Learning Path Optimization based on Multi-Attribute Matching and Variable Length Continuous Representation

Yong-Wei Zhang (ywzhang@just.edu.cn)  
Jiangsu University of Science and Technology

Qin Xiao  
Jiangsu University of Science and Technology

Ying-Lei Song  
Jiangsu University of Science and Technology

Mi-Mi Chen  
Jiangsu University of Science and Technology

Research Article

Keywords: personalized learning path, multi-attribute matching, differential evolution, problem representation, combination optimization

Posted Date: October 9th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1400545/v2

License: Creative Commons Attribution 4.0 International License. Read Full License
Learning Path Optimization based on Multi-Attribute Matching and Variable Length Continuous Representation

Yong-Wei Zhang 1,*, Qin Xiao 2, and Ying-Lei Song 1 and Mi-Mi Chen 3

1 College of Automation, Jiangsu University of Science and Technology; (Y.Z.: ywzhang@just.edu.cn, Y.S.: jjsx_yxyl@just.edu.cn)
2 Information Construction and Management Office, Jiangsu University of Science and Technology; xiaoqinch@just.edu.cn
3 Principal’s Office, Jiangsu University of Science and Technology; icmm@just.edu.cn
* Correspondence: ywzhang@just.edu.cn;

Abstract: Personalized learning path considers both learner and resource attributes. The evolutionary algorithm approach usually forms the learning path generation problem into a problem that optimizes the matching degree of the learner and the generated learning path. The proposed work considers the learner attributes as ability level, learning objective, learning style, and expected learning time. The learning path attributes include difficulty level, covered concepts, supported learning styles, required learning time, and prerequisites. A variable-length representation of the learning path is adopted based on floating numbers, which significantly reduces the encoding length and simplifies the learning path generating process. An improved differential evolution algorithm is applied to optimize the matching degree of learning path and learner. The quantitative experiments on different problem scales show the proposed system outperforms the binary-based representation approaches in scaling ability and outperforms the comparative algorithms in efficiency.

Keywords: personalized learning path; multi-attribute matching; differential evolution; problem representation; combination optimization

1. Introduction

The large-scale application of online education gives more and more learners access to personalized E-learning resources. Many E-learning systems were proposed which provide unique learning experiences for each learner, such as [1–3], to name a few. Among these, the learning path personalization is an essential implementation towards the learner-oriented curriculum design [4].

The learning path personalization methods can be categorized into two classes [5]: Course Generation (CG) and Course Sequence (CS). The former generates a complete learning path for a learner, while the latter recursively generates a learning path based on the transactions of the learner during the learning process. Researchers applied various techniques to create personalized learning paths through the years, such as recommendation systems, decision support systems, evolutionary algorithm (EA), Data mining, artificial intelligence, etc.

The learning path planning problem essentially picks up a sequence of learning materials that satisfy the learner’s needs. The EA approaches usually consider the learning path planning problem as an optimization problem that minimizes a cost function described by the attributes of the learner and generated learning path. The learner attributes may include profile, ability, preferences, and goals, etc. The path attributes may include difficulty level, covered concepts, types, etc.

The learning path in the EA approaches usually represented by binary strings[6–8]. The issues of binary representation are two-fold: (1) binary representation is not capable
of representing the sequence of materials in a path; (2) binary representation suffers severe scaling issues, which usually lead to poor performance when the problem scales up. On the other hand, integer representation is capable of representing the material sequence [9,10], but the operators of the associated algorithm have to be carefully designed to deal with invalid solutions. Moreover, the algorithm-specific design of operators is challenging to implement into other algorithms, thus limiting the application.

Floating-point number representation, however, can represent any sequence by mapping the floating numbers to integers. The optimization problems associated with floating-number representation are generally referred to as continuous optimization problems. Usually, the algorithms for solving the continuous optimization problems have the same scaling issue with the increasing number of variables. However, for a learning path planning problem, the length of the learning path is limited by the learner’s energy. Therefore, the scaling problem may be avoided by carefully designed path representation.

In this research, a multi-attribute matching (MAM) model is proposed to describe the affinity between the learner attributes and the learning path attributes. The attributes regarding a learner include ability level, learning objective, learning style, and the expected learning time. The attributes regarding the learning path include difficulty level, covered concepts, supported learning style, required learning time, and the prerequisites. A set of affinity functions are proposed to represent several aspects of the MAM model. A variable-length continuous representation (VLCR) of a learning path is proposed to utilizes the powerful search ability of continuous evolution algorithms and reduce the search space. An adaptive differential evolution algorithm based on [11] is presented to optimize the MAM model and generate the learning path for a given learner.

The rest of the paper is organized as follows: Section 2 reviews the related works, Section 3 presents the formalized problem, Section 4 introduces the VLCR and the improved differential evolution algorithm, Section 5 evaluates the proposed system with problems of different scales, and Section 6 concludes the paper.

