Constructing a personalized e-learning system based on genetic algorithm and case-based reasoning approach

Mu-Jung Huang a, Hwa-Shan Huang a, Mu-Yen Chen b,*

a Department of Information Management, National Changhua University of Education, Changhua 50058, Taiwan, ROC
b Department of Accounting, National Changhua University of Education, #2 Shi-da Road, Changhua 50058, Taiwan, ROC

Abstract

The Internet and the World Wide Web in particular provide a unique platform to connect learners with educational resources. Educational material in hypermedia form in a Web-based educational system makes learning a task-driven process. It motivates learners to explore alternative navigational paths through the domain knowledge and from different resources around the globe. Consequently, many researchers have focused on developing e-learning systems with personalized learning mechanisms to assist on-line Web-based learning and to adaptively provide learning paths. However, although most personalized systems consider learner preferences, interests and browsing behaviors when providing personalized curriculum sequencing services, these systems usually neglect to consider whether learner ability and the difficulty level of the recommended curriculums are matched to each other.

Therefore, our proposed approach is based on the evolvement technique through computerized adaptive testing (CAT). Then the genetic algorithm (GA) and case-based reasoning (CBR) are employed to construct an optimal learning path for each learner. This paper makes three critical contributions: (1) it presents a genetic-based curriculum sequencing approach that will generate a personalized curriculum sequencing; (2) it illustrates the case-based reasoning to develop a summative examination or assessment analysis; and (3) it uses empirical research to indicate that the proposed approach can generate the appropriate course materials for learners, based on individual learner requirements, to help them to learn more effectively in a Web-based environment.

Keywords: Computer-assisted testing; Case-based reasoning; Genetic algorithm

1. Introduction

The Internet and the World Wide Web in particular provide a unique platform to connect learners with educational resources. Educational material in hypermedia form in a Web-based educational system makes learning a task-driven process. It motivates learners to explore alternative navigational paths through the domain knowledge and from different resources around the globe. However, the structure of the presented domain and the content are usually presented in the same way, without taking into account the learners’ goals for browsing, their experience, their existing knowledge, etc. This is an issue that needs further attention, especially when it comes to Web-based instruction, where the learners’ population is usually characterized by considerable heterogeneity with respect to background knowledge, age, experiences, cultural backgrounds, professions, motivation, and goals, and where learners take the main responsibility for their own learning.

Curriculum sequencing is a well-established technology in the field of intelligent tutoring system (ITS). The idea of curriculum sequencing is to generate an individualized course for each student by dynamically selecting the most optimal teaching operation (presentation, example, question, or problem) at any given moment. By optimal teaching operation we mean an operation that in the context of other available operations brings the student closest to the...
ultimate learning goal. Most often the goal is to learn a required set of knowledge up to a specific level in a minimal amount of time. However, it is easy to imagine other learning goals, such as minimizing student error rates in problem solving.

Various approaches to sequencing have been explored in numerous ITS projects. The majority of existing ITSs can sequence only one kind of teaching operation. For example, a number of sequencing systems including the oldest sequencing systems (Barr, Beard, & Atkinson, 1976; Brusilovsky, 1993) and some others (Brusilovsky, 1993; Brusilovsky & Vassileva, 2003; Rios, Millán, Trela, Pérez, & Conejo, 1999) can only manipulate the order of problems or questions, an approach usually called task sequencing. A number of systems can do sequencing of lessons – reasonably big chunks of educational material complete with presentation and assessment (Brusilovsky, 1994; Capell & Dannenberg, 1993). The most advanced systems are able to sequence several kinds of teaching operations such as presentation, examples, and assessments (Brusilovsky, 1992; Chen, Lee, & Chen, 2005; Chen, Chang, Liu, & Chiu, 2005; Khuwaja, Desmarais, & Cheng, 1996).

One could say that sequencing is an excellent technology for distance education. Indeed, sequencing is presently the most popular technology in Web-based ITS (Papanikolaou & Grigoriadou, 2002). Therefore, the proposed approach is based on a pre-test to collect incorrect learning concepts of learners through the computerized adaptive testing (CAT) (Hsu & Sadock, 1985). Afterwards the genetic algorithm and case-based reasoning are employed to construct a near-optimal learning path according to these incorrect response patterns of the pre-test.

The rest of this paper is organized as follows: Section 2 describes the related literatures review, and we present the research methodology in Section 3. This is followed by a description of the proposed system in Section 4, while the evaluation of the system is reported in Section 5. Finally, we draw our conclusion in Section 6.

