

Learning Path Recommendation Method Based on Improved Particle Swarm Optimization with Elite Learning and Adaptive Mutation

Yanan Guo

School of Computer, Xi'an Shiyu University, Xi'an, 710065 China

Abstract: To address the problem that personalized learning path recommendation in online learning struggles to balance knowledge logic constraints with learner adaptation, this paper proposes a personalized learning path recommendation method based on an improved particle swarm optimization algorithm (IHPSO). First, taking the "C Programming" course as an example, this method constructs a course knowledge graph containing 197 knowledge points and 189 prerequisite relations, and generates a knowledge point sequence conforming to instructional logic through topological sorting. Second, a learner model is constructed from three dimensions: cognitive level, learning style, and available learning time, and a comprehensive objective function is designed to quantify the degree of path adaptation. Finally, three improvements are made to the standard particle swarm optimization algorithm: introducing an elite learning strategy to enhance convergence performance, designing an adaptive mutation mechanism to maintain population diversity, and proposing heuristic local search to achieve fine-grained path optimization. Experimental results show that IHPSO achieves an average fitness of 0.2039, which is 62.46% lower than that of the standard binary particle swarm optimization algorithm (BPSO) and 20.17% lower than that of the genetic algorithm (GA), while the convergence speed is improved by 65% compared with GA. The differences in all metrics are statistically significant ($p < 0.001$).

Keywords: Learning path recommendation, particle swarm optimization, elite learning, adaptive mutation, knowledge graph.

1. Introduction

With the increasing abundance of online learning resources, learners are facing the dual dilemma of "information overload" and "learning disorientation" [1]. Personalized learning path recommendation, which can accurately match learning content and reasonably plan learning sequences according to individual learner differences, has become a key technology for improving learning efficiency [2]. However, existing methods struggle to balance the scientificity of knowledge system logic with the satisfaction of learners' personalized needs, which has become a core issue restricting the effectiveness of online learning.

From a technical perspective, existing research can be mainly divided into three categories. The first category is based on log data mining, which discovers learning patterns by analyzing learners' historical behavioral data and makes recommendations accordingly. Such methods can utilize behavioral information generated during real learning processes, but the mining results depend on the completeness and coverage of the data, and they struggle to explicitly express the intrinsic logical relationships of knowledge [3]. The second category is based on feature matching, which customizes differentiated paths by quantifying the degree of feature matching between learners and learning resources. Such methods achieve a high degree of personalization, but in pursuing feature matching, they tend to treat knowledge points as independent units while neglecting the logical associations among them, potentially leading to recommended paths that, although conforming to learner preferences, violate the proper instructional order inherent to the knowledge itself [4]. The third category is based on knowledge graphs, which explicitly express the logical relationships among knowledge points by constructing

knowledge networks, thereby fully ensuring the scientificity and logical coherence of the generated paths. However, such methods suffer from high construction and maintenance costs for knowledge graphs, and the integration with learner characteristics still requires further exploration [5]. It can be seen that balancing knowledge logic with personalized needs remains a core issue that urgently needs to be addressed in current research.

To address the above problems, this paper proposes a personalized learning path recommendation method based on an improved particle swarm optimization algorithm (Improved Hybrid Particle Swarm Optimization, IHPSO). The core idea of this method is as follows: first, a course knowledge graph is constructed and a knowledge point sequence is generated to provide a clear logical framework for learning paths; second, a multidimensional learner characteristic model is established and a comprehensive objective function is designed to quantify the degree of adaptation between the path and the learner; finally, an intelligent optimization algorithm is adopted to search for the optimal path in the solution space under logical constraints. At the algorithmic level, to address the problems of insufficient encoding adaptability and susceptibility to local optima when the standard particle swarm optimization algorithm is applied to discrete space optimization, this study proposes three improvements: introducing an elite learning strategy to enhance convergence performance, designing an adaptive mutation mechanism to maintain population diversity, and proposing heuristic local search to achieve fine-grained path optimization.

2. Related Models

2.1. Knowledge Point Sequence Generation Based on Knowledge Graph

The primary task of personalized learning path recommendation is to construct a scientific and complete course knowledge system. Taking the C Programming course as an example, this study crawls course resources from the Chinese University MOOC platform and extracts 197 core knowledge points through data cleaning, knowledge point identification, and manual verification.

