

# Competency-based Intelligent Curriculum Sequencing using Particle Swarms

Luis de-Marcos, Roberto Barchino, José-Javier Martínez, José-Antonio Gutiérrez  
Computer Science Department. University of Alcalá.  
Ctra. Barcelona km 33.6, Alcalá de Henares, Spain  
{luis.demarcos, roberto.barchino, josej.martinez, jantonio.gutierrez}@uah.es

## Abstract

*As a part of many e-learning initiatives, a set of learning units must be arranged in a particular order to meet the learners' requirements. This process is known as sequencing and it is typically performed by instructors, who create wide-public ordered series rather than learner personalized sequences. This paper proposes an innovative intelligent technique for learning object automated sequencing using particle swarms. E-Learning standards are promoted in order to ensure interoperability. Competencies are used to define relations among learning objects within a sequence, so that the sequencing problem turns into a permutation problem and a particle swarm optimization algorithm can be applied to solve it. Results demonstrate that the new agent succeeds and it shows a good performance in real and tests scenarios.*

## 1. Introduction

Curriculum sequencing is a technique used to build adaptive learning resources whose main objective “*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*” [1]. These methods derive from the adaptive hypermedia field and rely on complex conceptual models, usually driven by sequencing rules that result in proprietary systems and non-portable courseware. The Learning Object (LO) paradigm encourages the creation of small reusable learning units which are later assembled and/or aggregated in order to create greater units of instruction, such as lessons or courses [2]. A set of standards and specifications promotes interoperability, but really hinders automatic content adaptation and consequently LOs reusability [3].

In this paper an intelligent agent that performs the sequencing process is proposed. LOs' sequences are defined in terms of competencies in such a way that

sequencing problem can be modeled like a classical Constraint Satisfaction Problem (CSP), and Particle Swarm Optimization (PSO) is used to find a suitable sequence within the solution space.

## 2. Competency-based 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*” [4]. Some e-learning trends are trying to standardize competency definitions so that they could be interchanged and processed by machines. The oldest and most used specification is IMS “*Reusable Definition of Competency or Educational Objective*” (RDCEO) [5]. 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. Metadata (MD) definitions are attached to LOs, and within those definitions references to competencies (prerequisites and learning outcomes) are included.

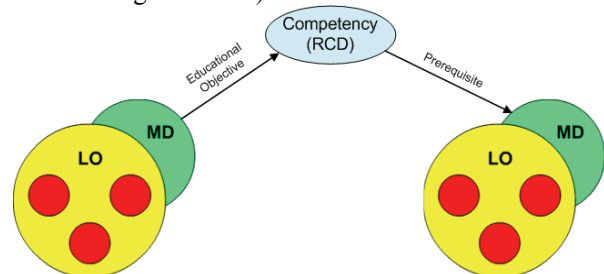


Figure 1. LO sequencing through competencies

Given a random LOs' sequence modeled in this way, the question of finding a correct sequence can be envisaged as a classical CSP. 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. LOs' permutations inside the sequence are the operations that define transitions between (candidate) solutions. So we face a permutation CSP (permut-CSP).

PSO is an evolutionary computing optimization algorithm that mimics the behavior of social insects like bees. The original PSO [6] is intended to work on continuous spaces. A version that deals with permutation problems was introduced in [7]. We take this approach and a permutation full-informed PSO was implemented in order to test its performance for solving the LO sequencing problem. A standard penalty function was used as fitness function. Parameters were set according to [7] and population size was set to 20.

During the agent development we found that in some situations the algorithm got stuck in local minimums, and it was not able to find a feasible solution. For that reason, three different tuning mechanisms were envisaged in order to improve agent performance. First option is to change  $pbest$  and  $gbest$  values when an equal or best fitness value is found by a particle (comparisons with the current state were set to less or equal ( $\leq$ )). The second tuning mechanism is to randomly decide whether the permutation of a particle's position was performed from  $gbest$  or from  $pbest$  ( $p=0.5$ ). Finally a velocity check mechanism was also tested. Each velocity value was limited to the number of LOs in the sequence.

### 3. Results

The agent was implemented using Microsoft Visual Studio C#. A problem concerning course sequencing for a Master in Engineering (M.Eng.) program in our institution was chosen for testing. The (web engineering) M.Eng. program comprises 23 courses (subjects) grouped in:

- Basic courses (7). All of them must be completed before taking any other kind of 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 exist.

All courses have a (expected) learning time that range from 30 to 50 hours. They are delivered online using a LMS [8] 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. A feasible sequence must have 23 LOs satisfying all constraints.

Once the problem was established, PSO agent was set to test four different configurations that reflect all possibilities concerning the first two tuning mechanisms. These configurations are:

- Configuration 1. Comparisons for changing particle  $pbest$  and  $gbest$  values are set to strictly less ( $<$ ). Permutation of the particle position is performed regarding  $gbest$ . These are the original settings.
- Configuration 2. Comparisons set to less or equal ( $\leq$ ). All permutations are performed from  $gbest$ .
- Configuration 3. Comparison set to strictly less ( $<$ ). Permutation of the particle position is randomly selected from  $gbest$  or from  $pbest$ .
- Configuration 4. Comparison set to less or equal ( $\leq$ ). Permutations from  $gbest$  or from  $pbest$ .

