to mastering advanced vocal techniques, possessing a high degree of individuality and creativity is particularly crucial. This not only prevents vocal performances from becoming formulaic but also highlights one's distinctive artistic style, making it easier to demonstrate exceptional professional competence [1-3]. Therefore, in the process of vocal music education at higher education institutions, emphasizing the individualized characteristics of talent cultivation and respecting students' unique traits is of great importance [4]. Vocal music teachers should help students break the habit of blindly imitating others, explore diverse approaches, and assist them in developing their own unique singing styles, thereby infusing vocal music performance with innovation and variety, and enabling students to achieve better development in vocal music artistry [5-6]. Therefore, given the current state of vocal music education in higher education institutions, the idea of strengthening the cultivation of personalized singing styles is proposed. This involves consciously guiding and implementing differentiated cultivation based on students' professional foundations and singing characteristics, thereby prompting them to form unique singing styles.

However, the phenomenon of a “one-size-fits-all” approach is prevalent in university vocal music education, primarily manifested in the identical goals and content of vocal music instruction for all students [7]. Additionally, vocal performance evaluations follow uniform standards. While each student possesses unique characteristics and personal styles in vocal performance, the standardized teaching model restricts their free development, forcing them to sing according to the teacher's requirements and lacking individual singing characteristics [8-9]. This also leads to students having similar singing styles with high similarity, resulting in vocal performances lacking charm and appeal, unable to achieve rich artistic expression, and students struggling to attain higher levels of vocal artistry [10-11]. Additionally, a



common issue in college vocal music education is the emphasis on training students' vocal techniques, while lacking systematic training and guidance on how to express emotions through voice during singing, thereby hindering the comprehensive improvement of students' vocal music literacy [12-13].

Against this backdrop, many scholars have proposed innovative training methods. For example, Sun integrated artificial intelligence (AI) technology to develop a vocal muscle capability training system, utilizing Variational Onsager Neural Networks (VONN) and Golden Search Optimization Algorithms (GSOA) to achieve personalized vocal training, resulting in significantly improved outcomes [14]. Zhang and Cheng employed neural networks and virtual reality technology to optimize vocal training, proposing a novel vocal performance simulation and training system, which offers students a more engaging and interactive experience, significantly improving their vocal control and consistency [15]. Shi et al. combined mobile internet platforms and applications to modernize vocal training, integrating technical solutions with teaching methods to provide students with personalized learning experiences [16]. Doganyigit and Islim investigated the impact of virtual reality on vocal performance training for music students, finding that it enhances breathing control, resonance effects, and vocal techniques, resulting in more powerful and comfortable singing performances [17]. Guo developed an adaptive vocal performance training system based on transfer learning and genetic algorithms, which continuously refines vocal training plans through multiple iterations, providing personalized vocal training schemes tailored to learners' vocal characteristics and learning requirements [18]. Shi's research aimed to improve modern music education through digital technology, focusing on the impact of auditory games for mastering auditory skills, real-time singing recording, and other functions on vocal training. The results confirmed that these technologies can serve as effective tools to provide students with more constructive feedback and personalized training plans [19].

This paper first describes the various characteristics and needs of learners through personalized learning parameters, then solves for the minimum value of the objective function based on the differences between learners and learning resources, thereby constructing a personalized learning path recommendation model. To further address the multi-modal, high-dimensional, and complex optimization issues in the personalized training process of vocal singing techniques in higher education institutions, this paper proposes a particle swarm optimization algorithm based on division of labor and fuzzy control (LDFPSO). This algorithm is optimized based on the standard particle swarm algorithm using three improved strategies: "division of labor strategy, parameter adaptive adjustment strategy, and simulated annealing strategy incorporating distance factors." By introducing the "probabilistic acceptance of inferior solutions" concept from simulated annealing, the algorithm enhances the population's ability to escape local optima and avoids getting stuck in local optima. Finally, experiments validate the effectiveness of this algorithm in personalized training for vocal performance techniques in higher education.

## **2. Personalized training model based on vocal singing techniques in colleges and universities**

### *2.1. Personalized training model based on vocal singing techniques in colleges and universities*

Personalized learning parameters are the fundamental basis for constructing learner models, describing various characteristics and needs of learners, such as their learning objectives, learning styles, and cognitive levels. Researchers often consider utilizing different learning parameters to construct learner models. Commonly used personalized learning parameters are as follows:

(1) Learning objectives: Learning objectives refer to the set of knowledge points that learners are required to master during the learning process. They are used to design and plan the learning process, arranging the learning of knowledge points in a path-like manner to meet the learners' objectives.

(2) Cognitive level: Cognitive level is used to describe the current learner's ability to master knowledge, indicating the extent to which the learner has mastered the knowledge or skills required for a particular course or knowledge point.

(3) Learning Style: Learning style refers to the consistent and stable learning methods exhibited by learners during the learning process, which may include visual, verbal, reading, writing, or other styles. There are many models available to analyze and determine learners' learning styles. The Felder and Silverman Learning Style Inventory is the most frequently used model in the field of personalized learning paths. It categorizes learners into four distinct dimensions based on their learning preferences, with two distinct characteristics under each dimension: Perceiving vs. Intuitive, Visual vs. Verbal, Active vs. Reflective, and Sequential vs. Holistic.

According to Felder and Silverman's theory, a learner's learning style can be represented by four key pairs, and the learning style formula is:

$$LS = [(T_1, e_1), (T_2, e_2), (T_3, e_3), (T_4, e_4)] \quad (1)$$

In the formula:  $T_i$  represents the specific type of learning style, and  $e_i$  is a fuzzy value less than 1.

To obtain the corresponding values for learning styles across each dimension, the Filder-Silverman Learning Style Index Survey Questionnaire (ILS) can be used to measure learning styles. The ILS consists of 40 questions divided into four groups, with 10 questions per group. Each question offers two options, and the learner's responses are used to assess their learning style preferences.

(4) Available learning time: This refers to the time learners can devote to learning activities. In current personalized learning path planning methods, the time required to complete a learning path is typically determined by the learning system, and learners can only passively accept it, leading to issues where insufficient time is allocated to complete the learning path. Therefore, the construction of personalized learning paths should consider whether learners can efficiently complete learning tasks within their available time.

## 2.2. Building a personalized training path model for vocal performance techniques in colleges and universities

### 2.2.1. Learner Model Construction

(1) Define learners  $L = \{L_1, L_2, L_3, \dots, L_K\}$ ,  $K$  represents the number of learners,  $L_k$  represents the  $k$  th learner, where  $1 \leq k \leq K$ .

(2) Define the ability level of learners  $A = \{A_1, A_2, A_3, \dots, A_K\}$ , where  $A_k$  represents the learning ability level of learner  $L_k$ , where  $1 \leq k \leq K$ .