2. Literature review

2.1. Mastery learning

Mastery learning is a theoretical perspective of education that has attracted much attention in the past. The article by Bloom (1968) on mastery learning is widely regarded as the classic theoretical perspective on this model of education (Bloom, 1968). Bloom’s article compares two models of education: the traditional model and the mastery model. The traditional model uses the same instruction for an entire class, regardless of aptitude. The instructor presents the required information to the students who are then tested to measure the information they have retained. Students are typically given only one chance to learn the material. The course then moves on to additional material. Once tested, students may learn what mistakes they made, but they are never retested to learn whether they have learned from those mistakes. Consequently, the amount of learning in a classroom varies between students. Students with an aptitude to learn the requisite material quickly move forward while slower students fall behind and receive lower grades. In contrast, the mastery model varies instruction according to aptitude, resulting in higher levels of learning for all students. If students have not learned the material by the first test, they can repeat it until they achieve the required level of competence. Then they move on to the next module. As a result, teachers employing a mastery learning model of education should hypothetically find high levels of achievement among all students.

Mastery learning has been widely applied in levels from primary education (Crijnen, Freehan, & Kellam, 1998), and a variety of subject matter from nursing (VanArsdale & Hammons, 1998) to economics (Laney, 1999), and for skills ranging from reading (Crijnen et al., 1998) to critical thinking (Mevarech & Susak, 1993). Many meta-analytic studies have demonstrated consistent positive effects for mastery learning programs (Guskey & Gates, 1985; Guskey & Pigott, 1988). According to these researches, they believe that 80 percent or more of the students in any given class can attain the same high level of achievement that only 20 percent attain under more traditional instructional methods.

The major steps in implementing mastery learning are outlined in Fig. 1. First, teachers must review their curriculum and instructional materials to decide what concepts or ideas are most important for students to learn and at what level. The next task is planning a formative assessment which is basically a diagnostic instrument or process used by the teacher. It is also a principal aid in the planning of corrective measures to remedy learning errors. Third, activities to correct and enrich may take a variety of forms and usually vary from one unit to the next. For instance, activities to correct may involve alternative materials or resources, peer or cross-age tutoring, computer-assisted lessons, or any type of learning activity that allows for a difference in sensory or motivational preference. Enrichment activities may also include tutoring special projects, problem-solving exercises, or any learning activity that is both stimulating and rewarding for fast learners. In the fourth step, this second assessment is parallel in form to the first formative assessment for the unit, but it is usually not identical; that is, it covers the same concepts and material as the first assessment but asks questions in a slightly different way or format. If the correctives have been successful in helping students remedy their learning difficulties, then almost all students will demonstrate their mastery in the second formative assessment.

Finally, this second formative assessment also becomes a powerful motivational device by showing students directly that they can improve their learning and become successful learners. Therefore, students can move to the next unit of instruction. Finally, there is the development of a summative examination or assessment. This examina-
tion is a culminating demonstration of what students have learned. Typically, it is administered after several units of instruction.

2.2. Genetic algorithm

Many machine learning approaches have been applied by learning module developers, including inductive logic programming, decision tree (e.g., ID3 and C4.5), deductive reasoning (e.g., expert system), fuzzy logic, neural networks, and genetic algorithm (GA). The including inductive logic programming methods are based on predicate calculus and they performed very well (Kovalerchuk & Vityaev, 2000), but they are hard to understand and implement. The decision tree methods are based on supervised learning and apply the information gain theory to determine the attributes to construct a minimal-attributes decision tree. However, they cannot adequately handle the exceptional situations (Giarratano & Riley, 1998). Traditional expert systems can obtain a very good trading performance when based on high quality knowledge. However, this is mostly very difficult to be acquired by human experts. Fuzzy logic is used to clarify ambiguous situations, but it is difficult to design a reasonable membership function (McIvor, McCloskey, Humphreys, & Maguire, 2004). Neural networks have been used in financial problems for a long time (Wang, 2003). Unfortunately, the learned environment patterns of neural networks are like being embedded in a black box, and are hard to explain.

Due to the huge search space, the optimization problem is often very difficult to solve. A genetic algorithm is usually used as an optimization technique to search the global optimum of a function. However, this is not the only possible use for GA. Other applications where robustness and global optimization are needed can also benefit greatly from the use of the GA. The GA performs the optimization process in four stages: initialization, selection, crossover, and mutation (Davis, 1991).

2.2.1. Initialization stage

The search space of all possible solutions is mapped onto a set of finite strings. Each string (called chromosomes) has a corresponding point in the search space. The algorithm starts with the initial solutions that are selected from a set of configurations in the search space called population using randomly generated solutions or by applying special algorithms. Each of the initial solutions (called an initial population) is evaluated using a user-defined fitness function. A fitness function exists to numerically encode the performance of the chromosome.

2.2.2. Selection stage

A set of individuals that have high scores in the fitness function is selected to reproduce itself. Such a selective process results in the best-performing chromosomes in the population to occupy an increasingly larger proportion of the population over time. From the selected set of individuals, some progeny is generated by applying different genetic operators (i.e., crossover, mutation).

2.2.3. Crossover stage

Crossover operates by swapping corresponding segments of a string representation of a couple of chromosomes (called “parents”) to produce two chromosomes (called “children”). The simplest one is one-point crossover: a cutting site is chosen at random and the children are obtained by each taking the first part from one parent and the second part from the other parent.

