

Research

Open Access



A low complexity heuristic to solve a learning objects recommendation problem

Samuel Henrique Falci^{1*†} , Fabiano Azevedo Dorça^{1†}, Alessandro Vivas Andrade^{2†} and Daniel Henrique Mourão Falci^{3†}

*Correspondence:

samuehfalci@gmail.com

[†]Samuel Henrique Falci, Fabiano Azevedo Dorça, Alessandro Vivas Andrade and Daniel Henrique Mourão Falci contributed equally to this work.

¹Universidade Federal de Uberlândia, Avenue João Naves de Ávila, 2121, Uberlândia, MG, Brazil
Full list of author information is available at the end of the article

Abstract

The recommendation of learning objects in virtual learning environments has become the focus of research to improve online learning experience. Several approaches have been presented in an attempt to model the individual characteristics of the students and offer learning objects that best suit their particularities. Most of them, though, are impractical in real-world scenarios due to the high computational cost as a huge number of repositories offering learning objects such as Youtube, Wikipedia, Stackoverflow, Github, discussion forums, social networks and many others are available and each has a large amount of learning objects that can be retrieved. In this work, we propose a low complexity heuristic to solve this problem, comparing it to a classical mixed-integer linear programming model and classical genetic algorithm in varying dataset sizes that contain from 2000 to 1360000 learning objects. Performance and optimality were analyzed. The results showed that the proposed technique was only slightly suboptimal, while its computational cost was considerably smaller than the one presented by the linear optimization approach.

Keywords: Learning object recommendation, Virtual learning environment optimization, Heuristics

Introduction

Given the popularization of the Internet, people increasingly choose online learning, which gave rise to systems known as Virtual Learning Environments (VLE). According to a report published by *Markets and Markets* (Markets 2019), the VLE market value is expected to grow from USD 9.2 billion in 2018 to USD 22.4 billion by 2023. In turn, the content available on such platforms has never been so vast and increases by exponential rates what hampers its usage. In this scenario, a vital feature at the core of any VLE is the ability to recommend the appropriate learning objects (LOs) for each learner considering a given topic.

The learning objects recommendation problem (LORP), though, is a challenging task. It requires a system that is able to automatically infer the behavior, cognitive level, and learning styles (LS) of its users. The detection of these characteristics were addressed in studies such as (Stathacopoulou et al. 2005; Crockett et al. 2011; Graf et al. 2008; Ferreira

et al. 2016) and others. Since it is possible to detect student characteristics and recommend LOs according to these characteristics, the challenge is to choose the best possible LOs from a given set at a given time for a specific student. Currently, there are huge LOs repositories to be exploited, besides tools such as Google Scholar, Wikipedia, Universities databases, YouTube, Discussion Forums and more.

Each of the repositories has a large number of LOs. As can be seen on Wikipedia website itself for example, considering only the English language, almost 6 million articles are available and can be converted into real learning objects that can be used by recommendation systems in Smart Learning Environments (SLE). SLE are VLE that employ Artificial Intelligence (AI) techniques to improve learning process with the required pedagogical shift (Chen et al. 2016).

To solve LORP, previous approaches employed evolutionary computation techniques such as Genetic Algorithms (GA), for example. These studies usually use private and small-sized learning object datasets, as may be observed in the works of (Beliz 2018) and (Tsai et al. 2006). However, when we increase the number of elements in a candidate solution, GA's do not scale well and tend to create an exponential search space. Another notable drawback of this technique is related to the high number of evaluations of the fitness function that is typically required (Goldberg 2013). These aspects, when combined, make its usage impracticable in real-world scenarios due to the highly cost processing and time required.