On this basis, a course knowledge graph is constructed with two core types of relations defined. The "containment relation" describes the hierarchical affiliation structure of "chapter → section → knowledge point", totaling 197 relations. The "prerequisite relation" describes the learning order dependencies among knowledge points, reflecting the logical progression of the knowledge system, totaling 189 relations. Based on the graph structure, attributes such as centrality, importance, and difficulty of each knowledge point are calculated. Finally, a topological sorting algorithm is employed to generate a knowledge point sequence that

$$F(P, l) = \underbrace{\omega_1 \frac{1}{m} \sum_{j=1}^m |D_{k_j} - C_l|}_{\text{Difficulty matching}} + \underbrace{\omega_2 \left(1 - \frac{1}{m} \sum_{j=1}^m \text{Sim}(l, k_j) \right)}_{\text{style matching}} + \underbrace{\omega_3 \mathbb{1}_{[T_l \notin P]} \cdot M}_{\text{goal matching}} + \underbrace{\omega_4 \max(0, \sum T_{k_j} - A_l)}_{\text{time matching}} \quad (1)$$

In the above equation, $\text{Sim}(l, k_j)$ is the cosine similarity between the learner's style vector and the media matching vector of knowledge point k_j , T_l is the target knowledge point of the learner, and M is a penalty constant.

2.3. Standard Particle Swarm Optimization Algorithm

Particle Swarm Optimization (PSO) is a typical swarm intelligence algorithm that achieves global search in the solution space by simulating the cooperative foraging behavior of bird flocks. Each particle possesses two attributes, position X_i and velocity V_i , and updates itself by following the personal best position $pbest_i$ and the global best position $gbest$:

$$V_i^{t+1} = \omega V_i^t + c_1 r_1 (pbest_i - X_i^t) + c_2 r_2 (gbest - X_i^t) \quad (2)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (3)$$

Where ω is the inertia weight, c_1 and c_2 are learning factors, and r_1, r_2 are random numbers in the interval $[0, 1]$.

satisfies the prerequisite constraints. The sequence follows the instructional principle of "from easy to difficult and from simple to complex", providing a logical framework for subsequent personalized learning path recommendation [6].

2.2. Learner Model and Matching Function

To achieve personalized recommendation, it is necessary to establish a quantitative matching relationship between learners and knowledge points. The learner model is characterized from three dimensions: (1) Cognitive level C_l : quantified based on the learner's average quiz score, ranging from 1 to 4, corresponding to the difficulty level of knowledge points; (2) Learning styles $l = (s_{l1}, s_{l2}, s_{l3}, s_{l4})$: based on the Felder-Silverman model, comprising four dimensions of visual/verbal and sequential/global; (3) Available learning time A_l : the total amount of time a learner can devote to learning.

Based on the above model, a comprehensive objective function $F(P, l)$ is designed to quantify the degree of adaptation between a learning path P and a learner l , where a smaller value indicates a higher degree of adaptation. The function is composed of four weighted components:

The PSO algorithm has been widely applied in fields such as path planning due to its advantages of fast convergence speed, few parameters, and ease of implementation [7]. However, the standard PSO is designed for continuous space optimization problems. When directly applied to learning path recommendation, which is a discrete combinatorial optimization problem, it faces limitations such as insufficient encoding adaptability and susceptibility to local optima, necessitating targeted improvements.

3. Learning Path Recommendation Method Based on Improved Particle Swarm Optimization

To address the limitations of the standard PSO in discrete space optimization, this section proposes an improved particle swarm optimization algorithm (IHPSO). The algorithm introduces innovations in four aspects: encoding mechanism, elite learning, adaptive mutation, and heuristic local search. The overall process is shown in Fig. 1.