2.2.4. Mutation stage

Mutation operates on a single chromosome: one element is chosen at random from the chain of symbols, and the bit string representation is changed with another one.

The GA has been applied to various kinds of domains such as autonomous robotics (Colombetti & Dorigo, 1994), knowledge discovery (Lanzi, Stolzmann, & Wilson, 2000), power system (El-Zonkoly, 2005), bankruptcy predictions (Back, Laitinen, & Sere, 1996; Shin & Lee, 2002), and computational economics (Chen & Chen, 2006). However, relatively little research of GA focuses on the computer-assisted learning (CAL) domain. Since the GA is dynamic, evolutionary, and rule-based, it can fit the fast-changing educational learning and behavior domain. Therefore, this research applies the GA to analyze learners’ behavior and performance.
2.3. Case-based reasoning

Case-based reasoning (CBR) can mean adapting old solutions to meet new demands, using old cases to explain new situations, or reasoning from precedents to interpret a new situation. The planner of CBR must be a learning system because it must reuse its own experiences. CBR requires a knowledge-based learning that makes the planner understand what should be learned and when it should be learned. This learning breaks down into three types: learning plans, learning expectations, and learning criticisms.

In general, the reasoning procedure of the CBR system is as shown in Fig. 2. There are five steps in case-based reasoning: (1) Assigning indexes. The indexes are critical features that characterize a case and determine how cases are stored in the case library. The major purpose of indexing is to allow a CBR system to retrieve one or more cases that are similar to the new problem. (2) Case retrieval. The goal of this step is to retrieve old cases stored in the case library. (3) Case adaptation. In case-based problem solving, the old solution case is used as an inspiration for solving new problems. Because a new case may not exactly match the old one, the old knowledge may often need to be fixed to fit the new one. (4) Case testing. The proposed solution will be tried out to see whether it really solves the new problem. (5) Case storage. Once the new problem is solved, it is stored in the knowledge base for future use.

In recent years, CBR has been used in a number of applications in supporting medical diagnoses (Gierl & Stengel-Rutkowski, 1994; Schmidt, Pollwein, & Gierl, 1999), sales forecasting (Chang & Lai, 2005), and bankruptcy predictions (Elhadi, 2000). However, relatively little research of CBR is focused on the e-learning domain. Basically CBR is a machine learning method that adapts previous similar cases to solve current problems. CBR shows significant promise for improving the effectiveness of complex and unstructured decision making. Therefore, it can be an appropriate approach to aid curriculum design in dealing with the e-learning system environment.

3. Methodology

This section describes the system architecture and personalized curriculum sequencing approach using the proposed mastery learning theory. First, an overview of the system architecture is presented in Sections 3.1.3.2 and 3.3 then describe the pre-test to calculate the curriculum difficulty parameters and the curriculum relation degree. After these two processes we will apply the GA and CBR to construct an appropriate learning path and carry out a summative assessment analysis.

3.1. System architecture

The personalized e-learning system based on mastery learning (PLS-ML) has an architecture that is designed as shown in Fig. 3. It is an implemented version of the general framework of the genetic-based and case-based reasoning system. When learners have studied unit 1 of the instruction, then they will first undergo their first formative assessment, and the system will calculate their scores and analyze their learning situation. As a result, if they fail to reach ‘mastery’ level in unit 1, the system will recommend an appropriate personalized curriculum sequencing suggestion based on GA. On the other hand, if they reach mastery level in unit 1 then they will continue with the additional interesting topics or extension materials in enrichment activities.

Therefore, learners can re-learn the same concepts through different curriculum sequencing and materials in corrective activities. When learners complete their corrective activities, they are administered a second formative assessment. It covers the same concepts and learning goals but is not composed of exactly the same problems or questions. This ensures that learners learn the important concepts rather than simply memorize the answers to specific questions. Besides, each individual curriculum sequencing and each formative assessment result will be stored in CBR. Afterward, if learners can pass the second formative assessment, then they will move into the next unit of instruction. Finally, CBR can provide a summative assessment for each individual learner after several units of instruction. Our goal is to generate appropriate curriculum materials to learners based on individual learner requirements, and help them to learn more effectively in a Web-based environment.

3.2. Estimation of curriculum difficulty parameters

The curriculum modeling process presents a detailed curriculum design procedure to establish the difficulty
parameters of the curriculum and the curriculum contents for personalized curriculum generation. This approach based on Chen, Lee, et al. (2005) concepts (Chen, Chang, et al., 2005) and uses a statistics-based method derived from CAT theory through a painstaking test process to determine the difficulty parameters. The detailed flowchart of the curriculum modeling process is illustrated in Fig. 4.