(3) Kolb's learning styles are divided into four types: divergent thinking, absorptive, convergent thinking, and adaptive thinking. Therefore, define the learning styles of  $K$  learners as  $LS = \{ls_1, ls_2, ls_3, \dots, ls_k\}$ , where  $ls_k$  denotes the style type of the  $k$  th learner,  $ls_k = \{ls_{k1}, ls_{k2}, ls_{k3}, ls_{k4}\}$ , where  $ls_{k1}$  represents the matching value for learners belonging to the divergent thinking type, similarly  $ls_{k2}$  represents the matching value for learners belonging to the absorptive type,  $ls_{k3}$  represents the matching value for learners belonging to the convergent thinking type, and  $ls_{k4}$  represents the matching value for learners belonging to the adaptive type. The learner's preference for a particular learning style is represented by the maximum matching value for that style, and satisfies  $\sum_{q=1}^4 e_q = 1$ .

(4) Define the learner's target knowledge points  $H = \{H_1, H_2, H_3, \dots, H_K\}$ , where  $H_k$  represents the target knowledge points of the  $k$  th learner, and each  $H_k$  has  $M$  binary values, then  $H_k = \{h_{k1}, h_{k2}, h_{k3}, \dots, h_{kM}\}$ , where  $1 \leq k \leq K$  and  $1 \leq m \leq M$ . If  $h_{km} = 1$ , then the  $m$  th knowledge point of the  $k$  th learner is a target knowledge point, otherwise  $h_{km} = 0$ .

(5)  $Lt$  is the time period within which the learner wants to master the target knowledge point,  $Lt_l \leq Lt \leq Lt_u$ , where  $Lt_l$  is the minimum value of the target time, and  $Lt_u$  is the maximum value of the target time.

### 2.2.2. Construction of a Learning Resource Model

(1) Define learning resources  $S = \{S_1, S_2, S_3, \dots, S_N\}$ , where  $S_n$  represents the  $n$  th learning resource, with  $1 \leq n \leq N$ . The learning resource that the learner starts learning is  $ns_i$ , and the next learning resource that the learner will learn is  $ns_j$ .

(2) Define the knowledge points  $U = \{U_1, U_2, U_3, \dots, U_M\}$ ,  $U_m$  to represent the first  $m$  knowledge point. Each  $U_m$  has  $N$  learning resources corresponding to it, then

$U_m = \{U_{m1}, U_{m2}, U_{m3}, \dots, U_{mN}\}$ , when  $U_{mn}$  represents the difference between learning the first  $m$  knowledge point and the corresponding learning resources,  $1 \leq n \leq N$ ,  $1 \leq m \leq M$ .

(3) Define the difficulty level of learning resources as  $D = \{D_1, D_2, D_3, \dots, D_N\}$ , where  $D_n$  represents the difficulty level of the  $n$ th learning resource, then the difficulty level of  $ns_i$  is  $D_{ns_i}$ , and the difficulty level of  $ns_j$  is  $D_{ns_j}$ .

(4) The time information of learning resources is  $T = \{T_1, T_2, T_3, \dots, T_N\}$ , where  $T_n$  represents the time required to learn the  $n$ th learning resource.

(5) The medium types of learning resources include text, video, audio, images, interactive learning software, and other types, so the medium type is defined as  $MT = \{mt_1, mt_2, mt_3, \dots, mt_N\}$ , where  $m_n$  denotes the resource type to which the  $n$ th learning resource belongs, and  $m_n = \{m_{n1}, m_{n2}, m_{n3}, m_{n4}\}$ , where  $mt_{n1}$ ,  $mt_{n2}$ ,  $mt_{n3}$ ,  $mt_{n4}$  respectively represent the matching values of learning resources with text, symbols (charts, animations, etc.), audio/video, and interactive software expressions, and  $\sum_{q=1}^Q mt_{nq} = 1$ .

### 2.2.3. Decision variables

Define the personalized learning path variable as a matrix  $X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1N} \\ x_{21} & x_{22} & \dots & x_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{NN} \end{bmatrix}$  is defined

as follows: if there is a path between the  $i$ th learning resource and the  $j$ th learning resource, then  $x_{ij} = 1$ ; otherwise,  $x_{ij} = 0$ .

Define the sequence relationship between learning resources:

$$S_{ij} = \begin{cases} 1, & \text{Learning resource } i \text{ is the predecessor of } j. \\ 3, & \text{Learning resource } i \text{ is the successor of } j. \\ 2, & \text{Learning resources } i \text{ and } j \text{ are parallel.} \end{cases} \quad (2)$$

### 2.2.4. Construction of the objective function

Based on the above learner and learning resource models, a mapping relationship between the two was established, forming the following personalized learning path recommendation model.

(1) Learner cognitive level function, which reflects the difference between the learner's current learning ability level and the difficulty level of the learning resource. Then, this learning resource is more likely to be recommended to the learner:

$$F_1 = \sum_{j=1}^N \left| \frac{\sum_{i=1}^N [X_{ij} (D_{ns_i} - A_k) + X_{ij} (D_{ns_j} - A_k)]}{2 \sum_{i=1}^N X_{ij}} \right| \quad (3)$$

(2) Learning style function, which represents the differences between learners' learning styles and the types of learning resource media:

$$F_2 = \frac{\sum_{n=1}^N \sum_{q=1}^Q X_{ij} |ls_{kq} - mt_{nq}|}{X_{ij}} \quad (4)$$

(3) Expected target expenditure function, representing the match value between the learner's target knowledge points and the corresponding learning resources, as well as the sequence relationship between

learning resources:

$$F_3 = \frac{\sum_{m=1}^M \sum_{n=1}^N X_{ij} U_{mn} H_{km} S_{ij}}{X_{ij}} \quad (5)$$

(4) Time constraint function, which represents the difference between the learner's desired learning duration and the time required to complete the learning resources. If the time spent on the recommended learning path is close to the specified time limits, the value of the function will be smaller, indicating that the path from learning resource  $i$  to  $j$  is more likely to be recommended to the learner. That is:

$$F_4 = \left( \max \left( t_k - \sum_{n=1}^N t_n X_{ij}, 0 \right) \right) + \left( \max \left( 0, \sum_{n=1}^N t_n X_{ij} - t_k \right) \right) \quad (6)$$

The above are the objective functions for the learner model and learning resource model. Since each objective function has a different degree of influence on learners, the overall objective function is:

$$\min F(x) = \sum_{i=1}^4 w_i F_i \quad (7)$$

The objective function is a problem of finding the minimum value, which ultimately allows us to find a set of optimal decision variables in the solution space. The cost incurred by learners on the learning path corresponds to the value of the objective function. Therefore, when the value of the objective function is smaller, the cost incurred by learners on that path is also smaller, and thus the generated personalized learning path better meets the learners' requirements.

