Automated Adaptive Learning using Smart Shortest Path Algorithm for Course Units

Ibrahim Alkore Alshalabi
Department of Computer Science and Engineering
University of Bridgeport
Bridgeport, CT, USA
ialkorea@my.bridgeport.edu

Samir Hamada
Department of Computer Science and Engineering
University of Bridgeport
Bridgeport- CT, USA
shamada@my.bridgeport.edu

Khaled Elleithy
Department of Computer Science and Engineering
University of Bridgeport
Bridgeport, CT, USA
elleithy@bridgeport.edu

Abstract—The graph is a significant, considerable, and efficient representation of online courses in the computer based implementation of an educational system. E-learning and M-learning systems are modeled as weighted directed graphs where each node represents a course unit. The learning Path Graph represents and describes the structure of domain knowledge as well as the learning goals and all available learning paths. In this paper we propose an optimal adaptive learning path algorithm using learner information from the learner’s profile to improve E-learning and M-learning system in order to provide suitable course content sequence in a dynamic form for each learner.

Keywords: Adaptive Learning, M-learning, Shortest Path, User Profile.

I. INTRODUCTION

Adaptive e-learning researchers have explored and developed adaptive techniques that provide a better educational experience which is able to offer accurate and personalized contents to the learners in an intelligent way [1], i.e. dynamically adjust course contents to the learners based on their most recent actions, allows the learner to skip unnecessary activities and provide automated personalized support for the learner [2]. Learners with different knowledge background are the main challenge of the E-learning and M-learning systems to provide personalized course units that meet the different learner’s needs.

Learning systems are faced with the challenge of providing personalized units to learners with different knowledge backgrounds, in order to meet different learner needs. [4].

A. Learning Path Graph

The learning Path Graph represents and describes the structure of domain knowledge as well as the learning goals and all available learning paths [5][6]. In order to create and generate the Domain model, a two-step procedure is used: [6]

Step 1: Designing the Learning Goals and Concepts Hierarchies of the domain model.

Concepts Path Graph is a directed acyclic graph which represents the structure of the concept of domain model, generated from the connection between the Learning Goals Hierarchy and the Domain Concept.

The learning path graph is a directed acyclic graph which represents all possible learning paths that match the learning goal. To build the Learning Path graph, for each concept of the Concept Path graph, associated learning resources are selected from media space. Media space describes the educational characteristics of the learning resources.

Step 2: A personalized learning path is selected from the graph that contains all the available learning paths, based on the learner’s attributes in the user model. The user model has a level of expertise (learner knowledge space), learning style (cognitive characteristics) and preferences which provide suitability for the Learning Resources. Suitability function is used for weighing each connection of the Learning path graph. Finally, the shortest path algorithm is applied on the weighted graph to find the optimal learning path for specific learner.

B. User profile

The main challenge of the E-learning and M-learning systems is creating an appropriate adaptive course sequence to different learners with different knowledge backgrounds. One of the most important aspects, which have not been completely explored, is the capability of the learning system to adapt to the learner’s profile [7].

An optimal adaptive learning path algorithm that uses the learner information form of the learner profile to improve the E-learning and M-learning systems in order to provide suitable course content sequence in a dynamic form for each learner is introduced in [8].

C. Using graphs to represents course units

Graph is a significant and considerable method for the efficient representation of online courses in the computer based implementation of an educational system. E-learning and M-learning systems are modeled as a weighted directed graph where each node represents a course unit.

The course content is divided into portions called learning atoms that could be implemented at all levels of learning modes [9]. Those nodes contain the course concepts after partitioning the graph. The course concepts could be: Slide, Text, Examples, and Video etc. [9].
In this paper, we introduce automated adaptive learning using smart shortest path algorithm for course units. In section II, we discuss the updated shortest path algorithm. Section III provides experimentation. Section IV offers conclusions.

II. AN OPTIMAL SHORTEST PATH

In this paper, we introduce a graph representing the course units, which consists of N units as represented in Figure 1.

Let G be a graph with n vertices, where \( n \geq 0 \). Let \( V(G) = \{ v_1, v_2, \ldots, v_n \} \). Let W be a two dimensional \( n \times n \) matrix such that

\[
W(i,j) = \begin{cases} 
  w_{ij} & \text{if } (v_i, v_j) \text{ is an edge in } G \\
  \infty & \text{if there is no edge from } v_i \text{ to } v_j 
\end{cases}
\]

![Figure 1. Weighted graph represents the course units structure.](image)

The adaptive shortest path consists of two stages:

**Stage 1:**

Create an initial weighted matrix \( W = \{ w_{ij} \} \), where \( w_{ij} \) is the arrowhead weight from \( CUi \) to \( CUj \).

