**Construction of Dynamic Learning Path with Ant Colony Optimization and Rough Sets**

**S.Chakraverty1, Sushma Hans2, Aditya Bindal3**

Division of Computer Engineering, Netaji Subhas Institute of Technology, Delhi, India

***Abstract:*** *The development of e-learning based systems has motivated us to delve into adaptive learning opportunity, which is apposite to students' need and their preferences. Significant work has been done to make learning path dynamic based on various optimization techniques. This paper provides an absolutely novel idea to modify the learning path through the course dynamically using blend of Ant colony optimization and rough sets. We construct the learning path by modifying learner's ability after each topic covered based on her exam performance as well as recommendations from the learners who have completed the course successfully.* ***results***

**Keywords:** Adaptive e-Learning, Learning Aims, Ant Colony Optimization (ACO), Rough Sets

1. **Introduction**

Adaptive e-learning provides well-organized and formal learning by supporting diverse learning paths and materials to fit learners' varied needs and backgrounds [1]. Adaptive e-learning systems allow real time performance monitoring, access to alternate materials as well as record of past student's performance. That's why these systems are more beneficial than traditional learning. However, most of the adaptive e-learning systems are not user centric and provide learning paths and content designed by experts. We promote proficient learning based on collective intelligence that combines user centric learning and adaptive learning. The user-centric adaptation of learning path increase learners' understanding and allow them to have more insight into the course. One of those adaptation strategies is to construct dynamic learning path through the course based on her ability and interests.

This study aims to build up a user-centric adaptive learning system. This system targets to increase learners' coverage and depth of course whenever she performs well. However, it also proposes decreases the expected achievement when learner fails to deliver. We try to build adaptive path by modifying learners' ability after each topic covered. We utilize matrices and vector algebra to compute learners' ability. In addition, we consider learners similar to the target learner and take their recommendations regarding the perspective she should take to increase her performance. We figure out these recommendations based on tolerance on rough sets.

Section 2 presents the relevant work in adaptive e-learning. Section 3 provides an introduction to optimization algorithm and rough set theory. Section 4 introduces the proposed approach for generation of dynamic learning path. In Section 5, we discuss the simulated results. Section 6 includes the concluding remarks and finally the future work of the study is covered in section 7.

1. **Relevant Work**

The main intend of adaptive e-learning systems is to maximize learner's satisfaction along with improvement of learning and evaluation results [2]. To achieve this goal, several researches have been done to identify the dimensions of learner's disparities. Table 1 recapitulates some important adaptive e-Learning systems (ALS) that are developed in the last decade along with their personalization parameters.

|  |  |
| --- | --- |
| **ALS** | **Personalization Parameters** |
| UALS[1] | Material difficulty, learner's ability, prior knowledge |
| FORMATION PATH [2] | Educational level, learning style |
| TSAL [3] | Learning behavior, learning style |
| WELSA [4] | Unified Learning Style Model (ULSM) i.e. collection of learning preferences extracted from main learning style model |
| TANGOW [5] | Based on two learning styles of Felder-Silverman model i.e. sequential/global and sensation/intuitive |
| AHA [6] | Felder-Silverman Learning Style Model, navigation and media preference |
| DYLPA [7] | Learner profile including prior knowledge, learning preferences, analytical skill competency, Language proficiency, occupation |
| PERSONAL [8] | Learner's needs, preferences and requirements |
| Souvik [9] | Frequent pattern graph of course created through association rule mining |
| Andharini [10] | Student's knowledge level and feedback of material difficulty level |

Table 1: Adaptive e-Learning Systems categorized according to the personalization parameters

Most of them grant credence to learner's learning style.

UALS (User-centric Adaptive Learning System [1]) applies sequential pattern mining to mine sequence pattern from course structure. The system employs Item Response Theory (IRT) to assess learners' abilities and recommend the most appropriate content for them.

FORMATION PATH [2] provides an environment that works on a hybrid approach based on ACO and Collaborative Filtering (CF). CF algorithm saves the calculation process by recommending a learning path to a new learner with the same profile.