### 2.3. Model solving based on particle swarm algorithm with fuzzy control

#### 2.3.1. Three Improvement Strategies

##### (1) Division of labor strategy

First, particles are sorted from best to worst based on fitness and position factors. For this purpose, the following rules are specified: the smaller the objective function value corresponding to a particle, the better the particle; the closer a particle is to the globally optimal particle  $Gbest$ , the better the particle. The distance between particle  $Gbest$  and  $dis\ tan\ ce\_gb$  is:

$$dis\ tan\ ce\_gb_i^k = \sqrt{\sum_{i=1}^D (x_{iD}^k - gbest_D^k)^2} \quad (8)$$

Sort the particles in descending order of fitness, and denote the resulting fitness ranking vector as  $rankf = (rf_1, \dots, rf_i, \dots, rf_N)$ , where  $rf_i$  is the rank of the  $i$ th particle sorted by fitness factor. Sort the particles from best to worst based on the position factor, and denote the position ranking vector as  $rankx = (rx_1, \dots, rx_i, \dots, rx_N)$ , where  $rx_i$  is the rank of the  $i$ th particle based on the position factor. Define the sum of the two ranking vectors as the comprehensive ranking vector, denoted as  $rankc = (rc_1, \dots, rc_i, \dots, rc_N)$ , and sort each particle according to its  $rc_i$  value from smallest to largest. If particles have the same  $rc_i$  value, they are sorted according to  $rf_i$ . If particles still have the same  $rf_i$  value, they are sorted according to  $rx_i$ . If particles still have the same  $rx_i$  value, they are sorted according to their original ranks, resulting in a new sorted population. The individual optimal update rate  $r_{pb}$  is defined as follows:

$$r_{pb} = \frac{n_{update}}{N} \quad (9)$$

Among these,  $n_{update}$  represents the number of particles whose optimal values are updated in a single iteration.  $r_{pb}$  denotes the ratio of particles whose optimal values are updated in a single iteration, reflecting the proportion of particles with potential in the current iteration. Based on  $r_{pb}$ , the population is divided into two parts: the front part consists of "scout particles," and the rear part consists of "rear guard particles." Scout particles abandon the learning strategy and perform random scouting flights based on the better particles in the population to search for potential optimal solutions and escape from

possible local optima. Inspired by the movement behavior of leaders in the sea squirt swarm algorithm [20], random scouting flights are performed as follows:

$$x_{id}^{k+1} = \begin{cases} x_{id}^k + \alpha [(ub_d - lb_d)\beta + lb_d], \gamma \geq 0.5 \\ x_{id}^k - \alpha [(ub_d - lb_d)\beta + lb_d], \gamma \leq 0.5 \end{cases} \quad (10)$$

$$\alpha = 2e^{-\frac{4k}{k_{\max}}} \quad (11)$$

Among them,  $\alpha$  is a decay factor that decreases with iteration, used to control the degree of random reconnaissance flight,  $ub_d$  and  $lb_d$  are the upper and lower bounds of the search space in the  $d$  th dimension, respectively, and  $\beta$  and  $\gamma$  are random numbers uniformly distributed on  $[0, 1]$ .

### (2) Parameter adaptive adjustment strategy

In the standard particle swarm algorithm [21] and its variants, the settings of the three parameters  $\omega$ ,  $c_1$ , and  $c_2$  have a significant impact on the performance of the algorithm. Adaptively adjusting these parameters based on the iteration situation and population characteristics will improve the performance of the algorithm.  $\omega$  influences the particles' inheritance of the movement trends of the previous generation. To ensure that the algorithm maintains strong exploration capabilities in the early stages and strong exploitation capabilities in the later stages, and that the value of  $\omega$  changes slowly in both the early and late stages of the algorithm, enabling the algorithm to fully learn from  $Pbest$  and  $Gbest$ , a Sigmoid function is designed to adaptively adjust the value of  $\omega$ , such that  $\omega$  decreases with the number of iterations, i.e.:

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{1 + e^{-15\left(\frac{k-1}{k_{\max}-2}\right)}} \quad (12)$$

Among these,  $\omega_{\max}$  and  $\omega_{\min}$  represent the maximum and minimum values of the inertial weights, respectively, while  $k_{\max}$  denotes the maximum number of iterations.

After introducing the division of labor strategy, some particles perform reconnaissance flights, which increases the diversity of the population. To balance the algorithm's exploration and exploitation capabilities, considering the fuzzy system's ability to mimic human flexible decision-making, a fuzzy inference system is used to achieve adaptive adjustment of  $c_1$  and  $c_2$ . Taking the iteration status and population diversity as input variables, the centroid method is used for defuzzification, and a dual-input dual-output fuzzy system is designed. Considering that the fuzzy system input should be defined in  $[0, 1]$  and the iteration process is normalized as  $k\_norm$ , we have:

$$k\_norm = \frac{k}{k_{\max}} \quad (13)$$

Population diversity can be used to measure the degree of aggregation and dispersion of a population and is an important characteristic of a population. When a population is aggregated, population diversity is low; when a population is dispersed, population diversity is high. Population diversity is defined as the average Euclidean distance of all particles from the population center, normalized to  $diversity\_norm$ , and calculated as follows:

$$diversity^k = \frac{1}{N} \sum_{i=1}^N \sqrt{\sum_{d=1}^D (x_{id}^k - x_{center,d}^k)^2} \quad (14)$$

$$diversity\_norm^k = \frac{diversity^k}{\max \left( \sqrt{\sum_{d=1}^D (x_{id}^k - x_{center,d}^k)^2}, i = 1, \dots, N \right)} \quad (15)$$

Among them,  $x_{center}$  is the center position of the current population.

### (3) Simulated annealing strategy with distance factor fusion

The idea of accepting inferior solutions with probability in the simulated annealing algorithm [22] can enhance the algorithm's ability to escape from local optima. Based on the simulated annealing algorithm, in order to reduce the occurrence of new solutions and old solutions falling near the same local optimum and thus affecting the optimization efficiency, inferior solutions that are too close in distance are not accepted. Define the simulated annealing factor  $p^k$  that incorporates the distance factor. When updating  $Gbest$ , it is allowed to accept poor solutions with probability  $p^k$ , and  $p^k$  decreases with each iteration. The calculation method is as follows:

$$p^k = \begin{cases} 1, f(X_i^k) \leq f(Gbest) \\ \delta \cdot e^{-\frac{f(x_i^k) - f(Gbest)}{T}}, f(x_i^k) > f(Gbest) \end{cases} \quad (16)$$

$$\delta = \begin{cases} 1, d_i^k \geq d_{\min}^k \\ 0, d_i^k < d_{\min}^k \end{cases} \quad (17)$$

$$d_i^k = \sqrt{\sum_{d=1}^D (x_{id}^k - gbest_d^k)^2} \quad (18)$$

$$d_{\min}^k = \min \left\{ \sqrt{\sum_{d=1}^D (x_{id}^k - x_{jd}^k)^2}, i, j = 1, \dots, N \right\} \quad (19)$$

$$T^k = \begin{cases} \frac{f(Gbest)}{\lg 5}, k = 1 \\ 0.95T^{k-1}, k > 1 \end{cases} \quad (20)$$

Among them,  $\delta$  is the decision coefficient,  $d_i^k$  is the Euclidean distance between the new solution  $X_i$  and  $Gbest$ ,  $d_{\min}^k$  is the shortest distance between two particles, and  $T^k$  is the temperature in the  $k$ th iteration.