2 Related Works
2.1 Learning Path Personalization

The development of technology makes e-learning evolve to s-learning, i.e., smart learning [12]. The personalized learning is considered an essential part of the smart learning environment, which provides the learner customized learning plan that include exercises, documents, videos, and audios. Learning path personalization provides a structure of organizing these learning materials, which concerns not only the contents to learn but the sequence of learning them.

The sequence of learning materials is constrained by the relation between the materials. The relation can be captured by a knowledge model, such as knowledge map [13,14], knowledge graph [15,16], or concept map [17], etc. The model can either be built from the experts’ experience [18] or via educational data mining technology[19]. Many methods are explored to construct a learning path that satisfies the learner’s needs and the constraints of knowledge model. The representative methods include machine learning [17,20] and evolutionary algorithms [21,22]. In [17], a set of learning paths are generated through topological sorting algorithm and the long short-term memory neural network is trained to predict the learning effect of the learning path. In [23], a learning path recommendation model is developed based on a knowledge graph. The learning paths that fit the learning objectives are generated at once, and the one with the highest score is recommended. [24] gathered the learner creativity information via three creativity games, and the data mining technology of the decision tree is employed to generate a personalized learning path that optimizes the performance of creativity. Jugo [25] developed a web-based system that analyzes the learning path pattern, recommends the learning path, and visualizes the learning path pattern.
The machine learning approach highly depends on existing learning pattern that generated by topological algorithms or data mining technologies, which cannot make effective recommendation for a new system, and the relations between the learners and the learning materials are often ignored.

2.2 Evolutionary Algorithms for Learning Path Generation

The evolutionary algorithm approach on the other hand, considers the learning path generation as an optimization problem that seeks the optimal match between a learning path and the learner attributes. The most representative evolutionary algorithms include genetic algorithms (GA), ant colony optimization (ACO), particle swarm optimization (PSO), and differential evolution (DE) [7], [9] applied the variable-length genetic algorithm to deal with the variable length in the learning path. [10] applies an integer-encoded genetic algorithm to optimize the learning path of a Java course. The partially-mapped crossover and the cycle crossover operators create offspring from parent solutions and prevent infeasible solutions. A discrete version of the hill-climbing algorithm is also used to create new solutions, and a simulated annealing mechanism is applied to select the solutions in the next generation. The combination of tournament selection with random initialization and cycle crossover is reported to have the highest solution quality. [26] combines the GA and ACO to forge a collaborative optimization algorithm. The learning path is optimized based on the learning object, emotional state, cognitive ability, and learners’ performance. In [22], the learning path of MOOC is recommended based on a multi-objective optimization model. Several learner criteria are considered, such as number of enrollments, learning duration, and popularity. The model is solved by GA and ACO as well.

The learning path is essentially a sequence of learning materials; therefore, the learning path generation is a combinatorial optimization and NP-Hard problem [27]. However, the evolutionary algorithms that designed for such problem require highly specialized operators to maintain the feasibility of the new solutions, such as the crossover operator in GA, the mutation operator in DE, and the velocity update in PSO. The continues version of these evolutionary algorithm, however, guarantees the feasibility of a new solution given suitable solution representation scheme. We will discuss the solution representation in the Section 4.

2.3 Adaptive Differential Evolution

As a competitive evolutionary algorithm, DE has been applied to generate the learning path [28]. The original DE algorithm was proposed by [29] for continuous optimization problems. DE optimizes the vectors of a floating-point number using the difference between randomly selected pairs of vectors. Nevertheless, the parameter settings of DE partially depend on the problem. Therefore, research on the parameter settings of DE [30] and the adaptive parameters of DE [31] are proposed to tackle this problem.

In the research stream of adaptive DE, [32] proposed a self-adaptive DE, JADE, which generalizes the ‘DE/current-to-best’ mutation strategy to ‘DE/current-to-p-best’, and controls the parameter in a self-adaptive manner. [33] proposed Success-History-based Adaptive DE (SHADE). As an enhancement to JADE, SHADE utilized a history-based parameter adaptation scheme and ranked third in the real-parameter single objective optimization competition, CEC 2013. Later, [34] proposed the L-SHADE algorithm, extending the SHADE with Linear Population Size Reduction (LPSR). The LPSR of L-SHADE reduces the number of function evaluations in the exploitation stage of optimization and further enhances the performance. L-SHADE wins the competition in CEC 2014.

Two years later, [35] proposed the LSHADE-EpSin algorithm, which incorporated the Ensemble Sinusoidal Parameter Adaptation and became the joint winner in competition of CEC 2016. One year later, [11] proposed an improved algorithm, LSHADE-cnEpSin, to tackle the problems with high correlation between variables. LSHADE-cnEpSin became 2nd winner in the competition of CEC 2017.
Learning path generation is a complex combination problem that involves a large number of decision variables. When the problem scales up, the performance of many traditional algorithms may deteriorate drastically. Furthermore, the conventional combination optimization algorithms require the delicate design of operators to avoid infeasible solutions. To design a system with high scaling and generalization ability, the LSHADE-cnEpSin and floating-point number representation are adopted in this research to ensure the quality of generated learning path.