As an example, to design a course of ‘Java Language Programming’, several experienced teachers were invited as curriculum experts to analyze the primary concepts for this course in the curriculum modeling process. The curriculum experts designed the corresponding tests item for each learning concept. That is, the test items were regarded as key characteristics of the corresponding learning content. In addition, about 300 examinees who majored in the course of ‘Java Language Programming’ were asked to join the pre-test, which contained 18 topics covering those learning concepts. Their test data was analyzed according to the item response theory in CAT (Baker, 1992) using the statistics-based BILOG program to obtain the appropriate difficulty parameters for these test items. Afterwards the Web page of the curriculum was designed following the conveying content of the corresponding test item. Since the content of the curriculum was derived from the concept of the test item, it is assumed that the difficulty of the curriculum equals the difficulty of the corresponding test item.

3.3. Estimation of the curriculum relation degree

In this research, we use the vector space model (VSM) to calculate the relation degree between curriculums. In the VSM each curriculum is represented as a vector, and their relevance to the queries submitted by the user is measured through appropriate matching functions. The information retrieval process has one major component: the extraction of the term by the document matrix, and it is performed once for any given database.

The term extracted by curriculum matrix $A$ is obtained through several steps: pre-processing, normalization, and indexing. Pre-processing removes all elements supposed to be useless in a retrieval process, and normalization removes the variability that is not useful to the retrieval process. After pre-processing and normalization, the original contents of the curriculums are converted into streams of terms. The result is the term by curriculum matrix $A$ where each column $j$ corresponds to a document, and each row $i$ corresponds to a term of the dictionary. The generic element $A_{ij}$ can be written as follows:

$$A_{ij} = L(i, j) \cdot G(i)$$

where $G(i)$ is a global weight taking into account information extracted from the whole database, and $L(i, j)$ is a local weight based on information from the only document $j$. An extensive survey about weighting schemes has been pro-
vided by Salton and Buckley (1988). In this work we apply the tf·idf scheme:

\[
A_{ij} = tf(i,j) \cdot \log \left( \frac{N}{N_i} \right) = tf(i,j) \cdot \text{idf}(2)
\]

where the local weight \(tf(i,j)\) is the term frequency of term \(i\) in curriculum \(j\) (i.e. the number of times term \(i\) appears in curriculum \(j\)). \(N\) is the total number of curriculums in the database, and \(N_i\) is the number of curriculums containing term \(i\). The logarithm is called the inverse document frequency (idf) and it has a higher value for terms appearing in fewer texts. The tf·idf is the most applied weighting scheme (Baeza-Yates & Ribeiro-Neto, 1999) and it embodies the intuition that the more a term appears in a curriculum, the more it is representative of its content, and that terms appearing in few curriculums allow better discrimination between different texts.

Assume that there are a total terms \(n\) under union of all linguistic terms of the \(i\) curriculum and the \(j\) curriculum. The concept relation degree for the \(i\) and \(j\) curriculum can be calculated using the cosine-measure, and listed as follows:

\[
r_{ij} = \text{Cosine Coefficient} = \frac{\sum_{k=1}^{n} A_{ik} A_{jk}}{\sqrt{\sum_{k=1}^{n} (A_{ik})^2} \sqrt{\sum_{k=1}^{n} (A_{jk})^2}}
\]

where \(C_i = \{A_{i1}, A_{i2}, \ldots, A_{in}\}\) and \(C_j = \{A_{j1}, A_{j2}, \ldots, A_{jn}\}\), respectively, representing the vectors in a multidimensional Euclidean space for the \(i\) and \(j\) curriculum, \(r_{ij}\) denotes the concept relation degree between the \(i\) and \(j\) curriculum.

4. System design and development

Fig. 3 shows the architecture of the PLS-ML system from the viewpoint of its functionalities. PLS-ML consists of two main modules, the GA-based module and the CBR-based module. The GA-based module is composed of a generation engine, and an XML-based knowledge description, while the CBR-based module consists of a knowledge base. We will discuss the detailed functions in the following sections.

4.1. User interface

Most of the Web-based systems seem to lack friendly end-to-end supporting tools for development and maintenance. Therefore, PLS-ML was developed using compound Web application techniques to represent a Web-based user interface, such as HTML (Hypertext Markup Language) and JSP (Java Server Page). On the other hand, several art design software such as Photoshop and Dreamweaver were used for their abundant functionalities. Different interfaces were designed for course content, exam, personal curriculum sequence recommendations, and a discussion board. Screenshots for course content and exam are shown in Figs. 5 and 6.

4.2. GA-based module

4.2.1. GA for personalized curriculum generation

This section explains how to generate the personalized learning path for Web-based instruction, utilizing the genetic algorithm.
4.2.1. Definition of chromosome strings. In this study, a serial number is assigned to each curriculum from 1 to \( n \) if there are a total of \( n \) curriculums in the curriculum database for the learning path generation. Thus, the assigned serial number of each curriculum is combined directly with the serial number of the successive curriculum as strings to represent the generated learning path for the genetic algorithm. The whole individual, represented by the chromosomes of all curriculum parameters for the genetic algorithm, is illustrated as Fig. 7.