WELSA (Web-based Educational with Learning Style adaptation [4]) follows the unified model of learning style that incorporates characteristics of several models proposed in literature, to adapt courses to learners.

DYLPA (DYnamic Learning Path Advisor [7]) combines rule-based prescriptive planning and ACO based inductive mechanism to build up personalized learning pathways for each learner.

Souvik et. al [9] propose a data mining based creation of frequent pattern graph model from repository of e-learning contents. This graph defines the association and sequencing between the content. Subsequently, ACO is applied on the graph to derive an efficient and optimized path.

Andharini et. al [10] propose a personalized learning path generation scheme that simultaneously consider student's knowledge level and students' feedback of course difficulty level.

A scrutiny of the above mentioned studies discloses that most of the systems give credence to learner's learning style for adaptation. Moreover, each of these systems except DYLPA [7] utilizes, at most three personalization parameters only. We recognize that learner's ability is a very important personalization parameter. Learner's ability is changeable by nature means it changes as you gradually precede through the course. UALS [1] personalizes learning path on the basis of ability but did not take its variable nature into account.

We recognize that the changeable nature of learner's ability helps us to give precise optimal path for each learner. We also classify learners based on their learning achievement through tolerance relation of rough sets. This classification provides more general classes of learners that cover the vast variety of learners. This classification helps us to recommend various perspectives of a topic.

1. **Designing the System**

The proposed system exploits a hybrid approach of Ant Colony Optimization and Rough sets theory to discover optimal path for each learner. The course has been organized into various levels, where each level has several learning objects to be completed and those are being presented by multiple perspectives at that level[]. The system contains three major parts which includes modification of the ability based on performance of user, providing recommendation for future topics and the main framework providing the learning path to the user based on his learning aim and time remaining in the course.

* 1. **Performance Evaluation**

Evaluating the learner’s performance through a series of exams is being used to modify her ability in the course. As a part of system, the performance of a user is being evaluated after completion of a level and the marks obtained are being used to compute the modified value of ability of the user. The new ability is being computed by the use of following equation:-

**α[i+1] = α[i] × k + αcomp × (1-k)** (1)

where **α[i]**  is the value of learner’s ability at level ‘i’, **αcomp** is the value of the ability computed by the performance of student in the exam and **k** is the constant of adaptability.

The algorithm to compute the ability takes as input the array of relevance of each question for user (ques\_relevance), array of marks of user (user\_marks), ability of user (α) and number of questions in exam (numques) and provides the value of computed ability αcomp as output. The algorithm for computing **αcomp** has been shown in figure 1.

|  |
| --- |
| **ComputeAbility(**  **Input:**  ques\_relevance[numques], user\_marks[numques], ability α, numques  **Output:** Computed ability αcomp**)**  **Begin**   1. αcomp = 0; 2. **For** counter i = 1 to numques do    * αcomp = αcomp + ques\_relevance[i] \* user\_marks[i];   **End For**   1. **Return** αcomp   **End** |

**Figure 1. Compute Ability**

* 1. **Getting Recommendation**

After completing a particular topic and giving the exam for that topic, the user has various perspectives to choose from at the next level. In such a situation, the system, using the tolerance relations on rough sets[], provides a perspective that is recommended for the user, based on the students who have already completed the course in past and also are quite similar to the present student in terms of abilities and learning aims.

The following equation has been used to compute the total set of students that are similar to current student.

**{Ttotal}=(({T*path*}{T*time*})**

**{T*α*}){T*aim*}** (2)

where **Ttotal** is the set of similar students, **Tpath**gives the students which are in tolerance range with respect to the path value, **Ttime** is similar students with respect to time, **Tα** is similar students based on ability and **Taim** is students that have same aim as that of current student. The sets of similar students are computed as rough sets by applying tolerance relations and taking the pathValue, timeTaken, ability and learning aim as decision variables.

An algorithm for providing recommendations using past students’ performance has been given in figure 2.