### 2.3.2. Improving the algorithm process

Based on the above description, the basic process of the proposed LDFPSO algorithm is as follows:

Step 1: Set the algorithm parameters, initialize the particle swarm to obtain the initial population, and set the first generation  $Pbest$ .

Step 2: Calculate the fitness of each particle, set the initial generation  $Gbest$ , and solve  $distance\_gb_i$ ;

Step 3: Set  $r_{pb}$  and  $T$ ;

Step 4: Reorder the particles according to the division of labor strategy and divide the particles into scout particles and rear guard particles;

Step 5: Calculate the input indicators of the fuzzy system, update  $\omega$ , update  $c_1$  and  $c_2$  by the fuzzy system, and update  $\alpha$ ;

Step 6: Update the scout particles and rear guard particles separately;

Step 7: Solve the fitness of each particle, update  $Pbest$ , count  $n_{update}$ , and update  $Gbest$  based on the simulated annealing strategy with the fusion distance factor:

Step 8: Determine whether the termination condition is met. If so, output the optimal fitness value and terminate the algorithm; otherwise, return to Step 3.

To estimate the execution efficiency of the proposed algorithm, we analyze its time complexity. Let  $T(D)$  be the number of times the basic operations are repeated in the algorithm, and  $O(D)$  be the time complexity of the algorithm, where  $D$  is the dimension of the problem and  $N$  is the population size. Then:

(1) Time complexity of population initialization:  $T_1(D) = D * N$ , so the time complexity of this part is  $O(D * N)$ ;

(2) Time complexity of updating particle velocity and position:  $T_2(D) = D * N = O(D * N)$ , so the time complexity of this part is  $O(D * N)$ ;

(3) Time complexity of updating the individual optimal and global optimal values of particles:  $T_3(D) = D * N * N * N$ , so the time complexity of this part is  $O(D * N^3)$ .

The time complexity of the LDFPSO algorithm is  $O(D * N^3)$ , so it can be seen that the time complexity of the algorithm in this paper is related to the dimension of the problem, the population size, and the number of iterations.

### 3. Analysis of the effectiveness of personalized recommendations for vocal performance techniques in higher education institutions

#### 3.1. Performance Analysis of Improved Particle Swarm Algorithm

The performance of the improved algorithm was tested using the following four evolutionary algorithm benchmark functions.  $f_1$  and  $f_2$  are single-peak functions, while  $f_3$  and  $f_4$  are multi-peak functions.

① Sphere function:

$$f_1(x) = \sum_{i=1}^n x_i^2 \quad (21)$$

② Discus function:

$$f_2(x) = 10^6 x_1^2 + \sum_{i=2}^n x_i^2 \quad (22)$$

③ Rosenbrock function:

$$f_3(x) = \sum_{i=1}^{n-1} \left( 100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2 \right) \quad (23)$$

④ Ackley function:

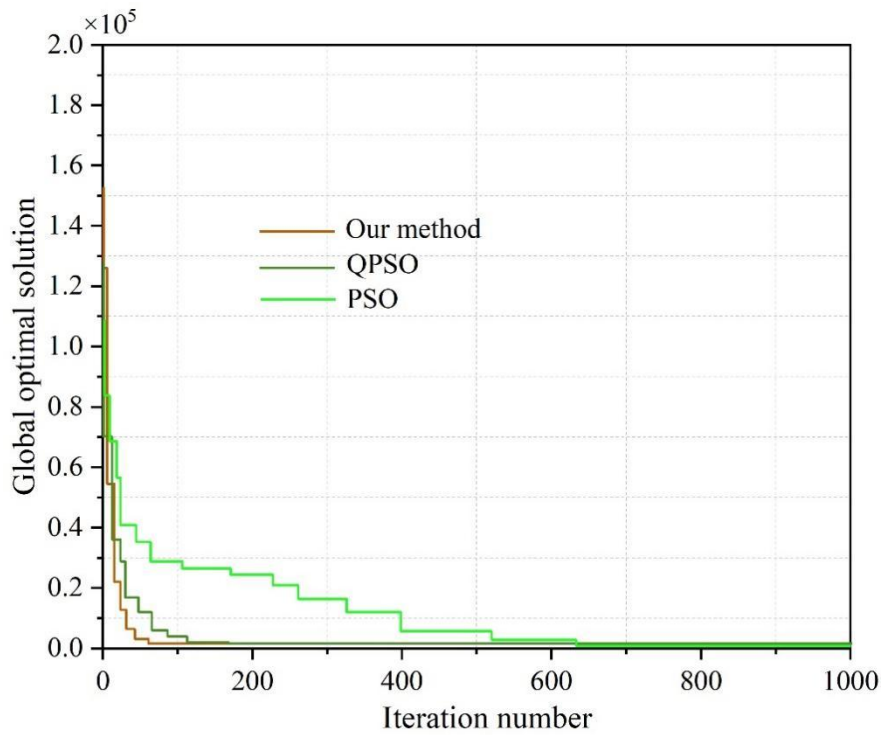
$$f_4(x) = -20 \exp \left( -0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left( \frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + e \quad (24)$$

Each test function was independently tested 30 times, with 1,000 iterations per test and 50 particles. Simulation tests and comparative analyses were performed on the PSO, QPSO, and LDFPSO algorithms. In this experiment, the parameters for each algorithm were set as follows: For the PSO algorithm, the learning factor  $c_1 = c_2 = 2$ , and the inertia weight  $\omega$  was set to decrease linearly; for the QPSO algorithm,  $c_1 = c_2 = 2.05$ , and  $\varphi = 4.1$ ; the LDFPSO algorithm used fuzzy dynamic parameter adjustment. The search space dimension of each test function was  $n = 30$ , and the simulation results (optimal value, worst value, and average value) of each test function were statistically analyzed. The function test results are shown in Table 1. The results show that there are obvious differences in the simulation results of the optimal value, worst value, and average value of the three methods selected in this paper on single-peak and multi-peak functions, such as PSO, QPSO, and LDFPSO on the single-peak function  $f_2$  are 14552.0336, 73263.7268, and 33444.3815, respectively, which are obvious differences. The simulation results show that each test function selected in this paper is highly representative and can be used in the following experiments.