4.2.1.2. Initial population size. Generally, the initial population size can be determined according to the complexity of the solved problem. A larger population size will reduce the search speed of the GA, but it will increase the probability of finding a high quality solution. To construct a high quality learning path for an individual learner, the initial population size in this research is chosen as 50 for the generation of a personalized curriculum.

4.2.1.3. Selecting the fitness function. The fitness function is a performance index that it is applied to judge the quality of the generated learning path for the GA. In order to generate a personalized learning path for an individual learner based on the pre-test results, the difficult parameters of the curriculum and the concept relation degrees of the curriculum must be considered simultaneously to determine the fitness function. In our method, the learning path constructed by the GA only considers the curriculum for which the learner gives incorrect pre-test results. Moreover, the curriculum with the smallest difficulty parameter is selected as the first curriculum ranked in the constructed learning path. Therefore, the fitness function is formulated as follows:

\[
f = \sum_{i=2}^{n} (w \times r_{(i-1)} + (1 - w) \times d_i)
\]

where \( f \) is the fitness function for the GA, \( r_{(i-1)} \) represents the concept relation degree of the \((i - 1)\) curriculum with the \( i \) curriculum in the constructed learning path, \( d_i \) is the difficulty parameter of the curriculum, \( w \) is a degree of learning, and \( n \) stands for the total number of curriculums considered for personalized curriculum generation.

4.2.1.4. Reproduction operation. In the reproduction operation, the chromosome with the larger fitness function value

Fig. 6. A screen shot for the user interface on exam.

Fig. 7. The individual strings combined with serial number of the curriculum for the GA (Chen, Chang, et al., 2005).
will have a higher probability to reproduce the next generation. The aim of this operation is to choose a good chromosome to achieve the goal of gene evolution. The most commonly used method of reproduction is the weighted roulette selection (Franz, 2002). In this study, we will use this method to perform the reproduction operation.

4.2.1.5. Crossover operation. In the crossover operation, the two randomly selected serial numbers of the chromosomes in two individuals exchange the entire chromosome by probability decision. This operation aims to combine two parent chromosomes to generate better child chromosomes. In our study, the uniform crossover operation is used. However, in order to avoid that the generated learning path has a duplicate serial number of chromosomes or that the serial number of the curriculum is over the total number of curriculums after the crossover operation is performed, the crossover operation will exchange the whole chromosome by probability decision. Fig. 8 illustrates the used crossover operation. With other words, the performed crossover operation can avoid the generation of an illegal learning path. In this research, the probability of mutation is set to be 1.

4.2.1.6. Mutation operation. In the mutation operation, the randomly selected two chromosomes in an individual are forced to exchange the position of chromosome. The mutation operation will create some new individuals that might not be produced by the reproduction and crossover operations. Fig. 9 illustrates the crossover operation that is used. Generally, a lower probability of mutation can guarantee the convergence of the GA, but it may lead to a poor solution quality. On the other hand, a higher probability of mutation may lead to the phenomenon of a random walk for the GA. In this research, the probability of mutation is set to be 0.1.

4.2.1.7. Stop criterion. The genetic algorithm repeatedly runs the reproduction, crossover, mutation, and replacement operations until it meets the stop criterion. The stop criterion is set to be 100 generations, because this criterion can obtain satisfied learning paths for the individual learner.

4.2.2. Experiments

The curriculum modeling process mentioned in Section 3.2 is used to determine the difficulty parameter of each piece of the curriculum. In other words, the curriculum organized on a single Web page is the smallest course element in the proposed personalized curriculum generation approach. In our experiments, the course unit, "Control Structure", in the course category, "JAVA Language Programming", is used to generate the personalized learning path, which includes many curriculums with various levels

![Crossover operation](Fig. 8. Crossover operation (Chen, Chang, et al., 2005)).

![Mutation operation](Fig. 9. Mutation operation (Chen, Chang, et al., 2005)).
of difficulty to convey the concept of the “Structure Programming”. Assume that a pre-test in the course unit “Control Structure” is performed by a learner, and that a total of 10 incorrect testing items occurred. Table 1 lists the titles of the corresponding curriculum and their difficult parameters to which the learner gives incorrect responses.

Based on the difficult parameters and the curriculum relation degree to which the learner gave incorrect test item responses, the GA-based module is employed to construct a personalized learning path according to the designed fitness function. Table 2 illustrates the generated learning path proposed by the GA. We find that the generated learning path recommends an appropriate learning path to the learner while simultaneously considering the difficult parameters of the curriculum and the relation degree of the curriculum. This can be very helpful for the learner because it can guide him/her to achieve his/her study goals more effectively. This result demonstrates that the proposed GA-based module can indeed generate a high quality learning path for personalized learning.

