Courseware Sequencing Using Heuristic and Local Search

Antonio Garcia[1], Luis de-Marcos[1], Eva García[1], and José-Antonio Gutiérrez[1]
[1] Computer Science Department, University of Alcalá, Alcalá de Henares, Madrid, Spain

Abstract - *This paper proposes two intelligent agents for learning object automated sequencing using heuristic and local search. In e-learning initiatives, sequencing problem concerns arranging a particular set of learning units in a suitable succession for a particular learner. Sequencing is usually performed by instructors, who create general and ordered series rather than learner personalized sequences. E-learning standards are promoted in order to ensure interoperability. Competencies are used to define relations between learning objects within a sequence, so that the sequencing problem turns into a permutation problem and AI techniques can be used to solve it. Heuristic and Local Search are two of such techniques. An implementation of the A* algorithm and Hill Climbing algorithm, for learning object sequencing, are presented and their performance in a real scenario is discussed.*

Keywords: Learning Objects, sequencing, heuristic search, local search

1 Introduction

Brusilovsky [1] envisages Web-based adaptive courses and systems as being able to achieve some important features including the ability to substitute teachers and other students support, and the ability to adapt (and so be used in) to different environments by different users (learners). These systems may use a wide variety of techniques and methods. Among them, curriculum sequencing technology "is to provide the students with the most suitable individually planned sequence of knowledge units to learn and sequence the learning tasks ... to work with". These methods derive from adaptive hypermedia field [2] and rely on complex conceptual models, usually driven by sequencing rules [3][4]. E-learning traditional approaches and paradigms, that promote reusability and interoperability, are generally ignored, thus resulting in (adaptive) proprietary systems (such as AHA! [5]) and non-portable courseware.

In this paper an innovative sequencing technique is proposed. E-learning standards and learning object paradigm are used in order to promote and ensure interoperability. Sequences of learning units are defined in terms of competencies in such a way that sequencing problem can be modeled like a classical Constraint Satisfaction Problem (CSP). Heuristic and Local Search are used to find a suitable sequence within the solution space satisfying all the constraints. In Section II, the problem model for competency-based learning object sequencing is presented. Section III describes the heuristic search using the A* algorithm for solving the problem. Section IV describes the local search using Hill Climbing Algorithm. Section V presents the results obtained from implementing and testing the intelligent algorithm in simulated tests and in a real world situation (course sequencing in an online Master in Engineering program). Section VI presents related work. And finally Section VII depicts conclusions and future research lines.

2 Learning objects and sequencing

Within e-learning, the learning object paradigm drives almost all initiatives. This paradigm encourages the creation of small reusable learning units called Learning Objects (LOs). These LOs are then assembled and/or aggregated in order to create greater units of instruction (lessons, courses, etc.) [6].

LOs must be arranged in a suitable sequence previously to its delivery to learners. Currently, sequencing is performed by instructors who do not create a personalized sequence for each learner, but instead create generic courses, targeting generic learner profiles. These sequences are then coded using a standard specification to ensure interoperability. Most commonly used specification is SCORM [7]. Courseware that conforms SCORM's Content Aggregation Model [8] is virtually portable between a wide variety of Learning Management Systems (LMSs). Though, SCORM usage hinders the automatic LO sequencing due to its system-centered vision. Other metadata-driven approaches offer better possibilities. Just LO metadata will enable automatic sequencing process to be performed. And the appropriate combination of metadata and competencies will enable adaptive and automatic content sequencing.

2.1 Competencies for interoperable Learning Object Sequencing

Competencies can be formally described as "multidimensional, comprised of knowledge, skills and psychological factors that are brought together in complex behavioral responses to environmental cues" [9]. Some e-learning trends are trying to standardize competency definitions so that they could be interchanged and processed by machines. It is worth quoting the following specifications:

- IMS "Reusable Definition of Competency or Educational Objective" (RDCEO) specification [10],
- IEEE Learning Technology Standards Committee (LTSC) "Standard for Learning Technology - Standard for Reusable Competency Definitions" specification [11],
- and HR-XML Consortium "Competencies (Measurable Characteristics) Recommendation" [12].