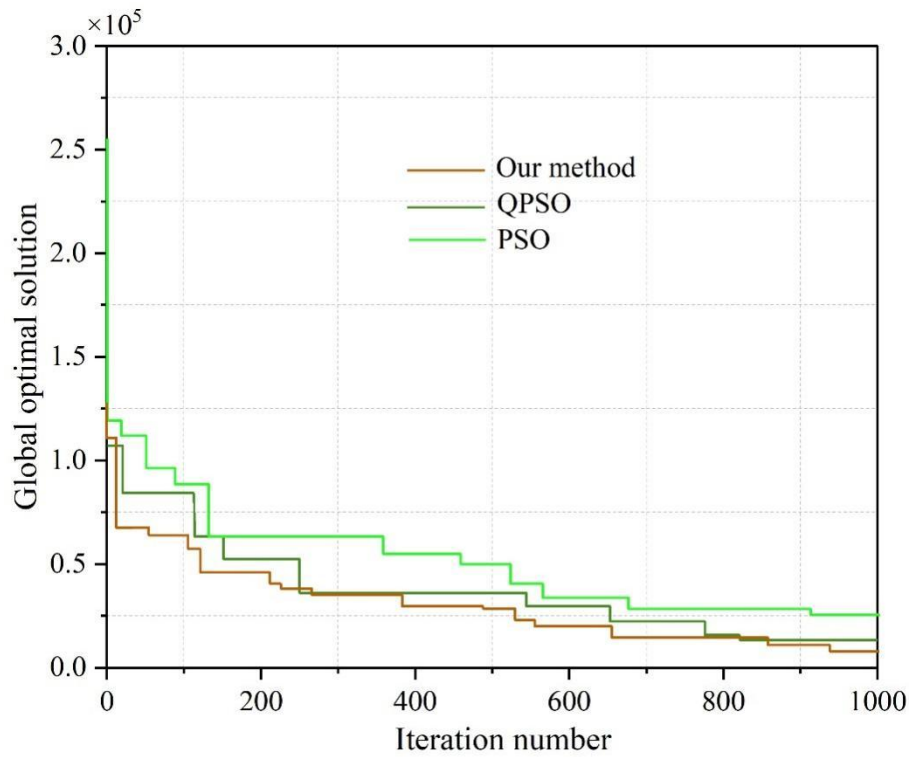
**Table 1.** Function test results.

Test function	Test index	PSO	QPSO	LDFSPSO
$f_1$	Best	4.7753	0.0003	7.7174
	Worst	7.876	7.0448	1.5858
	Average	1.7641	1.4087	6.1192
$f_2$	Best	14552.0336	73263.7268	33444.3815
	Worst	10689.9275	63993.1177	34514.9045
	Average	10125.6901	62072.8883	32834.8615
$f_3$	Best	19.1941	171.7253	111.2331
	Worst	22.6111	187.9737	108.0414
	Average	0.0566	162.0489	95.0865
$f_4$	Best	21.1897	21.3192	20.2549
	Worst	20.1723	20.7254	21.5398
	Average	20.1288	21.8858	20.5498

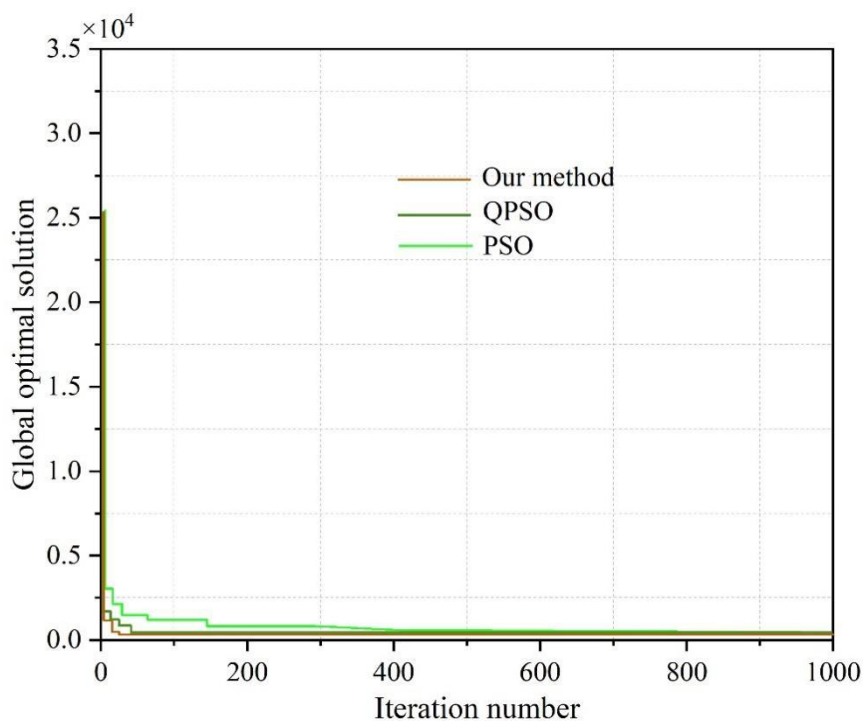
The convergence process of the algorithms for functions  $f_1$  to  $f_4$  is shown in Figure 1, where (a) to (d) represent test functions  $f_1, f_2, f_3$ , and  $f_4$ , respectively. The results show that, compared with the other two methods, the algorithm in this paper has the fastest convergence speed and the fewest iterations required for the four test functions, demonstrating its excellent universality.