Table 1
The corresponding difficulty parameter for each curriculum in the “Control Structure” unit

<table>
<thead>
<tr>
<th>Curriculum</th>
<th>Title of curriculum</th>
<th>Difficulty</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Introduction of syntax of “do while” instruction</td>
<td>0.5</td>
</tr>
<tr>
<td>C2</td>
<td>Introduction of syntax of “break” instruction</td>
<td>0.7</td>
</tr>
<tr>
<td>C3</td>
<td>Example 2 of “do while” instruction</td>
<td>1.6</td>
</tr>
<tr>
<td>C4</td>
<td>Using opportunity of repetition programming</td>
<td>1.6</td>
</tr>
<tr>
<td>C5</td>
<td>Introduction of syntax of “for” instruction</td>
<td>-0.5</td>
</tr>
<tr>
<td>C6</td>
<td>Introduction of syntax of “continue” instruction</td>
<td>0.3</td>
</tr>
<tr>
<td>C7</td>
<td>Introduction of syntax of “switch” instruction</td>
<td>0.8</td>
</tr>
<tr>
<td>C8</td>
<td>Example 1 of “for” instruction</td>
<td>1.8</td>
</tr>
<tr>
<td>C9</td>
<td>Example 1 of “do while” instruction</td>
<td>1.2</td>
</tr>
<tr>
<td>C10</td>
<td>Example 1 of “switch” instruction</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Table 2
The generated learning path by GA

<table>
<thead>
<tr>
<th>Learning path</th>
<th>Difficulty</th>
<th>CRD</th>
</tr>
</thead>
<tbody>
<tr>
<td>C5</td>
<td>-0.5</td>
<td>-</td>
</tr>
<tr>
<td>C8</td>
<td>1.8</td>
<td>0.9308</td>
</tr>
<tr>
<td>C7</td>
<td>0.8</td>
<td>0.3625</td>
</tr>
<tr>
<td>C10</td>
<td>0.6</td>
<td>0.7821</td>
</tr>
<tr>
<td>C4</td>
<td>1.6</td>
<td>0.287</td>
</tr>
<tr>
<td>C1</td>
<td>0.5</td>
<td>0.585</td>
</tr>
<tr>
<td>C9</td>
<td>1.2</td>
<td>0.846</td>
</tr>
<tr>
<td>C3</td>
<td>1.6</td>
<td>0.684</td>
</tr>
<tr>
<td>C2</td>
<td>0.7</td>
<td>0.473</td>
</tr>
<tr>
<td>C6</td>
<td>0.3</td>
<td>0.426</td>
</tr>
</tbody>
</table>

CRD: Curriculum relation degree.

4.3. CBR-based module

4.3.1. The learning process of the CBR module

Summative assessments are used to evaluate the learning of a learner, based on broad course goals, in order to assign grades. There are two features that distinguish summative assessments from formative assessments. The first pertains to the portion of the course covered by the assessment. A summative assessment is much broader in scope and covers a much larger portion of the course. The second feature that distinguishes summative assessments from formative assessments is the level of generalization. Therefore, we apply CBR technology to analyze the summative assessment results. Besides, it also provides capability to support corrective activities, including different course materials and second formative assessment. The reasoning procedures of CBR are listed below and are shown in Fig. 10. Besides, a screen shot for a summative assessment is shown in Fig. 11.

4.3.1.1. Inputting a new case. A new case means that the learner fail to reach the mastery level in current unit. Therefore, a new learner case will trigger the CBR system to predict the probabilities of that specific case. The reasoning of the CBR mechanism will start to search for the case most similar to the new case in order to support the corrective activities. In the meantime, the learner can study different materials in the same concept. Afterward, they can also attend the second assessment which arranged by the CBR system.

4.3.1.2. Analyze an inquiry. A case consists of many indexes. How to choose the case index for important factors of case evaluation will significantly impact on the completeness of the case and its assessment results.

4.3.1.3. Assign index weight. Based on the different levels of importance among each index of the evaluation results, it is expected that the evaluation result for the new case will be more suitable. In this research, PLS-ML may extract rules and weights according to the thresholds that are set up by the experts. If the degree of similarity determined by a similarity function is higher than the fixed threshold value, then the system applies an existing similar case, or the system regards it as a nonexistent case and creates a new case in the case base.

4.3.1.4. Case retrieval. The case-based reasoning system saves a lot of data for reference. It locates the solution method for the most similar case by comparing similarities, and as such assesses the appropriateness. To efficiently retrieve the most similar case from the case library is an important task for a CBR system. Case retrieval usually falls into one of four categories: nearest neighbor, inductive learning, knowledge-guide, and any combination of these. Owing to the PLS-ML system dealing with more than one curriculum in the same time, its case retrieval method
integrates the knowledge-guide method and the weight ratio functionality (WRF) method.