Each configuration was run 100 times and results representing mean fitness values' evolution were computed (figure 2). All configurations converge to a feasible solution, but configuration 1 (original settings) outperform all others. Configurations 1 and 2 show similar performance but configuration 1 reaches before any other a 100% success ratio for 100 runs.

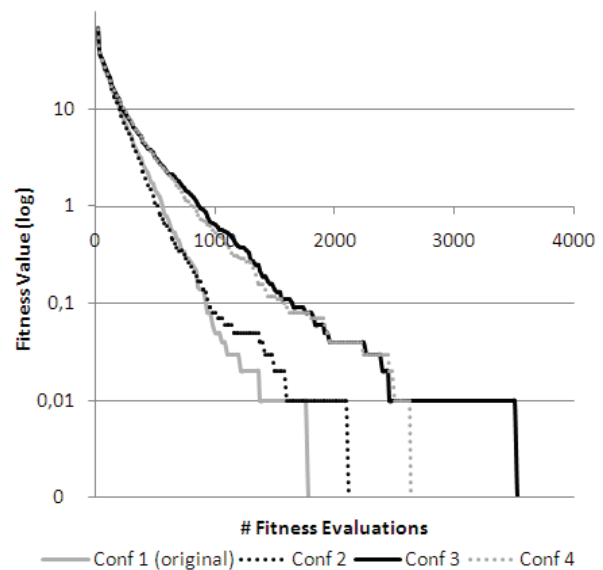


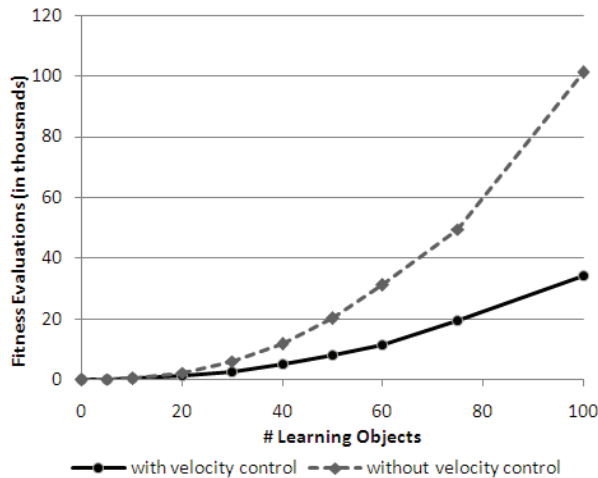
Figure 2. PSO Configurations performance

All these tests were run checking the normalized velocity limit. In order to test the real performance of this improvement, the four configuration sets were run without performing the velocity check (table1). Velocity check dramatically improves performance, and configuration 1 also displays better performance in both cases.

**Table 1. Mean number of fitness evaluations for each configuration (100 runs)**

Configuration	Without Velocity Check	With Velocity Check
Conf 1.	1158	641
Conf 2.	1237	645
Conf 3.	1817	1008
Conf 4.	1412	975

The tested scenario may seem to have many feasible solutions that would make doubtful PSO performance in more ‘challenging’ scenarios, so PSO agent was tested in ‘more difficult’ situations. Test sequences of 5, 10, 20, 30, 40, 50, 60, 75 and 100 LOs with only one feasible solution were designed. Figure 3 shows the results and it supports the argument that velocity control improves agent performance as the solution space size grows. It could also be inferred that the proposed PSO agent handles reasonably combinatorial explosion for this particular problem.



**Figure 3. Mean number of fitness evaluations required for different number of LOs (100 runs)**

#### 4. Conclusion

The purpose of the study was to design, develop and test a PSO agent that performs automatic LO sequencing through competencies. The permutPSO

have been extended to solve the LO sequencing problem. The algorithm has been also properly tuned. Results show that: (1) PSO succeeds in solving the problem, (2) the original configuration is the best one, and (3) a velocity check that limits the normalized velocity of each particle value improves performance in the tested scenarios.

#### 5. Acknowledgments

This research is co-funded by: (1) UAH FPI program, (2) MITyC PROFIT and Plan Avanza programs (grants FIT-350200-2007-6 and PAV-070000-2007-103), (3) MEC PROFIT program (grant CIT-410000-2007-5) and (4) JCCM educational innovation cooperation program (grant EM2007-004).

#### 6. 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] D. A. Wiley, "Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy," in *The Instructional Use of Learning Objects*, D. A. Wiley, Ed., 2000.
- [3] N. A. Abdullah and H. Davis, "Is simple sequencing simple adaptive hypermedia?," in *Proceedings of the fourteenth ACM conference on Hypertext and hypermedia* Nottingham, UK: ACM Press, 2003.
- [4] 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.
- [5] IMS, "Reusable Definition of Competency or Educational Objective - Information Model," IMS Global Learning Consortium, 2002.
- [6] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science. MHS '95.*, Nagoya, Japan, 1995, pp. 39-43.
- [7] X. Hu, R. C. Eberhart, and Y. Shi, "Swarm intelligence for permutation optimization: a case study of n-queens problem," in *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*, Indianapolis, USA, 2003, pp. 243-246.
- [8] R. Barchino, J. M. Gutiérrez, and S. Otón, "An Example of Learning Management System," in *IADIS Virtual Multi Conference on Computer Science and Information Systems (MCCSIS 2005)*, Virtual, 2005, pp. 140-141.