According to RDCEO and IEEE nomenclature, a competency record is called 'Reusable Competency Definition' (RCD). RCDs can be attached to LOs in order to define their prerequisites and their learning outcomes. We have used this approach to model LO sequences. By defining a competency (or a set of competencies) as a LO outcome, and by defining the same competency as the prerequisite for another LO (figure 1), a constraint between the two LOs is established so that the first one must precede the second LO in a valid sequence.

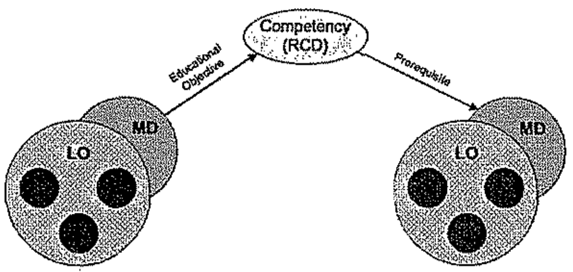

Figure 1. LO sequencing through competencies

Metadata (MD) definitions are attached to LOs, and within those definitions references to competencies (prerequisites and learning outcomes) are included. LOM (Learning Object Metadata) [13] records have been used for specifying LO metadata. LOM element 9, 'Classification', is used to include competency references as recommended in [14][15]. So, LOM element 9.1, 'Purpose', is set to 'prerequisite' or 'educational objective' from among the permitted vocabulary for this element; and LOM element 9.2 'Taxon Path', including its sub-elements, is used to reference the competency (note that more than one Classification element can be included in one single LO in order to specify more than one prerequisite and/or learning outcome).

Simple metadata (i.e. LOM records) is enough to model LOs' sequences in a similar way. So, why use competencies? Competency usage is encouraged, besides its usefulness for modeling prerequisites and learning outcomes, because competencies are also useful for modeling user current knowledge and learning initiatives' expected outcomes (future learner knowledge). We are proposing a wider framework (see Section VII) in which learner (user) modeling is done in terms of competencies, and these competencies are also used to define the expected learning outcomes from a learning program.

2.2 Problem of Learning Objects Sequencing

Given a random LOs' sequence modeled as described above, the question of finding a correct sequence can be envisaged as a classical artificial intelligent Constraint Satisfaction Problem (CSP). In this manner, the solution space comprises all possible sequences (*n!* will be its size, total number of states, for *n* LOs), and a (feasible) solution is a sequence that satisfies all established constraints. LO permutations inside the sequence are the operations that define transitions between states. So we face a permutation problem, which is a special kind of CSP.

2.2.1 Fitness Function

It is critical to choose a function that accurately represents the goodness of a solution. The following function is suggested:

$$\mathrm{fitness}(s) = \sum_{i=0}^{n} s[i].\mathrm{pr}_n \tag{1}$$

Where s is the LO sequence, n is the number of LOs in *s*, *s[i]* is the i-th LO in the sequence, and pr_n is the number of prerequisites in a LO not delivered by their predecessors in the sequence. pr_n is computed using a function that recursively process all outcomes delivered by previous LOs in the sequence, checking for each prerequisite accomplishment. The fitness value of a feasible solution should be zero, so the algorithms try to minimize this function. When a solution fitness function call returns 0, the operation of the algorithm is stopped returning the current state (solution).

2.2.2 Successor generation

Many heuristic and non-heuristic algorithms need to generate successor states from current state to work. A possible way to do this is to swap each LO in the sequence with every other position in the sequence (Figure 2).

The first LO is then permuted with the second LO, with the third one, etc., and finally with the last one. Then, the second LO is permuted with the third, with the fourth, etc. Each of these permutations is a successor state. In this way we generate all possible successors, *n*(n-1)/2* states, where *n* is the number of LOs' in the sequence.