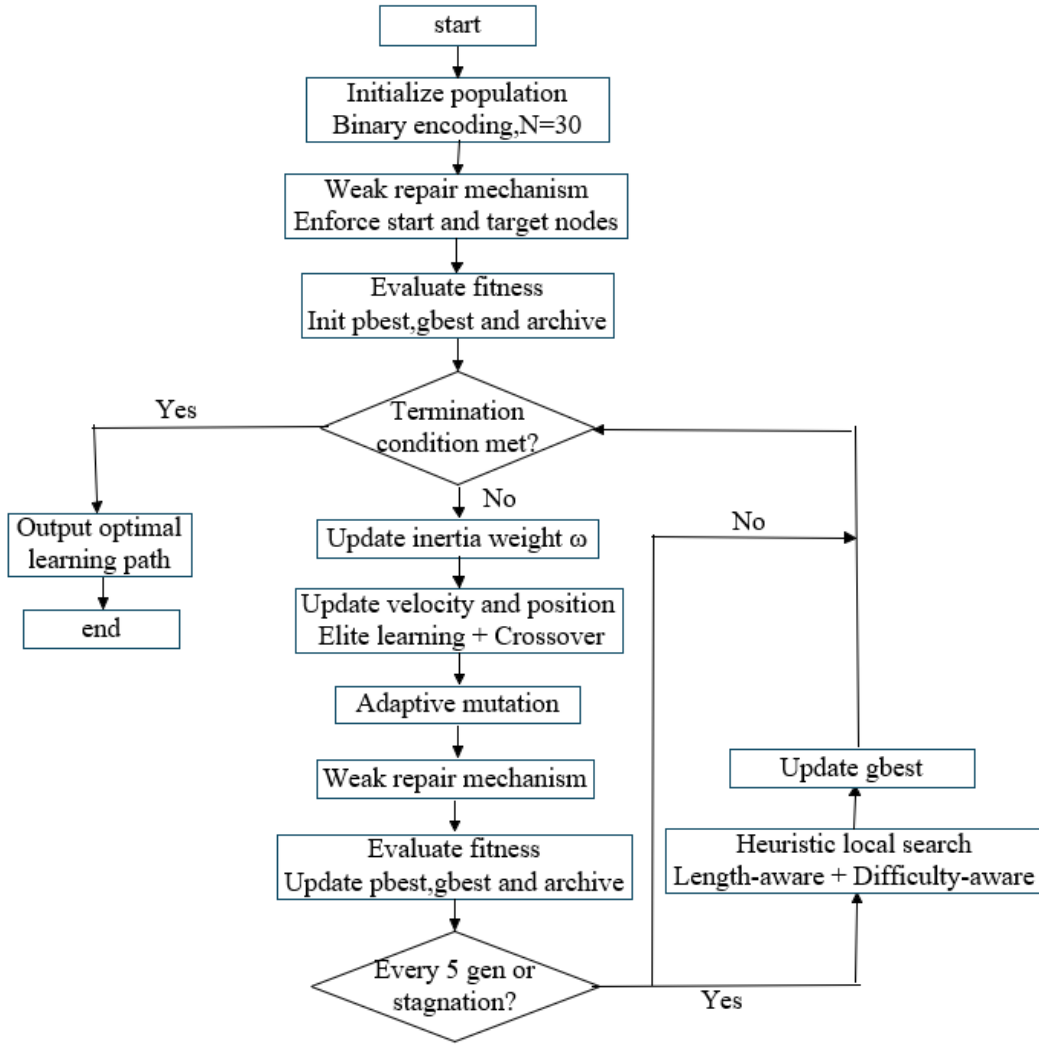


Figure 1. Flowchart of IHPSO Algorithm

3.1. Binary Encoding and Weak Repair Mechanism

The learning path recommendation problem is transformed into a discrete combinatorial optimization problem using binary encoding to represent knowledge point selection states. Each particle is represented as a binary vector $x = x_1, x_2, \dots, x_n$ of length n (total number of knowledge points $n=197$), where $x_i = 1$ indicates that knowledge point kp_i is selected and $x_i = 0$ indicates it is not selected.