3 Problem Formulation

The problem of personalized learning path consists of four parts: learner attributes, resource attributes, decision variables, and cost function. The learner attributes describe the factors of the learner that are relevant to the learning path, while the resource attributes describe the characteristics of the learning material that are most relevant to the learner. Decision variables describe how the learning path is constructed from the learning resources, and the cost function represents the quality of the generated learning path.

3.1 Learner Attributes

A set \( L = \{L_1, L_2, \cdots, L_K\} \) that includes \( K \) learners is defined, where \( L_k = \{A_k, L_{Gk}, L_{Sg}, LT_k\} \) is the attribute set for the \( k \)-th learner. \( A_k \in [0,1] \) represents the learning ability of the \( k \)-th learner. The value is unified based on the difficulty level of the learning resources in the database. A learner who understands a material with a particular difficulty level is considered equal or greater ability than the resource.

\[ L_{Gk} = \{L_{G1k}, L_{G2k}, \cdots, L_{Gmk}\} \] represents the learner objects of the \( k \)-th learner. \( M \) is the number of total learning objects that are pre-determined by the nature of the learning resources. \( LT_{mk} = 1 \) denotes that the learner \( k \) has \( m \)-th learning object, \( LT_{mk} = 0 \) denotes not.

\[ LS_k = \{LS_{1k}, LS_{2k}, LS_{3k}, LS_{4k}\} \] represents the degree of inclination of the \( k \)-th learner towards four types of learning styles proposed by Felder and Silverman[36]: visual/verbal, sensing/intuitive, sequential/global, and active/reflective. The learning style is unified into \([0,1]\) for computational simplicity. For example, \( LS_{4k} \in [0,1] \) is a unified active/reflective learning style attribute, when \( LS_{4k} > 0.5 \) the learner inclines to reflective learning style, \( LS_{4k} < 0.5 \) the learner inclines to active learning style, and \( LS_{4k} = 0.5 \) represents neutral inclination.

\[ LT_k = \{LT_{lk}, LT_{uk}\} \] denotes the upper and lower bound of the expected course duration of learner \( k \), where \( LT_{lk} \) is the lower bound and \( LT_{uk} \) is the upper bound, and \( LE_{uk} \leq LE_{uk} \).

3.2 Resource Attributes

A set \( LM = \{LM_1, LM_2, \cdots, LM_N\} \) of \( N \) learning resources is defined, where \( LM_n = \{D_n, MC_n, MS_n, MT_n, MP_n\} \) is the attribute set of the \( n \)-th learning resource. \( D_n \in [0,1] \) denotes the difficulty level of the \( n \)-th learning resource. The difficulty level is closely related to the learner ability mentioned above, and can be ascertained based on the learner performance of the past.

\( MC_n = \{MC_{1n}, MC_{2n}, \cdots, MC_{Mn}\} \) denotes the covered concepts of the \( n \)-th learning resource. The total number of concepts is the same as the number of learning objects \( M \). \( MC_{mn} = 1 \) represents the resource \( n \) contains concept \( m \), while \( MC_{mn} = 0 \) means the opposite. A resource covers one concept at least and \( M \) concepts at most.

\( MS_n = \{MS_{1n}, MS_{2n}, MS_{3n}, MS_{4n}\} \) represents the matching level of resource \( n \) to the four learning styles. A visual resource, such as a picture or video, has a high match with the visual learning style, while the reading resource has a high matching with the verbal learning style. The intuitive learning style welcomes a material presented with symbols and equations, and a sensing learner prefers facts and data. Sequential learners learn materials step by step, while the global learner likes the big picture of the course. Active learners prefer interactive learning scenarios and reflective learners are usually theorists.
\(MT_n\) is the required time to learn material \(LM_n\), which can be ascertained by the learning history of the previous learners.

\(MP_n = \{MP_{1n}, MP_{2n}, \ldots, MP_{Mn}\}\) is the prerequisites of resource \(n\), where \(MP_{mn} = 1\) represents that resource \(n\) requires concept \(m\) as a prerequisite, while \(MP_{mn} = 0\) means the opposite. A resource can have 0-M prerequisite(s).

### 3.3 Decision Variables

For learner \(L_k\), \(NX\), the number of resources in the learning path, is determined by the following:

\[
NL = \frac{LT_{ik}}{\max_{n \in [1,\ldots,N]} MT_n} \leq NX \leq \frac{LT_{uk}}{\min_{n \in [1,\ldots,N]} MT_n} = NU,
\]

where \(NL\) and \(NU\) are the upper and lower bounds of the number of included resources, respectively. The decision variables are \(x = \{x_1, x_2, \ldots, x_{NX}\}\), where \(x_i \in [1,\ldots,N] \subset \mathbb{Z}\) represents the resource index to be chosen in the resource database. The last decision variable, \(NX\), represents the actual length of the learning path. Therefore, the actual learning path is \(P = \{x_1, x_2, \ldots, x_{NX}\}\).