- If there is no arrowhead existing between the two CUs the \( W_{ij} = \infty \).
- For each \( i=j \) in \( W_{ij} = \infty \).

Let \( P \) be a two dimensional \( n \times n \) matrix such that for each \( P(i,j) = \{ p_{11} = p_{12} = p_{13} \ldots \ n \} \).

**Algorithm-1**

1. **Step 1**: Initialize \( CU \) by 0 and \( cmID \) by 1 (CU is ID number of course unit and \( cmID \) is cost matrix ID).
2. **Step 2**: for \( x=1 \) to \( N \) repeat step 3 to step 8 where \( N \) is equal to number of CUs.
3. **Step 3**: for \( i = 1 \) and \( i \leq N \) (Number of CUs) repeat step 4.
4. **Step 4**: for \( j=2 \) and \( j \leq N \) (Number of CUs) repeat step 5, step 6 and step 7.
5. **Step 5**: Compare \( W_{ij} > (W_{ix} + W_{xj}) \) if true [10]

   \[
   W_{ij} = (W_{ix} + W_{xj}) \quad \text{and} \quad P_{ij} = CU+1.
   \]
6. **Step 6**: Compare \( W_{ij} < (W_{ix} + W_{xj}) \) if true

   No change.
7. **Step 7**: Compare if \( W_{ij} = (W_{ix} + W_{xj}) \), if true

   \[
   \{ \quad W_{ij} = (W_{ix} + W_{xj}) \quad \text{and} \quad P_{ij} = CU+1.
   \]
   Create new \( PcmID+1 \) (i, j) =CU.
   (Alternative path for ij).
   \( cmID = cmID+1 \).
8. **Step 8**: \( CU = CU+1 \).
9. **Step 9**: End.

We assume there is no path between the same node, so for each \( i=j \), \( W_{ij} = \infty \).

We modify \( P_{ij} \). For each \( W_{ij} = \infty \) then \( P_{ij} = \infty \).
A. Stage 2:

Shortest path movement between CU_i and CU_j.

**Algorithm-2**

**Step 1**: CU_i start node and CU_j end node (Target node).

**Step 2**: if P_{ij} = \infty then no path between CU_i and CU_j go to Step 7.

**Step 3**: If P_{ij} = 0 the shortest is CU_i \rightarrow CU_j then Step 7 else Step 4.

**Step 4**: Repeat Step 5, Step 6 until P_{ij} = 0.

**Step 5**: CU_{ij} is next node in shortest path.

**Step 6**: i=P_{ij}.

**Step 7**: End.

III. EXPERIMENTS

A. Stage 1:

Create an initial weighted matrix W=wij, where wij is the arrowhead weight from CU_i to CU_j.

- If there is no arrowhead existing between the two CUS the W_{ij} = \infty.
- For each i=j in W_{ij} = \infty.

Initialize the weighted matrix W form graph in Figure 1 as shown in Table I.

<table>
<thead>
<tr>
<th>W(5,5)</th>
<th>CU-1</th>
<th>CU-2</th>
<th>CU-3</th>
<th>CU-4</th>
<th>CU-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>CU-1</td>
<td>\infty</td>
<td>8</td>
<td>7</td>
<td>11</td>
<td>\infty</td>
</tr>
<tr>
<td>CU-2</td>
<td>\infty</td>
<td>\infty</td>
<td>4</td>
<td>\infty</td>
<td>6</td>
</tr>
<tr>
<td>CU-3</td>
<td>\infty</td>
<td>5</td>
<td>\infty</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>CU-4</td>
<td>\infty</td>
<td>\infty</td>
<td>7</td>
<td>\infty</td>
<td>9</td>
</tr>
<tr>
<td>CU-5</td>
<td>\infty</td>
<td>\infty</td>
<td>\infty</td>
<td>\infty</td>
<td>\infty</td>
</tr>
</tbody>
</table>

Let P be a two dimensional n x n matrix such that for each P(i,j) = \{ p_{11} = p_{12} = p_{13} .... p_{nn} = 0 \}.

The result of creating an initial Path matrix P by zero, P matrix is shown in Table II.