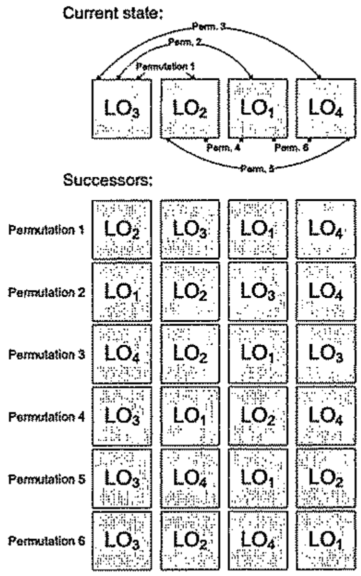

Figure 2. Example of generation of successors

3 Heuristic Search in Learning Object Sequencing

This section describes the proposed of heuristic search to solve the learning object sequencing problem. Heuristic search algorithms use an estimate cost of the solution to guide their search, this implies the existence of an evaluation function that should measure the estimated distance to the solution (*fitness(n)*). This evaluation function is used to guide the process, selecting the most promising state to explore. A* is the heuristic search algorithm used. Its evaluation function has the form: *fitness(n) = g'(n)+h'(n)*

This evaluation function has two components:

- h' (heuristic) is a estimated value of cost to arrive to the objective.
- g' is the real cost. Minimal cost between initial state and current state.

This algorithm also uses two lists of nodes: the first store the opened nodes that are generated from the exploration of the nodes, the second list store the visited nodes. The second list prevents exploration of nodes that have already been visited.

The Figure 3 shows the pseudo-code of A* algorithm for learning object sequencing.

```
OpenList.insert(initial_state)
current=OpenList.first()
While no is_solution?(current) and no OpenList.empty?() Do
    OpenList.delete_first()
    CloseList.insert(current)
    successors=generate_successors(current)
    successors=delete_repeated(successors,
CloseList, OpenList)
    OpenList.insert(successors)
    current=OpenList.first()
EndWhile
```

Figure 3. Pseudo-code of A* algorithm

The search will finish when a solution is found or when the algorithm has explored all possible states and has not found a solution.

4 Local Search in Learning Object Sequencing

The local search is performed from an initial solution that is subsequently improved. In this case we propose the use of Hill Climbing algorithm for the learning object sequencing. This algorithm selects the best state of those that entail any heuristic improvement over the current state. This algorithm has several problems which must be considered:

- Local Maxima: No successor improves the current state.
- Plateau: Set of nodes that have the same fitness than current state.

To solve the local maxima problem, "random restart" is used. With this if a local maximum is found, the algorithm is restarted with a new random sequence and it continues the exploration. To avoid the plateau problem happens, it is proposed to "climb" always to a state that improves the current state, avoiding explore states that have the same fitness.
Next figure (Figure 4) shows the pseudo-code of the Hill Climbing algorithm with random restart.

```
current=initial_state
While no is_solution?(current) and no maxReset Do
    successors=generate_successors(current)
    current=best_option(successors)
    if no best option
        current=random_reset()
EndWhile
```

Figure 4. Pseudo-code of Hill Climbing algorithm

This algorithm will finish their search when it has found a solution, or has been restarted a predefined number of times.

5 Results

The A* and Hill Climbing algorithms for LOs sequencing described above were implemented using Microsoft Visual Studio C#. Different tests were conducted to evaluate algorithms' performance. Simulated test cases with 5, 10, 20, 30, 40, 50, 60, 75 and 100 LOs with just one feasible solution were designed. Each algorithm was run 100 times for each test case and mean values were computed (figure 5).

Looking the statistics we can not determine that there is a large difference in performance in both algorithms (the two graphics are overlapping) because the number of calls to fitness function for the same number of learning objects is similar in both algorithms.