In this paper we present and evaluate a heuristic to solve the LORP, initially proposed in Beliz (2018). In order to validate it, the heuristic proposed in this work has been compared to a classical integer linear programming formulation for the same problem. In our experiment, two dimensions are measured and assessed: performance and optimality. The results show that while displaying a small mean absolute error when compared to the classical approach, its performance is considerably superior. The heuristic may also be easily parallelize or distributed. Furthermore, in this work we evolved the tests applying the same heuristic and comparing with results obtained with the use of genetic algorithms. Results have shown that the proposed approach surpass other approaches to solve LORP, indicating that is a promising approach.

The remainder of this paper is organized as follows: “[Related works](#)” section presents some important related researches on the recommendation of LOs. “[Learning objects recommendation architecture: an overview](#)” section presents an overview of our SLE architecture, in which the proposed approach is being integrated, discussing some important pedagogical and technological aspects. “[Proposed approach](#)” section formally introduces the learning object recommendation problem and describes our approach to efficiently solve it. “[Experiments and analysis of results](#)” section presents our methods and materials, and presents experimental analysis and results. Finally, “[Conclusion](#)” section concludes our work.

Related works

In Dorça et al. (2016), authors considered the learning styles model specified by *Felder & Silverman* (Felder et al. 1988) to delimit students characteristics and recommend specific learning objects that meet each of these characteristics. There are several ways of automatically inferring the LS of a student. In this work, authors employed a methodology in which the LS are treated as probabilities, not as certainties. In this way, to start the LS

inference, a questionnaire can be answered. Otherwise, if it fails, the system starts with chance of 50% for each LS. In this context, the approach defines which IEEE LOM standard fields allow the classification of a learning object according to each delimited LS. Finally, using student interactivity data and specific rules, a specialist system was created to recommend LO.

In Mendes et al. (2017) authors proposed a technique for clustering LOs with similar characteristics, considering LS. From this grouping schema it recommends content for a student according to their corresponding LS. In this work, the LOs are made available in a repository where a vectorization process is executed (feature extraction). These vectors are created based on LO attributes extracted from the evaluation of the rules satisfied by the system proposed by Mendes et al. (2017). Afterwards these vectors are processed, using standard clustering algorithms, making recommendation process easier and lower costly.

Another work that considers LOs recommendation was proposed by Christalin et al. (2017). This work took into account three important characteristics for recommending contents: the level of compatibility of LOs concerning the students LS, the level of complexity of LOs given the level of learners knowledge and the level of interactivity and satisfaction of the students. A Compatible Genetic Algorithm (CGA) is applied when a student has the same LS as a user already trained by the system. In this technique, a modified genetic algorithm is used to reduce the search space, filling it with better chromosomes.

Additionally, in Hwang et al. (2010) authors argue that the learning happens sequentially. In a universe of several LOs a student must go through a learning path, passing through LOs with different degrees of relevance and similarity. Thus, a heuristic is created to determine personalized learning path. The optimization algorithm consists of two steps: determine the relevance between each pair of learning objects, and find the quality learning paths for individuals based on the relevant information. In this way, a learning path can be created to guide the student.

In Gaeta et al. (2013), authors used ontologies to model knowledge domains and a greedy algorithm to LOs recommendation process. In Júnior et al. (2012), in turn, authors considered ontologies to represent students' static and dynamic profiles, which are combined with software agents and an Genetic Algorithms for the recommendation of LOs in an autonomous way.

In Menolli et al. (2012) authors used text mining techniques on wiki pages to generate some metadata, which are reviewed by a human specialist, who can create metadata manually. Authors did not use ontologies or other semantic web technologies, which makes the work much harder and high costly.

In relation to Wikipedia, there are many works that use it to support learning systems. An interesting example is the approach presented in Limongelli et al. (2015), in which the authors developed a system module to assist teachers in building courses that contain only Wikipedia pages. The page recommendation process considers filtering by content and social characteristics, where the teacher can benefit from the choices already made by other users who have a similar teaching style.