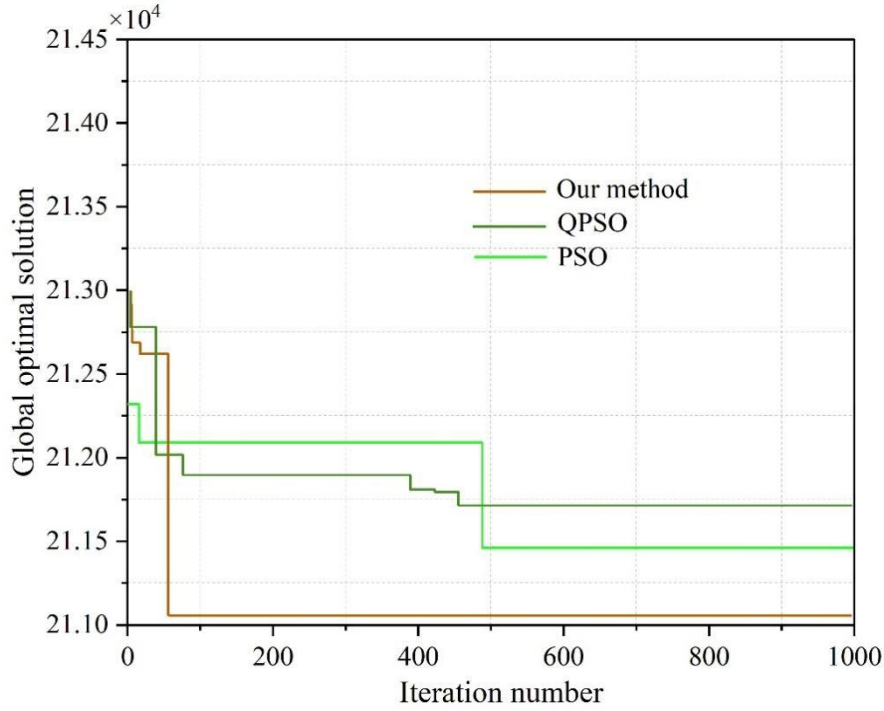
(a) Test function  $f_1$



(b) Test function  $f_2$



(c) Test function  $f_3$



(d) Test function  $f_4$

**Figure 1.** The process of convergence of the  $f_1 \sim f_4$  algorithm of the running function.

To more rigorously compare the performance of the LDFSPSO algorithm and the QPSO algorithm, this paper employs hypothesis testing methods to analyze the experimental data of the LDFSPSO and QPSO algorithms. We first use the K-S test to determine whether the algorithm data follow a normal distribution. When the experimental data of the algorithms follow a normal distribution, we consider using the t-test to assess the significant differences between the LDFSPSO and QPSO algorithms.

The K-S test results for the LDFSPSO and QPSO algorithms are shown in Table 2. The results indicate that not all experimental data for the functions follow a normal distribution, so we do not choose to use the parametric t-test. Since the experimental data does not meet the conditions for parametric testing, we consider using non-parametric testing. The Wilcoxon signed-rank test is more sensitive than the t-test and does not require the differences between paired data to follow a normal distribution.

**Table 2.** LDFSPSO and QPSO algorithm's k-S test results.

Algorithm	$f_1$	$F_2$	$f_3$	$f_4$
QPSO	0.0000	0.2625	0.5081	0.8319
LDFSPSO	0.0000	0.7867	0.9974	0.8206

Therefore, we perform a Wilcoxon signed-rank test on the data. The null hypothesis  $H_0$  states that there is no statistical difference between the experimental results of the two algorithms, while the alternative hypothesis  $H_1$  states that there is a difference between the experimental results of the two algorithms. The results of the particle swarm optimization algorithm based on LDFSPSO are shown in Table 3. The significance symbol “\*” indicates that the LDFSPSO algorithm is superior to the QPSO algorithm. The results show that the LDFSPSO algorithm, by adjusting its global and local search capabilities, can quickly find the global optimal solution for single-peak functions. The LDFSPSO algorithm has a significant advantage over the other two algorithms when dealing with multi-peak functions such as the Ackley function, which have multiple local extrema points with small differences in their extrema values. This is because, although the QPSO algorithm and PSO algorithm improve convergence speed through specific parameter settings, they cannot prevent particles from getting stuck in local optima. During the iteration process of the LDFSPSO algorithm, the learning factor is adaptively adjusted based on the current population's best estimate and the current iteration count, thereby better guiding particles to find the global optimum solution.

**Table 3.** The results of the particle swarm algorithm based on LDFSPSO.

Test index	$f_1$	$F_2$	$f_3$	$f_4$
Normal rank mean	28.7957	27.0362	25.2301	29.7582
Negative rank mean	20.3205	19.5055	0.0000	18.3963
Normal rank sum	365.4703	979.5222	1275.2871	835.1213
Negative rank sum	909.947	298.2123	0.0000	392.1211
P value	0.0088	0.0008	0.0000	0.0281
Sig. symbol	*	*	*	*

### 3.2. Analysis of personalized learning resource recommendation results

#### 3.2.1. Example Analysis

The example selects some knowledge points from the vocal singing techniques section of the Python course. Assuming that learner A is currently studying this course, the initial parameter settings for the experiment are as follows: difficulty level: level 3, expected learning time: 15-30 minutes, media type preference: video, content type preference: singing technique details.

The LDFSPSO algorithm is then used to recommend a personalized sequence of learning resources to the learner, based on the parameter and objective function construction rules of the LDFSPSO recommendation model. In this experiment, 100 learning resources are set, and all learning resources are calculated against the initial parameter conditions of the experiment to optimize and obtain the recommended resource sequence results. The recommendation results of the LDFSPSO model are shown in Table 4, where 1 indicates that the learning resource with that sequence number is recommended to learner A, and 0 indicates that it is not recommended. It can be observed that the 1st, 2nd, 5th, 6th, 73rd, 86th, 97th, 98th, and 100th learning resources are recommended resources, indicating that these resources meet the initial parameter conditions of learner A.

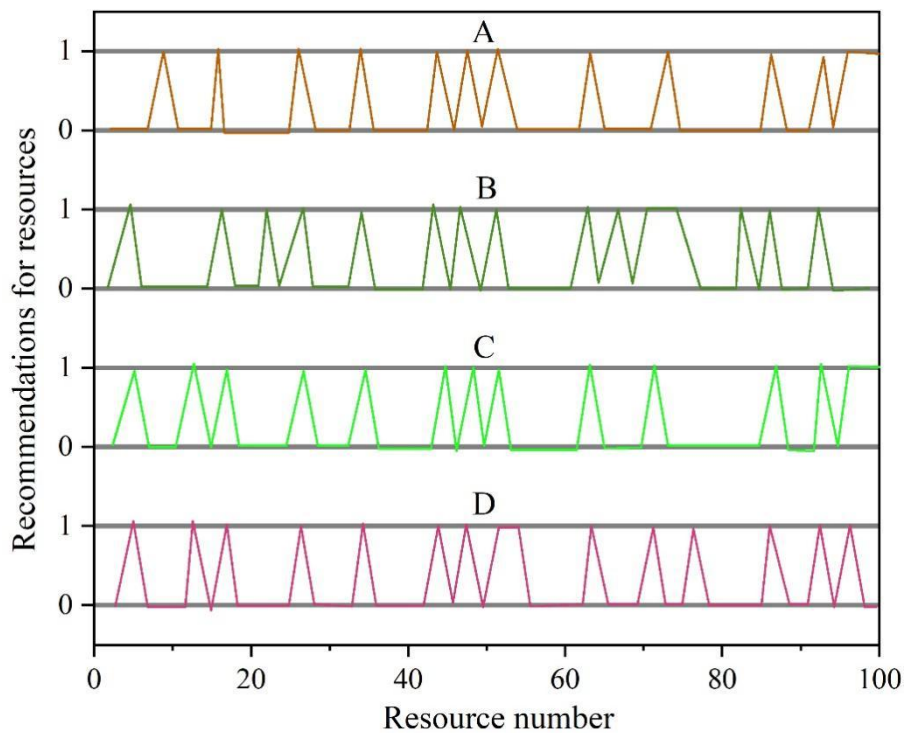
**Table 4.** The recommendation of the LDFSPSO model.