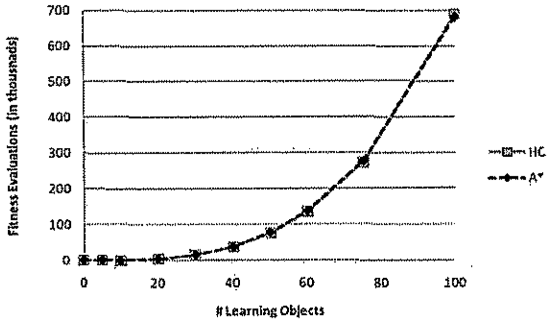

Figure 5. Statistics of A* and Hill Climbing

Their performance has also been tested in a real scenario. A problem concerning course sequencing for a Master in Engineering (M.Eng.) program (figure 6) in our institution was chosen for testing. The (web engineering) M.Eng. program comprises 23 courses (subjects) grouped in:

- Basic courses (7) that must be taken before any other (course). There may be restrictions between two basic courses, for example 'HTML' course must precede Javascript course,
- 'Itinerary' courses (5) that must be taken in a fixed ordered sequence.
- Compulsory courses (5). There may be restrictions between two compulsory courses.
- Elective courses (6). Additional constraints regarding any other course may be set.

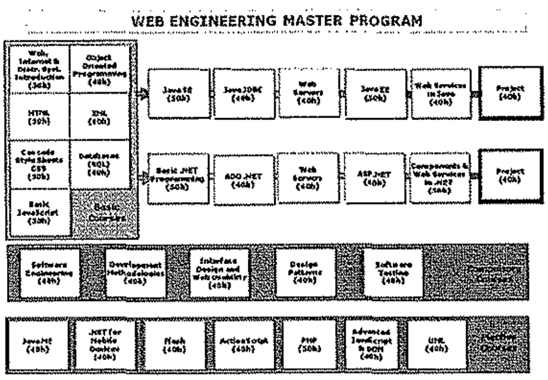

Figure 6. Web Engineering Master Program

All courses have a (expected) learning time that range from 30 to 50 hours. They are delivered online using a LMS and they have their metadata records. Competency records were created to specify LOs' restrictions, and LOs' metadata records were updated to reflect prerequisite and learning outcome competencies as detailed in Section II. A feasible sequence must have 23 LOs satisfying all constraints. The graph showing all LOs and constraints is very complex, and so it is to calculate the exact number of feasible solutions. Just estimations have been used. We have estimated that the relation between feasible solutions and total solutions order is $8{,}9\times10^{12}$. This number reflects the number of states (non-feasible solutions) for each feasible solution.

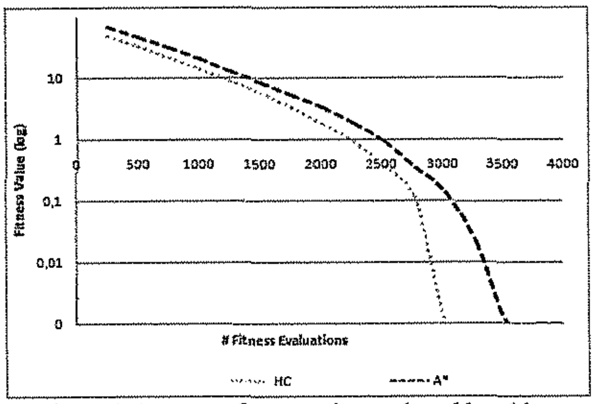

Figure 7. Agents performance in a real world problem

Once the problem is represented, each algorithm was run 100 times with this sequence to obtain statistics to compare the performance in both algorithms (Figure 7).

Looking theses results we can determine that the Hill Climbing algorithm is better adapted to this specific problem because it converges soonest.