In Boticario et al. (2012) authors proposed a framework for content recommendation based on standards and interoperable components that are integrated through a service-oriented architecture. The student model stores personal information,

preferences, learning styles according to the FLSM, goals and competencies, progress and indicators about attention, memory and time management. However, their work is very much related to what is proposed here, and some differences can be highlighted. It does not consider hierarchical LOs with different granularity levels. Also, they are not dynamically produced in a ubiquitous learning environment, and they are not classified according to LS dimensions based on their metadata, as proposed in this work.

In these works, the concept of LO together with the representation of standard metadata is not always explored, nor its enrichment and refinement. Considering this, the approach proposed here aims to use the information produced by a ubiquitous learning environment to make the web-based adaptive learning process more effective, as presented in the following section.

Furthermore, considering that dynamic personalization of learning content has been an important research topic that comes up with intelligent techniques aimed at adapting content to students real needs, this work has the main objective of providing a more personalised and individualised learning experience. Considering this, this work uses a student model that adequately describes and monitors the students' cognitive state, considering knowledge and behaviour.

To this end, our approach considers a hybrid student model that combines an ontology and a Bayesian network dedicated to identify students' knowledge based on their characteristics and behavior during the learning process, as described in Araújo et al. (2018b).

Although student modeling is not the specific interest of this work, the implemented student modeling approach has opened several new possibilities, as the advances and results presented in this work.

Considering the papers cited in this section, the advances of this work are related to dynamic analysis and extraction of content from the web in order to dynamically create new LOs during the learning process, when no suitable LOs are found in specific repositories. In addition, in this paper we propose a heuristic which has as main objective to find a subset of LOs that covers all the concepts to be learned by the students with a minimal cost, as explained in "[Learning objects recommendation architecture: an overview](#)" section. This is a Linear Programming problem that is known in literature as the *Set Covering problem*. The proposed approach has brought interesting advances and promising results when compared to other approaches, as presented in "[Experiments and analysis of results](#)" section.

Learning objects recommendation architecture: an overview

The main features of our approach are:

- Proposing of an ontology for modeling LOs according to IEEE-LOM standard (RISK 2002) and student modeling, whose LS are stored according to the FLSM. The choice of this model is due to the fact that it covers more psychological aspects than other models (Deborah et al. 2014);
- Automatic creation of LOs based on a ubiquitous learning system for multimedia capture and dynamic search and use of Wikipedia content considering that LOs authoring is an arduous and time-consuming task, both in terms of design and

preparation of the content itself, as well as the process of filling out their associated metadata;

- An efficient heuristic for automatic recommendation of LOs considering their quality, students learning styles and some other characteristics, as knowledge. This is an Integer Linear Programming problem known in the literature as the Set Coverage Problem (PCC), belonging to the NP-Hard class, as depicted later.

In this work, it is proposed a learning object recommendation approach that uses ontologies to represent student data and to store the LOs. Each LO is associated to the search terms that the user types in the system. According to (Gruber 1993), an ontology is an explicit specification of a shared conceptualization. Therefore, in this work, it is applied to computationally represent students information such as personal information, contextual information, personal preferences, online interactions, cognitive information, and knowledge information. In addition to this information it is also possible to store user chosen search parameters (ideal LO), permanent learning objects, suggested learning objects and temporary learning objects.

When a student searches for content, it is the representation of a concept they want to learn. These concepts will be provided by the LOs that are stored in the ontology. If the LOs do not exist yet, a search will be done in Wikipedia, in order to try to obtain an adequate content. The permanent LO is the information that is already stored in the ontology, the suggested LO are the objects that the system returns after the student's search, and the temporary LO are the contents extracted from Wikipedia.