### 3.4 Cost Function

Without loss of generality, the problem of recommending learning path to learner \(L_k\) is formalized as following optimization problem:

\[
P = \arg \min_x F(L_k, LM, x),
\]

where \(F\) is a mapping function which maps the attributes of learner \(k\), the attributes of each material, and the learning path to a scaler set: \(F: \{L_k, LM, x\} \rightarrow \mathbb{R}\). When the learning path optimization problem takes a minimization form, the mapping function \(F\) represents the degree of repulsion between the learner attributes and the learning path. The lower the mapped value is, the better the learner is matched to the learning path. To fully cover the relation between the learner attributes and the learning path, different matchings should be considered, which includes 1) learner ability and resource difficulty, 2) learning objective and the covered concepts of the selected resources, 3) learning style and resource type, 4) the sequence of the resources and their prerequisites, and 5) the expected learning time of the learner and the required learning time of the learning path. The matchings mentioned above can be formalized into five affinity functions \(F_1 \sim F_5\), whose relation with the main mapping function \(F\) is as follows:

\[
F = \sum_{i=1}^{5} w_i F_i
\]

where \(w_i\) is the weight of the affinity functions.

#### 3.4.1 Learner ability and resource difficulty

The affinity function \(F_1\) describes the averaged difference between the learner’s ability level and the materials’ difficulty level, as defined as follows.

\[
F_1 = \frac{\sum_{i=1}^{NX} |D_{x_i} - A_k|}{NX}
\]

#### 3.4.2 Learning object and covered concepts of resource

The affinity function \(F_2\) describes the average difference between the learning objectives and the covered concepts of the resources.

\[
F_2 = \frac{\sum_{m=1}^{M} \sum_{i=1}^{NX} |MC_{mx_i} - LC_{mk}| \cdot NM}{\sum_{m=1}^{M} LC_{mk}}
\]

where \(NM\) is the average number of materials for a learning objective and is defined as follows.
\[ NM = \sum_{m=1}^{M} \sum_{x=1}^{NX} MC_{mx_i}. \]

And \( NC \) is the number of covered concepts of the selected materials.

\[ NC = \sum_{m=1}^{M} \left( \sum_{i=1}^{NX} MC_{mx_i} > 0 \right). \]

The affinity \( F_2 \) punishes two cases: (1) the learning objectives are not covered by the selected materials, and (2) the materials cover more concepts than the objectives. Meanwhile, the affinity function \( F_2 \) considers the balance of the materials for each learning objective. If the covered concepts of the selected materials are the same as the learning objectives, and the number of materials is the same for each objective, \( F_2 = 0 \).

3.4.3 Learning style and material type

The affinity function \( F_3 \) describes the average difference between the learning style and the supported learning style of the materials.

\[ F_3 = \frac{\sum_{j=1}^{4} \sum_{i=1}^{NX} |MS_{xj_i} - LS_{jk}|}{NX} \]

3.4.4 Material and its prerequisites

The affinity function \( F_4 \) describes the average difference between the concept coverage of the learned materials before the current material and the prerequisites of the current material.

\[ F_4 = \frac{\sum_{m=1}^{M} \sum_{x=1}^{NX} (MP_{mx_i} > MC_{mx_i}^{sum})}{NX}, \]

where \( MC_{mx_i}^{sum} \) represents the coverage of concept \( m \) of the materials that precede the current material \( LM_{x,i} \) which is defined as follows.

\[ MC_{mx_i}^{sum} = \begin{cases} 1, & \sum_{j=1}^{i-1} MC_{mx_j} > 0; \\ 0, & \text{otherwise}. \end{cases} \]

If the previous materials cover the prerequisite \( m \) of the current material, \( MP_{mx_i} \) returns zero, otherwise, one. Suppose the prerequisites of all the materials are satisfied by the previous materials, \( F_4 = 0 \).

3.4.5 The Required Learning Time and Expected Learning Time

\[ F_5 = \max \left( LT_{lk} - \sum_{m=1}^{M} MT_{mx_i}, 0 \right) + \max \left( \sum_{m=1}^{M} MT_{mx_i} - LT_{uk}, 0 \right) \]

The affinity function \( F_5 \) describes the degree that the required learning time violates the constraints of the expected learning time. If the required learning time \( \sum_{m=1}^{M} MT_{mx_i} \) satisfies \( LT_{lk} \leq \sum_{m=1}^{M} MT_{mx_i} \leq LT_{uk} \), then \( F_5 = 0 \).

3.4.6 Weight Selection of the Sub Cost Functions

The five affinity functions will force the algorithm to obtain a learning path that best fits a given learner’s needs. In a sense, the affinity functions serve as the constraints of the learning path construction problem. However, a learning path that satisfies all the requirements may be challenging to find. In other words, it is difficult to find a learning path with a cost function value that equals zero. To help an algorithm finding a better solution, the above-mentioned constraints can be violated for some solutions. To distinguish the “bad” solutions from the good ones, proper weights must be assigned to each affinity function.