Resource number	Recommended results
1	1
2	1
3	0
4	0
5	1
6	1
...	0
73	1
...	0
86	1
...	0
97	1
98	1
99	0
100	1

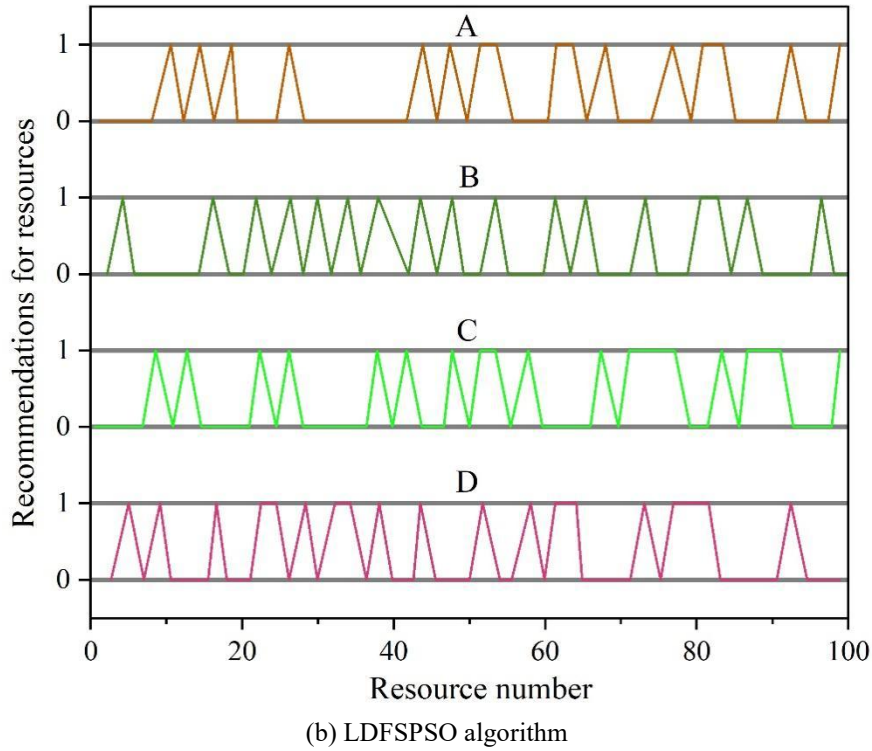
### 3.2.2. Analysis of Differences in Personalized Learning Resource Recommendation Sequences

In order to verify that the generated personalized learning resource recommendation sequence has a differential advantage, four learners were randomly generated, namely Learner A, Learner B, Learner C, and Learner D. The initial parameter conditions of each learner are not necessarily the same. Next, 100 learning resources were randomly generated and sorted in order from 1 to 100, with the order remaining fixed. Among the 100 learning resources, several resources would match each of the four learners, resulting in each learner having a personalized learning resource recommendation sequence. Based on the objective function constructed in this paper, the differences between the learning resource features and learner features were calculated.

The recommendation sequences of the two algorithms are shown in Figure 2, where (a) and (b) represent the QPSO algorithm and the LDFSPSO method, respectively; 1 indicates that the resource with that serial number is recommended, and 0 indicates that it is not recommended. In the previous convergence and accuracy comparison, the mean optimal solution of LDFSPSO was much smaller than that of other algorithms, including QPSO, indicating that the learning resources obtained through LDFSPSO optimization best match the needs of learners. Since the initial parameters (i.e., needs) of the four learners are different, under this premise, the resource sequences obtained by the four learners for 100 learning resources should actually be different. Additionally, the resource recommendation sequences for the four learners show little variation, indicating that QPSO is less sensitive to feature differentiation when targeting different learners. In the LDFSPSO algorithm, the resource recommendation sequences for the four learners exhibit significant variation. Under the condition of a smaller average optimal solution, the resource sequences recommended by LDFSPSO demonstrate a significant advantage in terms of variability.



(a) QPSO algorithm



**Figure 2.** The recommendation sequence of two algorithms.

### 3.3. Analysis of examples of personalized learning resource recommendations

This case study focuses on selected key points from a personalized training course on vocal performance techniques for university students. Assuming there is a learner A who intends to take this course, based on the description of three-dimensional features and the construction rules of four objective functions, and ensuring that the case can specifically recommend the optimal learning resources, the experimental setup initializes parameters using random initialization. According to the method described in this paper, by constructing the objective functions, the learning resources are calculated based on the experimental initialization parameters, and the fitness function is optimized using an improved algorithm to obtain the recommended resource sequence results. The recommended results are listed in Table 5. 1 indicates that the learning resource with that serial number is recommended to learner A, while 0 indicates that it is not recommended. It can be observed that the learning resources with serial numbers 1, 3, 5, 6, 70, 86, 99, and 100 are recommended resources, indicating that these resources meet the initialization parameter conditions.

**Table 5.** Recommended results.

Resource number	Recommended results
1	1
2	0
3	1
4	0
5	1
6	1
...	0
70	1
...	0

86	1
...	0
...	0
98	0
99	1
100	1

Furthermore, in order to verify that the learning resource sequence recommended by the method in this paper has a differential advantage, learning resources were recommended to four learners, “A, B, C, and D.” The initial parameters of the four learners were different, and the 10 learning resources were numbered from 1 to 10, so that each learner could obtain a resource recommendation sequence. The results of the learning resource recommendation sequence are shown in Figure 3. For the recommendation problem, the objective function constructed in this paper matches resource features with learner features for calculation. In previous experiments, it has been verified that the optimal values obtained by LDFSPSO are significantly smaller than those of other comparison algorithms. Under this premise, it can be observed that the resource sequences recommended by LDFSPSO exhibit significant diversity, while the sequences recommended by QPSO, which was used for comparison, are relatively similar. This indicates that LDFSPSO has strong adaptability to this problem and better capability to identify resources that align with learner needs. For learners with different characteristics, it can recommend resources with higher matching degrees.