After the extraction from Wikipedia the objects are categorized using the IEEE-LOM standard (RISK 2002) and extensions such as the Customized Learning Experience Online (CLEO) (Taliesin 2003). The LO metadata contain the following fields:

- 1 Title: Name given to a LO;
- 2 Description: Description of the LO content;
- 3 Keywords: Describes the topic of the LO;
- 4 Type of interactivity: Predominant LO learning mode, which can be active, expositive or mixed;
- 5 Type of learning resource (CLEO Extension): Corresponds to the predominant type(s) of LO, such as an exercise;
- 6 Level of interactivity: It represents the degree of interactivity of the LO to the learner, being able to assume the values very low, low, medium, high or very high;
- 7 Semantic density: Refers to the number of concepts quoted per unit of time or content presentation time, which can be very low, low, medium, high or very high;
- 8 Difficulty level: It is linked to the level of difficulty of the LO for the target audience, which can be very easy, easy, medium, difficult or very difficult.

Once an object has been extracted from Wikipedia and subsequently categorized, if appropriate for the intended concept, it can become a permanent LO, inserting the new objects into the ontology. The LO can then be reused in a new search. The Learning Resource Type field from IEEE-LOM is extended here with metadata vocabulary provided by CLEO (Customized Learning Experience Online). This provide more flexibility and opens up new possibilities in the recommendation process, allowing to provide personalized content to a large amount of different students profiles. The previous work has mapped a subset of IEEE-LOM fields that can define different dimensions of the FLSM

and proposed an automatic and dynamic approach to classify LOs (Dorça et al. 2016). Rules were developed to estimate the relationships between the FLSM and IEEE-LOM fields.

The CLEO extended vocabulary for the Learning Resource Type field has also been mapped out to different dimensions of the FLSM in order to allow personalized recommendation considering specific LS. Analogy, example, non example, practice and scenario represent CLEO's learning resource types that best suit sensing learners because they tend to be facts-oriented and they prefer concrete and practical content.

Intuitive learners prefer abstract content, such as definitions, demonstrations, glossaries, and guidelines. Reflective learners prefer abstract content in which they can learn through observations in passive way. LOs containing analogies, definitions, demonstrations, examples, non examples, glossaries, notes, presentation, scenarios or summary tend to be preferred by such kind of learners. On the other hand, assessments and their items, community resources and practical content are LOs that directly induces productive action, and therefore they are better accepted by active learners.

Visual learners prefer LOs that address information in a visual way, such as illustrations and items that attract their attention. Textual or oral content are preferred by verbal learners. Assessments, community resources (such as chats), definitions, demonstrations, glossaries, introductions, notes, and objectives are more appropriate for them.

It is known that sequential learners prefer to learn in a progressive and linear way. A guidance content is important for this kind of learner. On the other hand, global learners need an overview of the content and then make connections between the topics. LOs that contain outlines, overviews or summaries are suitable for global learners.

In addition, in this proposal a Ubiquitous Learning Environment (ULE), previously presented in Araújo et al. (2017b, 2018b), records activities in an instrumented classroom, as shown in Fig. 1, with an electronic whiteboard, projectors, microphones and video-cameras. In such scenario, various multimedia artifacts are generated in a single capture session. Therefore, those environments have the potential to generate a huge amount of LOs that could become useless if students cannot easily find what they need. In this way, the main goal of the approach presented in this paper is to fine tune the recommendation process of an existing computational architecture for personalizing and recommending educational content.

Recording lectures for later retrieval, multimedia authoring and improving learning through the Web have become increasingly used to enhance the interactivity between instructors and students. In this context, LOs have become an important tool to