* 1. **Main Adaptive Framework**

The system applies the meta-heuristic to provide the user, a path through the course which she has to follow to maximize her time, Coverage Factor (CF) and Depth Factor (DF) in the course and also minimizing the difficulty she faces during the course span [e-learning]. The module computes this path for the user after he clears a particular level in the course. This module uses the ACO [e-learning /ACO] for predicting paths and uses the recommendation modules as parameters for the ACO.

The algorithm needs the course graph (CG) given by list of perspectives (V) and list of edges (E), the priority table (LALOPT), contribution table (PACT), user’s learning aim (LAα), user’s learning ability (α), maximum time (Tmax), number of levels in course (numlevels) and source vertex (Vs) as inputs and gives Followed Learning Path(LPfollowed) as output[]. The algorithm is called Dynamic Perspectives and Aims for Learning (DPAL). The algorithm is shown in figure 3.

|  |
| --- |
| **Recommend(**  **Inputs:** level Index idlevel, LAα, α, CG=(V,E), current path LP  **Output:** next perspective to take Vp**)**  **Begin**  **Compute Tolerance**  Get the set of similar students using (2).  Find the total number of perspectives at level idlevel + 1 and store it in numpers  Initialize total[numpers], good[numpers] and success\_rate[numpers] as 0.  /\* array total \*/  **For** Student Sk: counter k=1 to |{Ttotal}| do  Vk = perspective taken by Sk on idlevel+1.  If Sk has passed the course then  Increment good[Vk]  End If  Increment total[Vk]  **End For**  **For** counter i=1 to numpers do  success\_rate[i] = good[i] / total[i]  **End For**  **Return** perspective with maximum success\_rate value  **End** |

**Figure 2. Recommendation by Past Students**

|  |
| --- |
| **DPAL(**  **Inputs:** CG=(V, E), LALOPT, PACT, LAα, α, Tmax, numlevels, Vs  **Outputs:** LPfollowed **)**  **Begin**   1. **Initialization Phase**  * Initialize LPfollowed as empty list * Initialize pathValue as 0 * Initialize Vtemp= Vs  1. **Start Giving Paths**   **For** counter i=1 to numlevelsdo   * Run ACO on the current course parameters and store the returned path in LPtemp. * Vi = vertices till level i in LPtemp * Add Vi to LPfollowed  1. **Modify the components**  * Tmax = Tmax – time spent on Vi * Make exam paper; store it in relevance [numques] and numques. * Compute marks of user, store in marks [numques]. * αcomp=**ComputeAbility**(relevance, marks, α, numques) * Adjust α using equation (1) * Vr=**Recommend**(i, LAα, α, CG, LPfollowed) * Set Vs as Vr. * Remove Vi and corresponding edges from CG.   **End For**   * **Return** LPfollowed   **End** |

**Figure 3. Dynamic Perspectives and Aims for Learning (DPAL)**

1. **Discussion and Results**
2. **Conclusion**
3. **Future Work**
4. **References**
5. Huang, S.-L., & Shiu, J.-H. (2012). A User-Centric Adaptive Learning System for E-Learning 2.0. *Educational Technology &Society*, *15* (3), 214–225.
6. An Adaptive Learning Based on Ant Colony and Collaborative Filtering
7. Tseng, C. R., Chu, H. C., Hwang, G. J., & Tsai, C. C. (2007). Development of an adaptive learning system with two sources of personalization information. *Computer & Education*, *51*(2), 776–786.
8. E. Popescu, “A Course Authoring Tool for WELSA Adaptive Educational System“. Proc. ICCC 2008, pp. 531-534. (2008). Accommodating Learning Styles in an Adaptive Educational System
9. P. Paredes, P. Rodriguez, “A Mixed Approach to Modeling Learning Styles in Adaptive Educational Hypermedia“. Advanced Technology for Learning, 1 (4), pp. 210– 215 (2004).
10. N. Stash, “Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System“. PhD Thesis, Eindhoven University of Technology, Netherlands.(2007).
11. Adaptable Learning Pathway Generation with Ant Colony Optimization
12. Adaptive learning paths for improving lifelong learning experiences
13. Construction of Learning Path Using Ant Colony Optimization from a Frequent Pattern Graph
14. Personalized Learning Path of a Web-based Learning System