Affinity function \( F_1, F_3 \) and \( F_5 \) are the “soft” constraints in the problem. A smaller weight can be assigned to \( F_1 \) because a slightly easier or more difficult material is acceptable in most cases. The mismatch of learning style and material type is acceptable to some extent, then smaller weight on \( F_3 \) is suggested. Similarly, a slight violation of the expected learning time is also acceptable, then a small weight is suggested.

Affinity function \( F_2 \) and \( F_4 \) are the “hard” constraints. A learning path with over coverage of the learning objectives is acceptable, but the under coverage is not permitted.
Therefore, the weight on $F_2$ should be greater. If the learned materials have over coverage on the prerequisite of the current material, the learning process can continue. However, if under coverage happens on the prerequisite, the learner will confuse and the learning process is likely disrupted. Therefore, greater weight on $F_4$ is suggested.

### 4 Algorithm

In this research, a floating-point number representation with variable length is proposed to tackle the scaling problem of learning path generation. The decision variables are encoded into a floating-point number vector as follows:

$$ r = [r_1, \cdots, r_{NU}, r_{NU+1}], r_i \in [0,1], $$

where $NU$ is the maximum length of the learning path, i.e., the number of learning materials included in the learning path $P$. The encoded string is decoded into the decision variables $x = \{x_1, x_2, \cdots, x_{NU}, NX\}$ as follows:

$$ x_i = \begin{cases} \lfloor E(r_{NU+1}(NU - NL)) + NL \rfloor & i = NU + 1; \\ [r_iN] & \text{otherwise}. \end{cases} $$

where $E(\cdot)$ denotes the rounding up function. The length of the actual learning path is controlled by the last variable $NX$ in the decision vector $x$.

Any continuous optimization algorithm can manipulate the encoded vector $r$ without changing any operator. The length of $r$ is bounded by $NU + 1$, which is determined by the upper bound of the expected learning time. Since the expected learning time for a practical learner is usually limited and does not grow with the increasing number of learning materials in the system, a similar performance can be expected even if the problem scales up.

The LSHADE-cnEpSin starts with a standard initialization procedure:

$$ r_{i,j} = \text{rand}(0,1), 1 \leq i \leq NP, 1 \leq j \leq NU + 1, $$

where $i$ is the index of variables in a set of $NP$ individuals, and $j$ is the index of variables in that vector. $\text{rand}(0,1)$ is a uniformly distributed random number in the range of $[0,1]$. The search range of the algorithm is bounded by $[0,1]$ as well. Next, the 'DE/current-to-p-best' mutation strategy [37] is applied to generate mutant vectors in the $+1$ generation:

$$ v_{i}^{g+1} = r_i^g + F_i^g \cdot (p_{\text{best}}^g - r_i^g) + F_i^g \cdot (r_a^g - r_b^g), $$

where $r_a^g$ is chosen randomly from the current population and $r_b^g$ is chosen from the union of the current population and an external archive. The archive is a set of the inferior vectors recently replaced by trial vectors, and its member will be randomly replaced if its size exceeds $NP$. $F_i^g \in [0.5,1.0]$ is an adaptive scaling factor updated with generation number $g$ and a sinusoidal function. For more details, please refer to [11].

The LSHADE-cnEpSin algorithm also adopts the covariance matrix learning with the Euclidean neighborhood, which guided the search process in an Eigen coordinate system that may reflect the landscape information of the problem. A partial set of the current population is selected whose members are the vectors near the best vector of the population. The covariance matrix $C$ of the selected vectors is computed, to which the Eigen decomposition is applied as follows: [38]

$$ C = BD^2B^T, $$

where $B$ is an orthogonal matrix composed by Eigenvectors of $C$, and $D$ is a diagonal matrix composed of Eigenvalues. The parent vector $r_i^g$ and the mutant vector $v_{i}^{g+1}$ are rotated to the Eigen coordinate system as follows:

$$ \tilde{r}_i^g = B T_{r_i^g}, $$

$$ \tilde{v}_{i}^{g+1} = B T_{v_{i}^{g+1}}, $$

Next, a trial vector is generated by a crossover of the rotated mutant vector and the rotated parent vector:

$$ \tilde{u}_{i}^{g+1} = \begin{cases} v_{i}^{g+1} & \text{if } \text{rand}(0,1) \leq CR_i^g \text{ or } j = \text{rnbr}(i); \\ r_i^g & \text{if } \text{rand}(0,1) > CR_i^g \text{ or } j \neq \text{rnbr}(i). \end{cases} $$
The random index \( \text{rnbr}(i) \in \{1, \ldots, UN + 1\} \) ensures that at least one variable of \( U_i^{g+1} \) is inherited from \( V_i^{g+1} \), and \( CR^{g} \in [0,1] \) is the crossover probability drawn from a normal distribution that updates with the successful history of crossover. Next, the obtained trial vector is transformed back to the original coordinate system:

\[
U_i^{g+1} = BU_i^{g+1}, \quad (19)
\]

The trial vector is decoded into the decision vector \( x_i^{g+1} \) and evaluated by the cost function (4)–(11). The fitness of the decision vector is obtained by Equation (3). If the fitness is better than the parent’s, the parent vector \( r_i^{g} \) is replaced by the trial vector \( U_i^{g+1} \), otherwise, the parent vector is reserved.