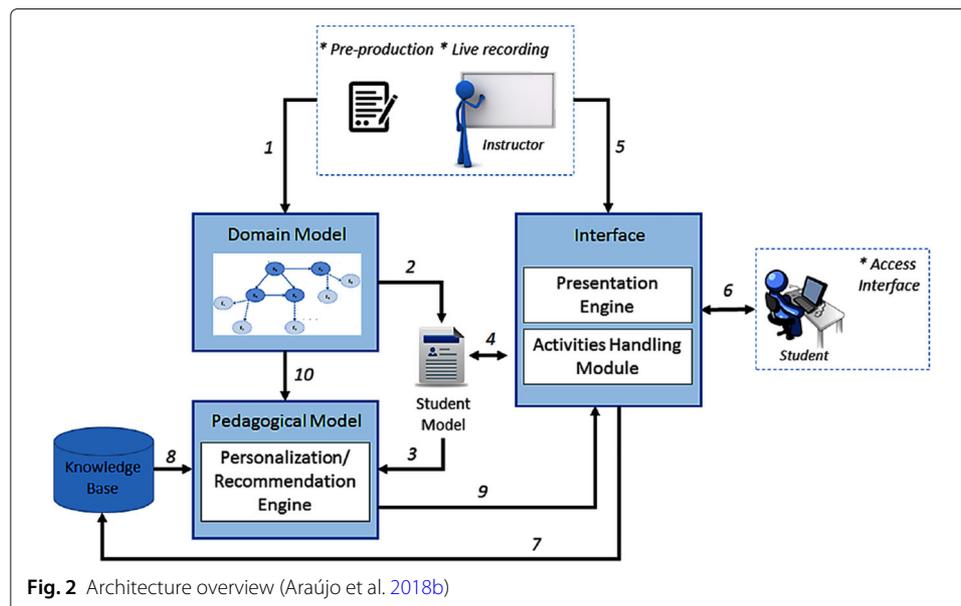
restructure pedagogical practices and improve learning experiences in advanced learning environments.

This system can assume a role of a LO generator, including its metadata. It has already more than 600 lectures captured and thousands of LOs generated. Its ubiquity happens at two different moments: when lectures are captured by computational devices in side a classroom and generates LOs; and when students access this content through a contextualized interface. Eventually, this repository tends to become larger and larger. In this context, strategies associated with smart data analysis and machine learning techniques for organizing LOs repositories are very welcome. Considering this, studies for clustering LOs based on LS have been conducted and can be found in Dorça et al. (2017) and Mendes et al. (2017).

For searching and recovering LOs efficiently, their metadata fields must be filled out with the largest possible amount of consistent values, which is a high costly process. The manual filling of metadata requires much time and effort from tutors, and may lead to mistakes and errors, resulting in inconsistent metadata. In this context, multimedia capture systems can support the semi-automatic filling of LOs metadata, minimizing the time and effort spent on their creation and avoiding erroneous input, resulting in more consistent metadata.

In this context, Fig. 2 presents an architecture overview of this system. It combines Capture and Access concepts (a ubiquitous computing subarea) together with Intelligent Tutoring Systems (ITS) concepts for providing personalized content to students (Araújo et al. 2018b).

Following, a brief description of this architecture is presented. The domain model manages information concerning the application domain, which is related to the offered courses. Once LOs and their metadata are available, students' characteristics need to be gathered in order to customize the content individually. Such characteristics should be stored in a student model capable of representing students' behavior and skills. An



ontology has been created with the purpose of representing students' characteristics and running inferences about their performance.

As a default cognitive feature, the architecture implements a structure to store the Learning Style (LS) of students and LOs according to the Felder-Silverman Learning Styles Model (FSLSM). The storage structure for this information is composed of eight numeric values from 0 to 100 (two for each dimension) representing the tendency of preference for each one of the learning styles dimensions by students, which gives a probabilistic nature to this information and indicates that students do not have a fixed preference for an LS. As a consequence, students preference for a learning style can evolve over time, as presented in Dorça et al. (2013).

This architecture includes a structure that allows LOs to be classified according to the FSLSM, which is used to recommend educational content based on students' LS. This structure consists of eight numeric values – two for each dimension – that represent the probability (or tendency) of the respective LO comply with each LS.