<table>
<thead>
<tr>
<th>P(5,5)</th>
<th>CU-1</th>
<th>CU-2</th>
<th>CU-3</th>
<th>CU-4</th>
<th>CU-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>CU-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CU-2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CU-3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CU-4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CU-5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

After applying algorithm-1 in stage 1 we have Table III and Table IV:

<table>
<thead>
<tr>
<th>W(5,5)</th>
<th>CU-1</th>
<th>CU-2</th>
<th>CU-3</th>
<th>CU-4</th>
<th>CU-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>CU-1</td>
<td>\infty</td>
<td>8</td>
<td>7</td>
<td>11</td>
<td>14</td>
</tr>
<tr>
<td>CU-2</td>
<td>\infty</td>
<td>9</td>
<td>4</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>CU-3</td>
<td>\infty</td>
<td>5</td>
<td>9</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>CU-4</td>
<td>\infty</td>
<td>12</td>
<td>7</td>
<td>11</td>
<td>9</td>
</tr>
<tr>
<td>CU-5</td>
<td>\infty</td>
<td>\infty</td>
<td>\infty</td>
<td>\infty</td>
<td>\infty</td>
</tr>
</tbody>
</table>

Next, for each i=j in cost matrix assign \infty to W(i,j) to get Table V:

<table>
<thead>
<tr>
<th>W(5,5)</th>
<th>CU-1</th>
<th>CU-2</th>
<th>CU-3</th>
<th>CU-4</th>
<th>CU-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>CU-1</td>
<td>\infty</td>
<td>8</td>
<td>7</td>
<td>11</td>
<td>14</td>
</tr>
<tr>
<td>CU-2</td>
<td>\infty</td>
<td>3</td>
<td>\infty</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>CU-3</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>\infty</td>
<td>0</td>
</tr>
<tr>
<td>CU-4</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>\infty</td>
</tr>
<tr>
<td>CU-5</td>
<td>\infty</td>
<td>\infty</td>
<td>\infty</td>
<td>\infty</td>
<td>\infty</td>
</tr>
</tbody>
</table>

Next, for each W_{ij}=\infty assign \infty to P_{ij} to get Table VI:

<table>
<thead>
<tr>
<th>P(5,5)</th>
<th>CU-1</th>
<th>CU-2</th>
<th>CU-3</th>
<th>CU-4</th>
<th>CU-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>CU-1</td>
<td>\infty</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>CU-2</td>
<td>\infty</td>
<td>\infty</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>CU-3</td>
<td>\infty</td>
<td>0</td>
<td>\infty</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CU-4</td>
<td>\infty</td>
<td>3</td>
<td>0</td>
<td>\infty</td>
<td>0</td>
</tr>
<tr>
<td>CU-5</td>
<td>\infty</td>
<td>\infty</td>
<td>\infty</td>
<td>\infty</td>
<td>\infty</td>
</tr>
</tbody>
</table>
B. Stage 2:

By applying the algorithm in stage 2 to find the shortest path movement between learning CUs depending on Table 6 generated by stage 1.

Now we need to find:

1) The shortest path cost \( CU1 \rightarrow CU5 \).
2) The path between \( CU1 \) and \( CU5 \).

Solution of 1:

We look at Table 5, the minimum shortest path cost is \( W_{15} = 14 \).

Solution of 2:

From Table 6, the path between \( CU1 \) and \( CU5 \) could be found from \( P \) as follow:

Path from \( CU1 \rightarrow CU5 = 2 \) then we find path start from \( CU2 \rightarrow CU5 \), we look at \( P \) matrix from \( CU2 \) to \( CU5 \) we see \( CU2 \rightarrow CU5 = 0 \) then the end nod is \( CU5 \), so the path is \( CU1 \rightarrow CU2 \rightarrow CU5 \).

IV. ADAPT THE SHORTEST PATH ALGORITHM TO THE LEARNER RELEVANT PERSONAL INFORMATION

Assume the learner at the initial state logs into the system and learns \( CU1 \), after she/he successfully passes \( CU1 \), the Learner decides to learn \( CU2 \), the system will direct him to the shortest path \( CU1 \rightarrow CU2 \) that is equal to 8.

Assume the learner logs out, as we know from previous information that the learner learned \( CU1 \) and \( CU2 \). Now learner logged in again and decide to learn \( CU3 \), automatically the system will direct the learner to \( CU1 \rightarrow CU2 \rightarrow CU3 \), which is equal to 7 without taking into consideration that the learner learned \( CU2 \) which will affect the path that the learner may take according to his personal information.

To make the algorithm more adaptive the flow must be modified:

If the learner successfully passes the CU, the system will update the learner’s personal profile with new information.

According to this piece of information, if the learner decides to learn a new CU, the algorithm will find the shortest path between each CU that the learner has learned and the new CU the learner is planning to learn.

A. Algorithm 3

Step 1: For each CU that has been learned in the learner’s profile perform step 2.