6 Related Work

There are some works related to this paper, especially in the field of curriculum sequencing [16], which provide different methods of representation of learning object sequencing:

- UML based sequencing: there are adaptations of UML to model hypermedia systems and educational hypermedia systems with two principal systems: UML-Guide y CADMOS-D. The first proposes to use state diagrams to model the navigational structure that the users must follow, and the second proposes a model based in a UML-profile which extends the basic elements of UML.
- Graphs bases sequencing: graphs are used to define LO sequences. The two most relevant approaches are AHA! and sequencing graphs [17].
- Stochastic sequencing: proposes an approach that includes a balance of probabilities to define the sequence of activities.
- Competency-based sequencing: these systems create the sequences bases on the skills or knowledge required and provided the realization of a particular learning object. In this category would be this paper.

Others works are closely related to this paper, one of them proposes an algorithm PSO (Particle Swarm Optimization) for the learning object sequencing [18] and other shows the comparison between the PSO algorithm, mentioned above, and an algorithm AG (algorithm genetic) [19].

7 Conclusions and Future Work

The purpose of the study was to design, develop and test two agents that perform automatic LO sequencing through competencies. The algorithms A* and Hill Climbing have been extended to LO sequencing problem. Results show that: (1) both algorithms solve the problem, but (2) Hill Climbing performs better.

Further implications arise from the model proposal (Section II): (1) E-learning standards are promoted. XML records and bindings are used, so elements will be easily interchanged and processed by compliant systems. (2) Instructor's role is automated, with reduced costs. Sequencing process works even in complex scenarios were humans face difficulties. And (3), the model can be extended to an automated intelligent system for building personalized e-learning experiences. But this third implication is more appertained to future work. This model has been envisaged and is depicted in figure 8. Sequencing process can be complemented with gap analysis process and competency learner modeling techniques to build personalized courses. This courses could also be SCORM [7] compliant, so they could be imported to current LMSs.

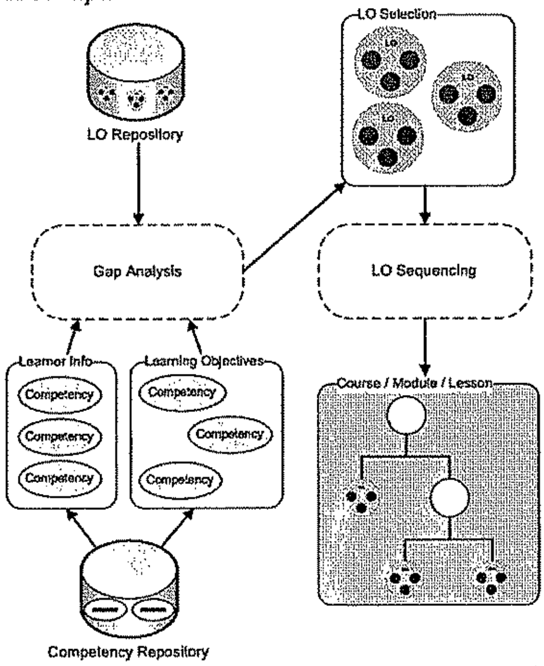

Figure 8. Competency-driven content generation model

Finally, other intelligent sequencing techniques should be analyzed. Particularly, ant colony optimization (ACO) [20][21] has proven optimal results regarding CSP solving. We plan to design and build an intelligent sequencing agent using this technique and check its results against implementation performance.

8 References

[1] P. Brusilovsky, "Adaptive and Intelligent Technologies for Web-based Education," Künstliche Intelligenz, Special Issue on Intelligent Systems and Teleteaching, vol. 4, pp. 19-25, 1999.

[2] P. Brusilovsky, "Methods and techniques of adaptive hypermedia," User Modeling and User-Adapted Interaction, vol. 6, pp. 87-129, 1996.

[3] P. De Bra, G.-J. Houben, and H. Wu, "AHAM: a Dexter-based reference model for adaptive hypermedia," in Proceedings of the tenth ACM Conference on Hypertext and hypermedia Darmstadt, Germany: ACM Press, 1999.