A linear population reduction is also applied to control the population size and enhance search efficiency. The population size \( NP \) decreases from the original value to 4, which is the minimum number of vectors to perform the ‘DE/current-to-p-best’ mutation strategy.

5 Evaluation

A series of numerical experiments are designed to validate the effectiveness of the proposed model and the algorithms. Four algorithms are selected for comparative study. They are BPSO [6,39], PSO [40,41], DE [29,42], and LSHADE-cnEpSin [11]. The reasons for selecting these algorithms are: (1) BPSO is the binary version of the original Particle Swarm Optimization algorithm, which has proved effective over the Genetic Algorithm for e-course composition problem [6]; (2) PSO and DE are well-known optimizers for the continuous problem; (3) DE is the basic version of all the modified algorithms related to LSHADE-cnEpSin; (4) LSHADE-cnEpSin is the 2nd winner in the competition of CEC 2017.

All the algorithms are executed on 100 learners, respectively. For each learner, 30 runs are executed. Given that the testing algorithms use different population sizes, and the LSHADE-cnEpSin even adopts a varying population size, the number of generations as the indicator of optimization progress is inaccurate. Therefore, the number of function evaluations (FEs) will be used as the progress indicator and the terminal condition for all tested algorithms.

The convergence analysis will be presented in Section 5.1, the influence of the number of materials is discussed in Section 5.2, and the scalability of the proposed method is analyzed in Section 5.3. The experiment environments are as follows:

(1) Hardware environment: 3.4 GHz Intel Core i7 processor with 32 GB memories;
(2) Software environment: MATLAB R2018a.

5.1 Convergence Analysis

The parameter settings are depicted in Table 1. The termination condition is FEs=20000.

<table>
<thead>
<tr>
<th>Parameter Class</th>
<th>Parameter Symbols</th>
<th>Parameters Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters regarding learners</td>
<td>( A_k ), ( 1 \leq k \leq 100 )</td>
<td>The ability of the learner is randomly divided into five levels: ([0.2,0.4,0.6,0.8,1.0]).</td>
</tr>
<tr>
<td></td>
<td>( LC_k ), ( 1 \leq k \leq 100 )</td>
<td>The learning targets are randomly assigned to 100 learners, respectively.</td>
</tr>
<tr>
<td></td>
<td>( LS_k ), ( 1 \leq k \leq 100 )</td>
<td>The normalized learning styles are randomly assigned to 100 learners, respectively.</td>
</tr>
<tr>
<td></td>
<td>( LT_k ), ( 1 \leq k \leq 100 )</td>
<td>The expected learning time (including upper and lower bound) is randomly assigned to 100 learners.</td>
</tr>
<tr>
<td></td>
<td>( D_n ), ( 1 \leq n \leq 200 )</td>
<td>Five difficulty levels, i.e. ([0.2,0.4,0.6,0.8,1.0]), are randomly assigned to 200 materials, respectively.</td>
</tr>
</tbody>
</table>
Parameters regarding materials

- \(MC_n, 1 \leq n \leq 200\): The covered concepts are randomly assigned to 200 materials, respectively.
- \(MS_n, 1 \leq n \leq 200\): The supported learning styles are randomly assigned to 200 materials, respectively.
- \(MT_n, 1 \leq n \leq 200\): The required time is randomly assigned to 200 materials, respectively.
- \(MP_n, 1 \leq n \leq 200\): The prerequisites are randomly assigned to 200 materials, respectively.

Parameters regarding LSHADE-cnEpSin

- Initial population size is determined by the length of the encoded vector. The minimum population size is set to 4 to ensure the mutation is applicable. The population size is linearly reduced depending on the number of generations.
- \(4 \leq NP \leq 18(NU + 1)\): The scaling factor is adaptively updated according to [11].
- \(CR = 0.5026\): Crossover probability is adaptively updated according to [11].
- \(F = 0.6714\): The maximum number of function evaluations.
- \(FE_{\text{max}} = 20000\)

Parameters regarding BPSO

- \(NP = 50\): Population size is set according to [39]
- \(c1 = c2 = 2\): Learning factors are set according to [39]

Parameters regarding DE

- \(NP = 18\): Population size is set according to [42]
- \(CR = 0.5026\): Crossover probability is set according to [42]
- \(F = 0.6714\): The scaling factor is set according to [42]

Parameters regarding PSO

- \(NP = 30\): Population size is set according to [41]
- \(c1 = c2 = 2\): Learning factors are set according to [41]
- \(w_{\text{min}} = 0.4\)
- \(w_{\text{max}} = 0.9\): The upper and lower bounds of the inertia weight are set according to [41].