To ensure the validity of the learning path, a weak repair mechanism is designed that only enforces that the starting knowledge point and the target knowledge point are always selected, without imposing constraints on other knowledge points, thereby avoiding excessive constraints that could affect the flexibility of the algorithm search:

$$\text{repair}(x) = \begin{cases} x[\text{start_idx}] = 1 \\ x[\text{target_idx}] = 1 \end{cases} \quad (4)$$

3.2. Elite Learning Strategy

To enhance the convergence performance of the algorithm, an elite learning term is introduced into the standard velocity update formula, and an archive of excellent solutions is established to store historical optimal solutions (archive size is set to 8). The elite sample $\text{exemplar}_{\text{exemplar}}$ is selected from the archive, and individuals with larger Hamming distances from the current particle are chosen to maintain population diversity. The velocity update formula is

as follows:

$$v_i^{t+1} = \omega v_i^t + c_1 r_1 (pbest_i - x_i^t) + c_2 r_2 (gbest - x_i^t) + c_3 r_3 (\text{exemplar} - x_i^t) \quad (5)$$

Where c_3 is the elite learning factor. The inertia weight adopts a nonlinear decreasing strategy to balance global exploration and local exploitation:

$$\omega = 0.92 - 0.55 \cdot \left(\frac{t}{T}\right)^{1.8} \quad (6)$$

A V-shaped transfer function is used to convert continuous velocity into flip probability, and the position update rule is as follows:

$$x_{ij}^{t+1} = \begin{cases} 1 - x_{ij}^t, & \text{if } \text{rand} < |\tanh(v_i^{t+1})| \\ x_{ij}^t, & \text{otherwise} \end{cases} \quad (7)$$

In each generation, a uniform crossover operation is performed with probability 0.45 using the elite sample to further guide the search direction.

3.3. Adaptive Mutation Mechanism

To maintain population diversity and avoid premature convergence, an adaptive mutation strategy is designed. The mutation probability is dynamically adjusted according to particle quality and algorithm stagnation status:

$$p_{\text{mut}} = 0.01 + 0.02 \cdot I_{\text{worst}} + 0.02 \cdot I_{\text{stagnation}} \quad (8)$$

Where I_{worst} indicates whether the particle belongs to the

worst group (the bottom 25% in fitness ranking), and $I_{stagnation}$ indicates whether the algorithm has failed to improve for 4 consecutive generations. The mutation operation flips each variable gene bit with probability p_{mut} . This strategy maintains a low mutation probability in the early stage to protect excellent solutions, increases the mutation probability when the algorithm falls into stagnation to enhance exploration capability, and applies strong mutation to inferior particles to improve overall population quality.

3.4. Heuristic Local Search

To address the insufficient local search capability of the particle swarm algorithm in discrete space, a heuristic local search with length awareness and difficulty awareness is designed. This search is triggered every 5 generations or when the algorithm fails to improve for 4 consecutive generations, performing fine-grained optimization on the current optimal solution with a budget of 16 evaluations.

First, the difficulty deviation Δ_{diff} between the current path and the learner's cognitive level, as well as the deviation from the ideal length, are calculated. If $\Delta_{diff} > 0.1$, the path is too difficult, and high-difficulty nodes are prioritized for removal; if $\Delta_{diff} < -0.1$, the path is too easy, and nodes with moderate difficulty and good matching are prioritized for addition. The ideal length L_{ideal} is determined by the available learning time and cognitive level:

$$L_{ideal} = \text{clip}\left(\frac{t_{avail}/15 + (8 + (\text{cog} - 1) \cdot 6)}{2}, 6, 50\right) \quad (9)$$

Based on a node goodness evaluation function, the local search prioritizes removing "bad nodes" (those with large difficulty deviation, poor style matching, or irrelevance to the goal) and adding "good nodes" (those with good difficulty matching, style matching, and complete prerequisite relations), optimizing the combination through balanced swap operations.

4. Experimental Design and Results Analysis

4.1. Dataset and Experimental Setup

This study employs a self-constructed C language course

dataset for experimental validation. The dataset contains 197 knowledge points, including 64 easy, 65 medium, and 68 difficult knowledge points. There are 189 prerequisite relations covering the logical dependencies of the course knowledge points. The learner data is derived from 100 simulated learners, with cognitive levels ranging from 1 to 4, learning styles represented as four-dimensional vectors, and available learning time ranging from 60 to 300 minutes.

To comprehensively evaluate the performance of the IHPSO algorithm, the standard binary particle swarm optimization algorithm (BPSO) and the standard genetic algorithm (GA) are selected as baseline methods for comparison. The parameter settings for each algorithm are shown in Table 1. To ensure a fair comparison, all three algorithms use the same initial population and evaluation budget (3030 evaluations), and 30 independent experiments are conducted to eliminate the impact of randomness.