[4] P. Karampiperis, "Automatic LO Selection and Sequencing in Web-Based Intelligent Learning Systems," in Web-Based Intelligent E-Learning Systems: Technologies and Applications, M. Zongmin, Ed. London. UK.: Idea Group, 2006.

[5] P. De Bra, A. Aerts, B. Berden, B. d. Lange, B. Rousseau, T. Santic, D. Smits, and N. Stash, "AHA! The adaptive hypermedia architecture," in Proceedings of the fourteenth ACM conference on Hypertext and hypermedia Nottingham, UK: ACM Press, 2003.

[6] D. A. Wiley, "Connecting LOs to instructional design theory: A definition, a metaphor, and a taxonomy," in The Instructional Use of LOs, D. A. Wiley, Ed., 2000.

[7] ADL, "Shareable Content Object Reference Model (SCORM). The SCORM 2004 Overview," Advanced Distributed Learning (ADL) Initiative, 2004.

[8] ADL, "Shareable Content Object Reference Model (SCORM). The SCORM 2004 Content Aggregation Model." vol. 2005: Advanced Distributed Learning (ADL) Initiative, 2004.

[9] J. Wilkinson, "A matter of life or death: re-engineering competency-based education through the use of a multimedia CD-ROM," in IEEE International Conference on Advanced Learning Technologies, 2001. Proceedings, 2001, pp. 205-208.

[10] IMS, "Reusable Definition of Competency or Educational Objective - Information Model," IMS Global Learning Consortium, 2002.

[11] IEEE, "Learning Technology Standards Committee (LTSC). Standard for Learning Technology - Data Model for Reusable Competency Definitions," IEEE, 2007.

[12] HR-XML, "Competencies (Measurable Characteristics) Recommendation," HR-XML Consortium, 2006.

[13] IEEE, "Learning Technology Standards Committee (LTSC). LO Metadata (LOM). 1484.12.1," IEEE, 2002.

[14] IEEE, "Learning Technology Standards Committee (LTSC). Standard for Learning Technology—Extensible Markup Language (XML) Schema Definition Language Binding for LO Metadata. 1484.12.3.," IEEE, 2005.

[15] IMS, "Reusable Definition of Competency or Educational Objective - Best Practice and Implementation Guide," IMS Global Learning Consortium, 2002.

[16] Gutiérrez, S. & Pardo, A. (2007) Sequencing in Web-Based education: Approaches, Standards and Future Trends. Studies in Computational Intelligence (SCI). Berlin (Germany), Springer-Verlag.

[17] Gutiérrez, S., Pardo, A. & Kloos, C. D. (2004) Beyond simple sequencing: Sequencing of learning activities using hierarchical graphs. Web-Based Education 2004

[18] De Marcos, L., Barchino, R. Martínez, J.J., Gutiérrez, J.A.: (2009). A New Method for Domain Independent Curriculum Sequencing: A Case Study in a Web Engineering Master Program. International Journal of Engineering Education (IJEE), Dublin (Ireland), Vol. 25, n. 5, 2009, ISSN: 0949-149X, pp. 632-645. Dublin, Ireland.

[19] De Marcos, L., Martínez J.J., R., Gutiérrez, J.A., Barchino, R., Gutiérrez J.M.: (2009). A New Sequencing Method in Web-Based Education. 2009 IEEE Congress on Evolutionary Computation CEC-2009 , Trondheim (Norway), 18–21 mayo, 2009. ISBN: 978-1-4244-2959-2, pp. 3219-3225.

[20] M. Dorigo and G. D. Caro, "The Ant Colony Optimization meta-heuristic," in New Ideas in Optimization, D. Corne, M. Dorigo, and F. Glover, Eds. London, UK: McGraw Hill, 1999, pp. 11-32.

[21] M. Dorigo, G. D. Caro, and L. M. Gambardella, "Ant algorithms for discrete optimization," Artificial Life, vol. 5, pp. 137-172, 1999.