Figure 1 shows the convergence curves of the tested algorithms for the first learner. Each curve is averaged on 30 independent runs. The error bars of LSHADE-cnEpSin are also drawn in every 1000 FEs.

The convergence curves show that BPSO and PSO have similar convergence precision, whereas PSO converges faster in the initial stage. DE has the fastest convergence and outperformed BPSO and PSO in the average quality of solutions. LSHADE-cnEpSin showed a moderate convergence behavior in the early stage, while it continuously improves the solution throughout the entire search history and obtains the best average performance.
Table 2 shows the success rate, averaged mean, average minimum, and standard deviation of 30 runs over 100 learners. The algorithm that found the best solution (with minimum cost function value) is counted as a success for each learner. If multiple algorithms obtain the best solution, they are all marked as a success. The results show that LSHADE-cnEpSin has the highest chance (80 times out of 100 learners) of finding the best solution for a random learner. The mean, minimum, and standard deviation of 30 runs are computed and averaged over 100 learners. The results show that LSHADE-cnEpSin outperforms the rest of the algorithms in solution quality and search stability. Additionally, DE in continuous form outperforms the binary PSO as well.

Table 2. Averaged mean, minimum, and standard deviation among 100 learners

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Success Rate</th>
<th>Avg. Mean</th>
<th>Avg. Min</th>
<th>Avg. Std</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPSO</td>
<td>44%</td>
<td>0.57</td>
<td>0.470972</td>
<td>0.0930082</td>
</tr>
<tr>
<td>PSO</td>
<td>25%</td>
<td>0.7</td>
<td>0.489049</td>
<td>0.122281</td>
</tr>
<tr>
<td>DE</td>
<td>60%</td>
<td>0.55</td>
<td>0.457864</td>
<td>0.0816821</td>
</tr>
<tr>
<td>LSHADE-cnEpSin</td>
<td>80%</td>
<td>0.53</td>
<td>0.448074</td>
<td>0.0584142</td>
</tr>
</tbody>
</table>

Take the first learner as an example. BPSO, DE, and LSHADE-cnEpSin obtain the best learning path. The details of the obtained learning path are shown in Table 3. The learner’s ability level is 0.4, the three selected materials have difficulty levels of [0.6,0.4,0.4], which fit the learner’s ability level well. The learning objective consists of concepts No. 1, No. 5, No. 9, and No. 10, and the materials cover just enough concepts. The time required to complete the learning path is 5.34, which satisfies the learner’s expectation. The first material has no prerequisite, which is suitable to be the beginning of the path. The second material requires concept No. 9 as the prerequisite, which is covered by the first material. The third material requires the first concept, which is covered by the second material. No prerequisite is violated. The learning style of the learner is rescaled to 15 levels. The learning style indicators [2;0;-1;5] mean that the learner inclines to visual (with strength 2), balanced in sensing/intuitive, sightly global (with strength 1), and more active (with strength 5). The selected materials do not seem to fit the learning style well. However, given the complex constraints about the learning objectives and the prerequisites, this path is the best we can found among the 200 materials in the database. With the richness of the materials, the matching is expected to be better. The impact of the number of materials on the quality of constructed learning paths will be discussed next.

Table 3. The best learning path obtained by BPSO, DE, and LSHADE-cnEpSin

<table>
<thead>
<tr>
<th>Learner</th>
<th>Ability level</th>
<th>Learning Objectives</th>
<th>Learning Style</th>
<th>Expected Learning Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Material No. 1</td>
<td>0.6</td>
<td>[5;9]</td>
<td>[-6;4;-3;7]</td>
<td>1.92</td>
</tr>
<tr>
<td>Material No. 2</td>
<td>0.4</td>
<td>1</td>
<td>[0;5;2;6]</td>
<td>1.43</td>
</tr>
<tr>
<td>Material No. 3</td>
<td>0.4</td>
<td>10</td>
<td>[7;4;2;2]</td>
<td>1.99</td>
</tr>
</tbody>
</table>

Remark: Average difficulty level: 0.47, Total covered concepts: [1;5;9;10], Averaged supported learning style: [0.33, 1.67, -2.33, 0.33], Total required learning time: 5.34, No prerequisite violation

5.2 Influence of the number of materials
The number of materials may affect the performance of the binary algorithms since the search space grows with it. Whereas the increased richness of the materials may be beneficial for learning path planners as well. We tested the performance of BPSO, PSO, DE, and LSHADE-cnEpSin with a different number of materials. The tested numbers of materials are 100, 200, ..., 1000, 2000, ..., 5000, and 10000. For each number of materials, 30 runs are executed for each learner. The obtained final cost function values are averaged on 30 runs and then averaged on 100 learners. The results are shown in Figure 2.

Figure 2. Comparison of the average cost function values obtained by BPSO, PSO, DE, and LSHADE-cnEpSin with a different number of materials.