Once defined both LOs and the way students are represented, it is possible to design algorithms able to choose and recommend LOs that best fits students' needs into a given context. Considering Fig. 2, the Pedagogical Model, also called tutoring module, is responsible to select the best content or learning activity at a given time. It uses LOs meta-data stored in the Knowledge Base and information from the Student Model to create strategies to present LOs in an individualized way. In this context, it must use intelligent mechanisms to guide students in their learning path considering individual needs aiming at maximizing their performance. Such strategies also consider information related to the course design defined in the Domain Model. This is a complex problem depicted in next section, considering our LORP solving approach. Curriculum sequencing, interactive problem-solving support, and intelligent analysis of student solutions are the traditional intelligent techniques applied in ITS.

Our approach establish connections between the knowledge space stated by the Domain Model and the hyperspace, considering Wikipedia sections for example, as shown in Fig. 3. To be able to create such connections, the domain knowledge and the educational content must be well-structured. It could be compared to the Domain model of traditional ITS, which comprises a set of small domain knowledge elements.

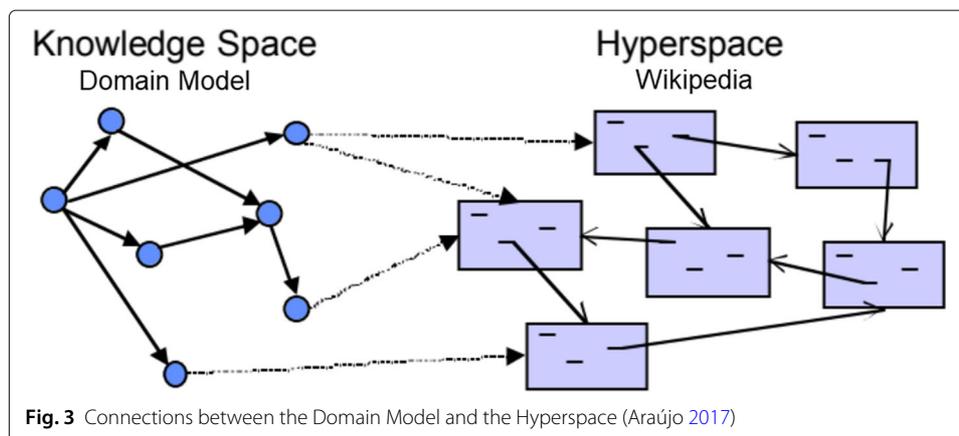


Fig. 3 Connections between the Domain Model and the Hyperspace (Araújo 2017)

The main goal of this paper is to present an approach able to identify which concepts must be presented, and which LOs must be dynamically linked to each concept, considering a particular student, and how to sequence them.

Next section describes in more detail the LOs recommendation problem, which is the main concern of this paper, and our proposed approach to solve it. The proposed solution is being integrated to the architecture presented in Fig. 2, and its implementation is being done at the Pedagogical Module, inside the Personalization/Recommendation Engine. Once this integration is done, experiments and validation with real students in real learning scenarios will be executed. For this paper, we present empirical results and analysis considering simulated scenarios. The results presented in this paper are prominent, and shows that this is a promising approach. The architecture presented in Fig. 2 have been tested considering different scenarios and techniques. Experiments for validation with real students in real different learning scenarios have been done, as can be seen in Araújo et al. (2017a, 2018a, b); Ferreira et al. (2019a, b). Following, we depict and discuss details related to our proposed approach for automatic and dynamic content recommendation.

Proposed approach

This section presents the proposal of a low complexity heuristic to solve the learning objects recommendation problem. The main goal of the proposed approach is to accurately recommend LOs for a given concept considering a specific student. Recurrently, the same LO can attend several concepts, and therefore the system has to choose the best LOs to a specific concept. To illustrate the problem, consider a situation where a student needs to learn five concepts, which belong to the finite set $X = \{C_1, C_2, C_3, C_4, C_5\}$ considering a collection of subsets of X given by $F = \{LO_1, LO_2, LO_3, LO_4, LO_5, LO_6\}$, where $LO_1 = \{C_1, C_2\}$, $LO_2 = \{C_3\}$, $LO_3 = \{C_4\}$, $LO_4 = \{C_5\}$, $LO_5 = \{C_2, C_3, C_4, C_5\}$, $LO_6 = \{C_2, C_3\}$. Each LO of F has an assigned a cost, respectively, given by the cost vector $(3, 7, 1, 1, 4, 2)$.