The WRF method is derived from the nearest neighbor method with the concept of critical weights. According to the concept of “Matching a critical feature is much more important than matching many trivial features”, the weight of a trivial feature is set up as “0”. The rules in the rule base are adopted as the resources of the knowledge-guide method to guide the new case to a feature similarity group. Next, the WRF method is employed to retrieve the most
similar case from the case library. The overall similarity determined by WRF method matching function is mathematically represented as follows:

\[
\text{Similarity} = \frac{\sum W_i \cdot Z_i}{\sum W_i}, \quad \text{if } x_i < y_i \text{ then } Z_i = \frac{x_i}{y_i}, \text{ else } Z_i = \frac{y_i}{x_i}
\]

(5)

where \( W_i \) is the weight of the feature \( i \), \( x_i \) is the value of feature \( i \) in the new case, and \( y_i \) is the value of feature \( i \) in a case stored in the case library. The WRF method is simply based on the similar ratio. For example, assuming that the age value of the new case is “18” \((x_i)\) and the age value of a case in the case library is “20” \((y_i)\), then the similar ratio \((Z_i)\) is equal to “0.9” \((18/20)\).

4.3.1.5. Case adaptation and reuse. The purpose of case adaptation is to modify the retrieved case in order to solve the problems of the new case. Four adaptation methods may be applied in this research: (1) exact adaptation, (2) interpolation, (3) using mean values for adaptation, and (4) applying adaptation rules.

Exact adaptation requires an experiment that exactly describes the difference between two parameter values in the retrieved case and the new case, respectively. Interpolation is done linear and is weighted by the ratio between the footholds and the overall value range. Another choice for case adaptation is to use mean values. If a parameter value \( v_1 \) has to be adapted to a value \( v_2 \), the method of using mean values collects cases that contain either value \( v_1 \) or value \( v_2 \). The system calculates the mean values of the result parameters within these two groups. The ratio of the second mean value to the first mean value is used as an adaptation rule. Cases used for computing the mean values are weighted according to their usability. The final method of adaptation is applying the adaptation rules. Adaptation rules describe value changes, which are expressed by factors, differences, or qualitative statements. Factors are combined by multiplication while differences are added. However, the basic functionality of above-mentioned four adaptation methods is to redefine the adaptation rules for the CBR system used to adapt the retrieved case in solving the problems of the new case. In this research, the WRF method will generate the similar ratio that is a good reference for the appropriate course materials and second assessment in solving the problems of the new case.

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Values</th>
<th>Attributes</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>User_ID</td>
<td>Int user_ID</td>
<td>User_ID</td>
<td>Int user_ID</td>
</tr>
<tr>
<td>Gender</td>
<td>String m or f</td>
<td>Gender</td>
<td>String m or f</td>
</tr>
<tr>
<td>Age</td>
<td>Int age</td>
<td>Age</td>
<td>Int age</td>
</tr>
<tr>
<td>Item_number</td>
<td>Int item</td>
<td>Item_number</td>
<td>Int item</td>
</tr>
<tr>
<td>Item_unit</td>
<td>Int unit</td>
<td>Item_unit</td>
<td>Int unit</td>
</tr>
<tr>
<td>Sol_time</td>
<td>Time time</td>
<td>Sol_time</td>
<td>Time time</td>
</tr>
<tr>
<td>Expert_level</td>
<td>Double difficulty</td>
<td>Misconcept_no</td>
<td>Int misconcept_no</td>
</tr>
<tr>
<td>Similar_item</td>
<td>Int sim_item</td>
<td>Similar_item</td>
<td>Int sim_item</td>
</tr>
<tr>
<td>Diagnosis</td>
<td>Level of learners</td>
<td>Diagnosis</td>
<td>Inference of</td>
</tr>
<tr>
<td></td>
<td>knowledge</td>
<td></td>
<td>misconception</td>
</tr>
<tr>
<td>Solution</td>
<td>Provide adaptive</td>
<td>Solution</td>
<td>Correct</td>
</tr>
<tr>
<td></td>
<td>learning</td>
<td></td>
<td>misconception</td>
</tr>
</tbody>
</table>

Table 3
The case representation of attribute-value

![Fig. 12. A screen shot for the user interface on enrichment activities.](image-url)
4.3.1.6. Revision of a case. When the solutions of the similar cases that were picked out are not suitable for the new case, revisions can be conducted based on professional knowledge to make them more appropriate to the current case in order to determine the final results.

4.3.1.7. Saving a case. Save the case into case base to enhance its completeness and to consolidate the self-learning mechanism of the system.

4.3.2. Analytical case representation

When learners finish their first formative assessment, the record will be stored in the personal database. The record is historical information, including total score, correct answer, and wrong answer. Then, a new record will trigger the CBR system to predict the probabilities of that specific case. In this research, we use structure representation of the case table as shown in Table 3. The case table consists of the features (or attributes) and values given in pairs (A-V table). The features in each case contain attributes, diagnoses, and solution. The values are represented by the variable type, the variable name, and the weight value. Case tables generated by the system are divided into a correct case table and an incorrect case table. The correct case was used to compare the learners’ knowledge level with the expert module. The incorrect case was used to remedy the learners’ misconceptions or errors.