Table 1. Algorithm parameter settings

| Parameter | BPSO | GA | IHPSO |
|-----------------------------|----------|------|---------------|
| Population size | 30 | 30 | 30 |
| Maximum evaluations | 3030 | 3030 | 3030 |
| Inertia weight | 0.72 | - | 0.35-0.92 |
| Learning factors c1, c2, c3 | 1.8, 1.8 | - | 1.5, 1.7, 1.2 |
| Crossover probability | - | 0.85 | 0.45 |
| Mutation probability | - | 0.02 | Adaptive |
| Elite size | - | 2 | 8 |
| Local search budget | - | - | 16 |
| Local search period | - | - | 5 |

4.2. Convergence Performance and Path Quality Analysis

Fig. 2 shows the average convergence curves of the three algorithms on 30 learners. As can be seen from the figure, BPSO has the slowest convergence speed, with fitness values stabilizing only after approximately 2000 evaluations. GA achieves moderate convergence speed, stabilizing after about 1500 evaluations. IHPSO demonstrates significantly better convergence speed than the comparison algorithms, entering a stable state after approximately 600 evaluations.

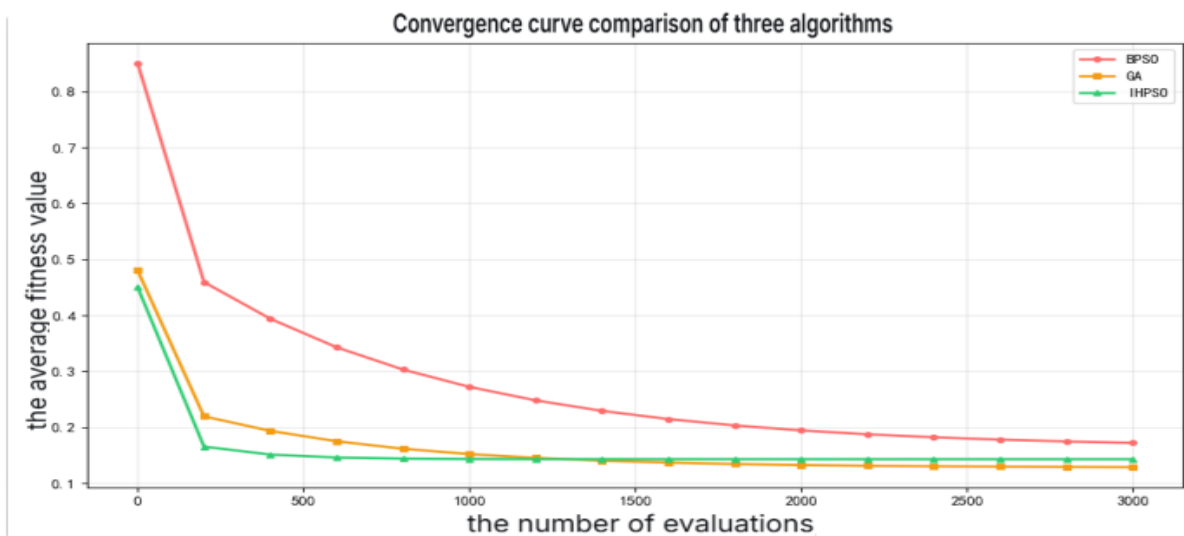


Figure 2. Convergence curve comparison of three algorithms

Table 2 presents the average fitness statistics of the three algorithms. IHPSO achieves an average fitness of 0.2039,

which is 62.46% lower than that of BPSO and 20.17% lower than that of GA. The average number of evaluations required

for IHPSO to reach 95% of its optimal performance is 683, compared to 1981 for GA and 2319 for BPSO, representing a convergence speed improvement of approximately 65% over

GA. Among the 30 independent runs, IHPSO outperforms GA in 28 cases, demonstrating the consistency of the improved algorithm's performance enhancement.