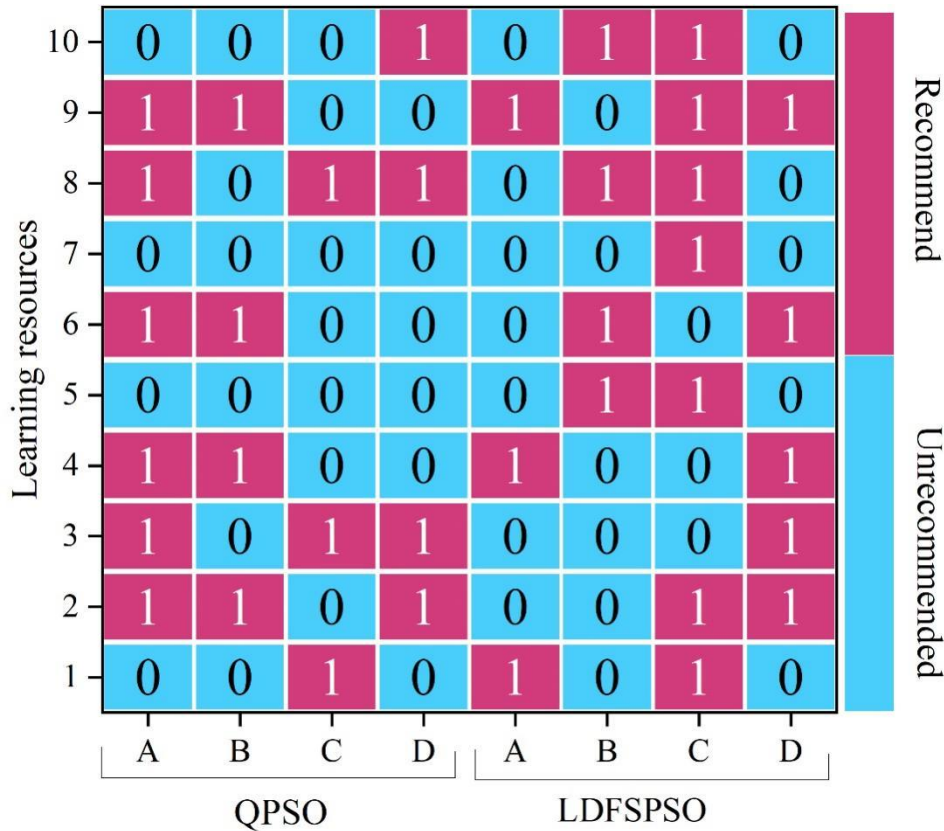


Figure 3. Learning resource recommendation sequence results.

#### 4. Conclusion

This paper analyzes the characteristics of learners and learning resources to construct a learning path recommendation model based on two-dimensional features of learners and learning resources. The LDFSPSO algorithm is applied to this model for solution, providing personalized recommendations for training paths in vocal singing techniques for college students.

The LDFPSO algorithm proposed in this paper outperforms the PSO and QPSO algorithms. It converges faster for single-peak functions and effectively avoids getting stuck in local optima for multi-peak functions. Compared to traditional recommendation methods, the LDFPSO algorithm offers greater variability in the recommended resource sequences for learners, providing a significant advantage in terms of diversity. Additionally, the proposed method can recommend the most suitable personalized learning resource sequences, validating its feasibility and efficiency. With the rapid growth of resources and data, future research should explore more intelligent technologies for application in the field of intelligent learning to better meet personalized learning needs.

## References

1. Chen, Z., & Tao, J. (2025). Exploring the Effects of Vocal Training Programs on the Skills and Development of Chorus Performers. *Journal of Voice*.
2. Bauer, W. I., & Dammers, R. J. (2016). Technology in music teacher education: A national survey. *Research Perspectives in Music Education*, 18(1), 2-15.
3. Dorfman, J. (2016). Music teachers' experiences in one-to-one computing environments. *Journal of Research in Music Education*, 64(2), 159-178.
4. Fang, P. E. N. G. (2021). Optimization of music teaching in colleges and universities based on multimedia technology. *Advances in Educational Technology and Psychology*, 5(5), 47-57.
5. Mawusi, E. F., Nkyi-Asamoah, J., & Kwadwo, K. E. (2020). Technology in Music Education: A survey of computer usage in teaching music in selected colleges of education in Ghana. *Technology*, 11(3), 126-151.
6. Lin, F., & Chen, F. (2025). Intelligent Music Education System: Utilizing Algorithms for Personalized Learning Experience. *International Journal of High Speed Electronics and Systems*, 2540253.
7. Lin, S. (2018, June). Innovation of Vocal Music Teaching in Colleges and Universities. In 2018 2nd International Conference on Management, Education and Social Science (ICMESS 2018) (pp. 838-840). Atlantis Press.
8. Chen, F., & Zhao, Y. (2025). Exploring the Reform of Music Education and Teaching Management Models in Ethnic Normal Universities from the Perspective of Aesthetic Education: A Case Study of Qiannan Minzu Normal University. *The Educational Review, USA*, 9(1), 146-153.
9. Bartleet, B. L., Ballico, C., Bennett, D., Bridgstock, R., Draper, P., Tomlinson, V., & Harrison, S. (2019). Building sustainable portfolio careers in music: Insights and implications for higher education. *Music Education Research*, 21(3), 282-294.
10. Longlong, C., & Luen, L. C. (2023). An In-Depth Blended Learning Strategies for Preschool Music Education in Higher Education. *International Journal of Academic Research in Business and Social Sciences*, 1(3), 12.
11. Yang, A. (2024). Research on the Talent Cultivation of Music Education in Colleges and Universities under the Perspective of New Liberal Arts. *Kuram ve Uygulamada Eğitim Bilimleri*, 24(2), 88-106.
12. Chen, Y. (2025). Innovation of music teaching methods in universities based on fuzzy decision support systems and deep learning. *International Journal of Fuzzy Systems*, 1-20.
13. Guo, S. (2025). Design of Vocal Music Performance Teaching System Based on Multimedia Intelligent Platform. *International Journal of High Speed Electronics and Systems*, 2540415.
14. Sun, L. (2024). Enhancing Vocal Performance using Variational Onsager Neural Network and Optimized with Golden Search Optimization Algorithm. *Applied Artificial Intelligence*, 38(1), 2340389.
15. Zhang, S. H., & Cheng, Y. (2025). Vocal Performance Simulation and Training System Optimized by Neural Networks in Virtual Reality Environment. *International Journal of High Speed Electronics and Systems*, 2540232.
16. Shi, Y. (2023). The use of mobile internet platforms and applications in vocal training: Synergy of technological and pedagogical solutions. *Interactive learning environments*, 31(6), 3780-3791.
17. Doganyigit, S., & Islim, O. F. (2021). Virtual reality in vocal training: A case study. *Music Education Research*, 23(3), 391-401.
18. Guo, J. (2025). Adaptive vocal performance training using transfer learning and genetic algorithm: Personalization and optimization. *Journal of Computational Methods in Sciences and Engineering*, 14727978251318800.
19. Şakalar, A., & Gürel, S. (2024). Academic perspectives on the use of digital platforms and mobile applications in vocal training. *Online Journal of Music Sciences*, 9(2), 389-404.
20. Yibo Sun, Lei Yue, Yi Liu, Weitong Chen & Zhe Sun. (2025). The Application of the SubChain Salp Swarm Algorithm in the Less-Than-Truckload Freight Matching Problem. *Applied Sciences*, 15(8), 4436-4436.
21. Zhan Bangshun, An Shun, He Yan & Wang Longjin. (2022). Three-Dimensional Path Planning for AUVs Based on Standard Particle Swarm Optimization Algorithm. *Journal of Marine Science and Engineering*, 10(9), 1253-1253.
22. Ling Zhu & Yuhui Zhu. (2025). Applications of simulated annealing algorithm in mathematical modelling for scheduling problems. *Journal of Computational Methods in Sciences and Engineering*, 25(4), 3050-3065.