Figure 2 shows that with a limited number of materials (100 materials), all four algorithms obtain an averaged final fitness above 0.5. With such solution quality, one or more constraints may be violated. Especially for the learning style and material type, a mismatch is almost inevitable since the materials are not specifically designed for a single learner.

When the number of materials increases, all algorithms can improve their solutions. The turning point of the performance is 1000 materials. After this point, PSO and LSHADE-cnEpSin can improve the solutions further. The performance of DE seems less affected, whereas the performance of BPSO deteriorates with the number of materials.

The observation suggests that additional materials allow algorithms to explore more combinations of materials, hence increase the possibility of producing better solutions for a learner. The search space does not increase with the number of materials for continuous algorithms with varying length representation. Therefore, an improvement in the solutions is expected. For binary representation, the search space increases exponentially with the number of materials. Thus, the bottleneck of performance is observed when the problem scale reaches a certain point.
Figure 3 shows the boxplots of the final cost value of the tested algorithms. Each box shows the median (red line), percentiles, and the outliers (red cross). The results show that LSHADE-cnEpSin has better statistics of final solutions when the materials are rich.

Figure 4 is the convergence curves of the four algorithms with a different number of materials. LSHADE-cnEpSin shows an advantage when the number of materials exceeds 500, which becomes apparent when the materials exceed 1000.

5.3 Scalability Analysis

The average computation time per 1000 function evaluations is in Figure 5. The results show that the algorithms with vary length representation, i.e., PSO, DE, and LSHADE-cnEpSin, require constant computation time for a different number of materials. While the algorithms with binary representation, i.e., BPSO, consumed computation time that increased linearly with the number of materials. LSHADE-cnEpSin takes about 200 ms to complete 1000 function evaluations, almost twice the time compared to PSO and DE. The additional computation time of LSHADE-cnEpSin is from the computation of Euclidean distance for each pair of individuals in a population and the subsequent coordinate transformation. Trade a constant extra computation time for improved performance seems a reasonable investment for the learning path problems.
6. Conclusions

In this paper, a multi-attribute match (MAM) model is proposed for the learning path planning problem. The MAM model considers various aspects of both learners and learning materials. The attributes regarding a learner include the ability level, learning target, learning style, and expected learning time. The attributes regarding a learning material include difficulty level, covered concepts, supported learning style, required learning time, and prerequisite concepts. Five affinity functions are proposed to describe the...
relations between a learner and the learning path. The weighted sum of the affinity function is used as the cost function to be optimized by evolutionary algorithms.

A variable-length continuous representation (VLCR) of a learning path is proposed to utilizes the powerful search ability of continuous evolution algorithms and reduce the search space. The numerical experiments show that the algorithms adopting the VLCR have comparable performance with the PSO adopting binary representation when the number of materials is smaller. When the number of materials is more than 1000, the performance of binary PSO starts to deteriorate. Eventually, BPSO fails to provide a valid solution if the number of materials exceeds 5000. In contrast, the number of materials has limited influence on producing high-quality learning paths for the algorithms adopting the VLCR. The results show the high scalability of VLCR when it is combined with continuous evolutionary algorithms.

Further, the constant computation time of VLCR shows the potential of dealing with large-scale learning path planning problems, which might be essential for a real-world learning management system that contains thousands of learning materials to generate a valid learning path before the learner loses patience.

It is expected that MAM and VLCR will make an impact on real-world learning path planning problems. Future work may include a dynamic learning path update using the interaction information of a learner or a collaborative filtering mechanism to improve the quality of the initial learning path.

**Author Contributions:** Conceptualization, Y.Z.; methodology, Q.X.; software, Y.Z.; validation, Y.Z. and Y.S.; formal analysis, Y.Z.; investigation, M.C.; resources, M.C.; data curation, Y.Z.; writing—original draft preparation, Y.Z.; writing—review and editing, Q.X.; visualization, Y.Z.; supervision, Y.S.; project administration, M.C.; funding acquisition, M.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by (1) Philosophy and Social Science Research Project for the Universities of Jiangsu Province "Research on the Personalized Design Strategy of Ubiquitous Learning Resources Based on the Learner Model", 2019, Grant number: 2019SJA1912; (2) Modern Education Technology Research Project of Jiangsu Province "Reform Exploration and Practice of Three-Stage Continuous Improvement Teaching Method Relying on the Online Teaching Platform", 2021, Grant number: 2021-R-93695; (3) Construction Project of Postgraduate Online Courses of Jiangsu University of Science and Technology, 2020; The APC was funded by Higher Education Project of Jiangsu University of Science and Technology "Research on Informatization Innovation Strategies Based on the Ecosystem of Adult Higher Education", 2020, Grant number: GJK-TYBJX202004.

**Data Availability Statement:** The data presented in this study are openly available in Science Data Bank at 10.11922/sciedb.01552, https://www.scidb.cn/en/s/e2meia.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.
References