Table 2. Comparison of average fitness of three algorithms

| Algorithm | Average fitness | Standard deviation | Improvement over BPSO | Improvement over GA |
|-----------|-----------------|--------------------|-----------------------|---------------------|
| BPSO | 0.5433 | 0.1969 | - | - |
| GA | 0.2555 | 0.0603 | 52.98% | - |
| IHPSO | 0.2039 | 0.0669 | 62.46% | 20.17% |

4.3. Personalized Indicator Analysis

To further evaluate the personalized recommendation capability of the algorithm, three dimensions are analyzed: difficulty matching, time matching, and style matching. The results are shown in Table 3. IHPSO outperforms the

comparison algorithms on all three indicators. The difficulty matching error is 0.1833, which is 26.8% lower than that of GA. The time matching error is 0.1780, which is 53.1% lower than that of GA. The style matching score is 0.9103, which is 5.2% higher than that of GA.

Table 3. Comparison of personalized indicators

| Algorithm | Difficulty matching error | Time matching error | Style matching score | Algorithm |
|-----------|---------------------------|---------------------|----------------------|-----------|
| BPSO | 0.5416 | 1.5418 | 0.8471 | 0.1870 |
| GA | 0.2504 | 0.3798 | 0.8652 | 0.2041 |
| IHPSO | 0.1833 | 0.1780 | 0.9103 | 0.3873 |

The above results validate the effectiveness of IHPSO in personalized adaptation. The improvement in difficulty matching is attributed to the difficulty guidance mechanism in the heuristic local search. The enhancement in time matching is due to the length-aware strategy. The optimization of style matching stems from the retention of high-quality feature combinations through elite learning and crossover operations.

The significance test results indicate that the performance differences between IHPSO and the comparison algorithms are all statistically significant ($p < 0.001$), demonstrating that the advantages of the improved algorithm are not coincidental and are statistically reliable.

5. Conclusion

This study focuses on the problem of personalized learning path recommendation in online learning scenarios. To address the bottleneck of balancing knowledge logic constraints with learner feature adaptation, a personalized learning path recommendation method based on an improved particle swarm optimization algorithm (IHPSO) is proposed. The method achieves innovations in three aspects: constructing a course knowledge graph and generating a knowledge point sequence through topological sorting to provide a logical framework for path planning; establishing a multidimensional learner feature model and designing a comprehensive objective function to quantify the degree of adaptation; and making targeted improvements to the standard particle swarm optimization algorithm, including introducing an elite learning strategy to enhance convergence performance, designing an adaptive mutation mechanism to maintain population diversity, and proposing heuristic local search to achieve fine-grained path optimization.

Experimental results show that IHPSO achieves an average fitness of 0.2039, which is 62.46% lower than that of BPSO and 20.17% lower than that of GA, with a convergence speed improvement of 65% over GA. The algorithm significantly

outperforms the comparison algorithms in personalized indicators such as difficulty matching, time matching, and style matching ($p < 0.001$), validating the effectiveness and superiority of the proposed method.

This study still has certain limitations. The construction of the course knowledge graph relies on manual annotation, resulting in high construction costs. The learner data is generated through simulation, which differs from real learning scenarios. Future work will explore automated knowledge extraction methods to reduce manual annotation costs and conduct user experiments on real learning platforms to validate the application effectiveness of the method.

References

- [1] Feng W, Tang J, Liu T X. Understanding dropouts in MOOCs [A]. Proceedings of the AAAI Conference on Artificial Intelligence [C]. Hawaii: AAAI Press, 2019: 517-524.
- [2] Li A, Li Y, Gao X. Personalized Learning Path Recommendation Based on Knowledge Graphs: A Survey [J]. Electronics, 2026, 15(1): 238.
- [3] Jiang B, Li X, Yang S, et al. Data-driven personalized learning path planning based on cognitive diagnostic assessments in MOOCs [J]. Applied Sciences, 2022, 12(8): 3982.
- [4] Li H J, Zhang P W, Zhang Z, et al. Online learning path optimization method based on multidimensional information feature mapping model [J]. Control and Decision, 2019, 34(6): 1132-1140. (in Chinese)
- [5] Shi D, Wang T, Xing H, et al. A learning path recommendation model based on a multidimensional knowledge graph framework for e-learning [J]. Knowledge-Based Systems, 2020, 195: 105618.
- [6] Gao J Q, Liu Q H, Huang W B. Research on automatic generation of learning path based on knowledge graph [J]. Modern Educational Technology, 2021(7): 88-96. (in Chinese)
- [7] Abualigah L. Particle swarm optimization: advances, applications, and experimental insights [J]. Computers, Materials, & Continua, 2025, 82(2): 1539.