Step 2: Find the shorts path between \( CU_i \) from the learner’s profile and the new target CU by using the result from Table 5.

Step 3: Find the minimum cost from all shortest paths to the new target CU according the successfully passed CU’s in learner profile.

Step 4: The shortest path according to the learner profile is \( CU_i (min \ cost) \) to new target CU.

Step 5: Use the algorithm in stage 2 and Table 6 to find the shortest path between learning \( CU_i \) (min cost) and new target CU.

Step 6: Update the personal profile if the new target CU is successfully passed.

Step 7: End.

B. Experiment

Assume the learner at the initial state logs into the system and she/he decides to learn \( CU2 \), the system will direct the learner to the shortest path \( CU1 \rightarrow CU2 \) which is equal to 8. This information is stored in the learner’s profile. The learner later logs in and decides to learn \( CU3 \), by applying Algorithm 3. According to the learner’s profile the system will compare \( CU1 \rightarrow CU3 = 7 \) with \( CU2 \rightarrow CU3 = 4 \) and will direct the learner to the path \( CU2 \rightarrow CU3 \) which is equal to 4.

V. CONCLUSIONS

The personalization of learning courses paths is the core feature of the E-learning and M-learning systems processes. The individual personalization-learning path is an important part in the educational system. Therefore, providing learners with an adaptive course which dynamically adjust to the personalized learning path during the learning process.

The proposed adaptive shortest path proposed in this paper, is a major step in constructing an adaptive M-learning system for several engineering disciplines.

REFERENCES


Forensic use to finding Uniqueness,” Computer Engineering and Intelligent Systems 4.9, 2013, 1-6.


Ibrahim M Alkore Alshalabi received the B.Sc. in Computer Science from Al-Isra Private University, Amman, Jordan in 1997, and the MCA( Master of Computer Applications ) from Bangalore University - India in 2007. In 2009 he joined University of Bridgeport as Ph.D. student in computer science and engineering at the University of Bridgeport, Connecticut-USA. From 1997 to 2004, he was Assistant Lecturer in Ma'an Community College - Al-Balqa Applied University-Jordan. From 2007 to 2009 he joined Al-Hussein Bin Talal University-Jordan as assistant lecturer. Ibrahim M Alkore Alshalabi has research interest is in the general area of E-Learning, M-Learning, wireless communications and networks. He actively participated as a committee member of International Conference on Engineering Education, instructional technology, Assessment, and E-Learning (EIAE 10, EIAE 11). Ibrahim M Alkore Alshalabi is a member of the International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering (CIBSE) and Ibrahim M Alkore Alshalabi is a member of IEEE.

Samir Hamada received the B.Sc. in Accounting from Ain Shams University, Cairo, Egypt in 1989, and the Master Degree of Computer Science from University of Bridgeport, Connecticut-USA in 2001; in 2008 he joined University of Bridgeport as Ph.D. Student in computer science and engineering at the University of Bridgeport, Connecticut-USA. Worked as an IT Manager for Atlantica Insurance Firm formally known as Action Auto Insurance From 2003 to Present, Samir thought the Data Computer Communication in the Spring and Fall of 2010, He also thought the course of Pre Calculus in the spring of 2011 and on Spring 2014 C++ and Data Structure, and currently teaching Mobile Communication, C++ and Data Structure. Worked in many Web development projects for Boehringer Ingelheim USA Corp, Publics Modern, Modern Media, Noble Americas, Dakota, Osprey, Byte Inter Active, Visual Concepts, Family Time, He was the web master for Connecticut Golf Club from 2005 till 2010. He was the only student to receive the First Info Tech scholarship at the University of Bridgeport at the year of 2000. He also received the General Diploma for Management Sciences in 1993 from the Sadat Academy, Cairo – Egypt.

Dr. Elleithy Dr. Elleithy is the Associate Vice President for Graduate Studies and Research at the University of Bridgeport. He is a professor of Computer Science and Engineering. He has research interests in the areas of wireless sensor networks, mobile communications, network security, quantum computing, and formal approaches for design and verification. He has published more than three hundreds research papers in international journals and conferences in his areas of expertise. Dr. Elleithy is the editor or co-editor for 12 books by Springer. He is a member of technical program committees of many international conferences as recognition of his research qualifications. He served as a guest editor for several International Journals. He was the chairperson for the International Conference on Industrial Electronics, Technology & Automation, IETA 2001, 19-21 December 2001, Cairo – Egypt. Also, he is the General Chair of the 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, and 2014 International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering virtual conferences.