4.4. Enrichment activities

Enrichment activities may include tutoring special projects, problem-solving exercises, or any learning activity that is both stimulating and rewarding for fast learners. Screenshots for enrichment activities are shown in Fig. 12.

5. Evaluation of the PLS-ML system

The main purpose of this research was to examine the potential benefits of the PLS-ML system for student education. The PLS-ML system was used as Web tools to construct, administer, and analyze a test according to the current contents of the JAVA textbook of a university in Taiwan. In order to verify the feasibility of PLS-ML System, this research adopted real discharged cases. Besides, it also used following tests: (1) check case retrieval accuracy, (2) check retrieval consistency, (3) check for case duplication, and (4) global test.

(1) Check case retrieval accuracy, which means that if the case-base is queried with one of its cases, it should give the same case with similar measure, equals 100%. The case verification of this system selected 20 cases as subjects of calculating reasoning accuracy of the system. Fig. 13 shows the similarity measure against the case identification number, from which it is indicated that the average similarity of the 20 cases reasoned by the system is 88.65%. Therefore, it presents a high feasibility of PLS-ML, and can be used as a supporting system in decision making of appropriate course materials for learners, to help them to learn more effectively in a Web-based environment.

(2) Check retrieval consistency, which means that if exactly the same search has been performed twice, the same source cases should be retrieved with the same accuracy. As the result, the retrieval consistency performance of PLS-ML reached 100% representation.

(3) Check for case duplication, which a case should exactly match itself, but should not be identical to other cases. As the result, the case duplication performance of PLS-ML reached 100% representation.

<table>
<thead>
<tr>
<th>Case number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diagnostic results from educational experts</td>
<td>P</td>
<td>F</td>
<td>c3, c4</td>
<td>P</td>
<td>F</td>
<td>c8, c9</td>
<td>P</td>
<td>F</td>
<td>c3, c8</td>
<td>P</td>
</tr>
<tr>
<td>PLS-ML Consulting result</td>
<td>P</td>
<td>F</td>
<td>c3, c4</td>
<td>P</td>
<td>F</td>
<td>c8, c9</td>
<td>P</td>
<td>F</td>
<td>c3, c8</td>
<td>P</td>
</tr>
<tr>
<td>Similarity</td>
<td>.94</td>
<td>.86</td>
<td>.9</td>
<td>.88</td>
<td>1</td>
<td>.84</td>
<td>.96</td>
<td>.92</td>
<td>1</td>
<td>.98</td>
</tr>
<tr>
<td>Case number</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>Diagnostic results from educational experts</td>
<td>F</td>
<td>P</td>
<td>P</td>
<td>F</td>
<td>P</td>
<td>P</td>
<td>F</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>PLS-ML Consulting result</td>
<td>F</td>
<td>P</td>
<td>P</td>
<td>F</td>
<td>P</td>
<td>P</td>
<td>F</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>Similarity</td>
<td>.95</td>
<td>.97</td>
<td>.92</td>
<td>1</td>
<td>1</td>
<td>.85</td>
<td>.78</td>
<td>.89</td>
<td>.94</td>
<td></td>
</tr>
</tbody>
</table>

P: Pass; F: fail; c1–c10: Curriculum code.
(4) Global test that is important to verify the overall performance of the system. To evaluate the global performance of PLS-ML, quite a lot of tests are performed. We have chosen 10 problems of “Control Structure” in JAVA domain. The subjects for experiment were 20 beginning level university students. We have presented student responses to show the problem-solving methods of each student in Table 4. We have compared the educational expert with the diagnosed results of the implemented PLS-ML to determine the formative assessment results such as P (pass) and F (fail). The type of the misconception which the student is making dynamically created by combining the lower-case letter (c1,c2,c3,...). It presents students’ weak curriculums when they reached a wrong answer. Each student’s misconception in solving a problem is independently diagnosed by the system and human expert. The values of Table 4 are a result of the analyses by the system and by the human expert. The results illustrated that PLS-ML has high reliable and accurate diagnostic capabilities by high similarity values.

6. Conclusion

This research proposed a personalized curriculum generation approach based on the proposed GA-based module for personalized learning path generation and CBR-based module for personalized knowledge database and summative assessment analysis. The proposed learning path generation approach can simultaneously consider the curriculum difficulty level and the curriculum continuity of successive curriculums while implementing a personalized curriculum generation in the learning processes.

This paper makes three critical contributions: (1) it presents a genetic-based curriculum sequencing approach to provide personalized curriculum sequencing generation; (2) it illustrates the case-based reasoning to develop a summative examination or assessment analysis; and (3) it uses empirical study to indicate that the proposed approach can generate appropriate course materials to learners based on individual learner requirements, and help them to learn more effectively in a Web-based environment.

References