In this approach the cost represents how appropriate the object is to meet the student's personal preferences. A LO can carry the necessary learning content, however it may not be the most suitable to meet the student's learning style for example. The higher the cost of an object, the less suitable it is to meet those preferences.

The goal is to find a subset of LOs of F that covers all the elements (concepts) of X with a minimal cost. This is a Linear Programming problem that is known in literature as the *Set Covering problem*. It is an NP-Hard problem and, for the aforementioned problematic situation, we have the following solution $S = \{LO_1, LO_3, LO_6\}$.

Given a matrix a_{ij} with m rows and n columns, Set Covering is a problem of line coverage by a set of columns with minimal cost. Let $x_j = 1$, if the column j (with cost $c_j > 0$) is in the solution, and $x_j = 0$ otherwise. The Set Covering formulation is given by the Eqs. 1 and 2:

$$\text{Minimize } \sum_{j=1}^n c_j x_j \quad (1)$$

$$\begin{aligned} \text{Subject to } & \sum_{j=1}^n a_{ij} x_j \geq 1, \quad i = 1, \dots, m, \\ & x_j \in \{0, 1\}. \end{aligned} \quad (2)$$

The restrictions of Eq. 2 ensures that each line is covered by at least one column. Notice that the columns of the matrix correspond to the LOs and the rows to the concepts. The cost of choosing an LO_j depends on how close this LO is to the ideal. The LO_{ideal} has a quality parameter and other user-defined parameters of the tool that implement the approach. The other LOs, which have these same parameters, are compared with the LO_{ideal} . Formally, let α_i be the value of the parameter i . The function c_j of Eq. 1 is calculated for each pair of LO_{ideal} parameters and a LO_j :

$$c_j = \sum_{i=1}^x (|\alpha_{i(ideal)} - \alpha_{i(j)}|) \quad (3)$$

where x is the number of parameters, $\alpha_{i(ideal)}$ is a parameter of LO_{ideal} and $\alpha_{i(j)}$ is a parameter of LO_j . Thus, the smaller the difference between the analyzed LO and the Ideal LO, the lower the cost. The total cost is given by the sum of the differences between the Ideal LO and analyzed LO.

To solve this LORP, previous tests were performed in a set of 200 LO's being 114 retrieved from Wikipedia articles and other 86 were randomly created respecting the metadata needed to characterize them as learning objects obtained from Wikipedia. Eighty six were randomly created respecting the metadata needed to characterize them as learning objects obtained from Wikipedia. Out of 200 objects created 190 serve one single concept and 10 of them serve two.

To find an appropriate set and sequence of LOs to be recommended to students, it has been used a genetic algorithm, obtaining good recommendation results. However, this type of algorithm becomes impracticable considering a real scenario, when the search universe is huge, like our current LOs repository. As the student will be waiting for an LO when he is accessing the AVA, it is desirable that the response time is almost real time, otherwise it can cause impatience or even cause the student to give up on the platform.

The approach using a Genetic Algorithm find appropriate LOs to solve a problem set but it spend impracticable response times in a search space of 2000 LO's as presented in Table 2. It is worth mentioning that the number of LO's retrieved from the web may be much greater that this. So, in this work it is proposed a heuristic to solve this problem, taking into account much larger search spaces.

The core intuition of this heuristic is that starting from a list of required concepts, the LOs that matches the learning style of an apprentice while displaying a greater concept coverage tend to offer better candidates for a final solution. The proposed heuristic is described by Algorithm 1.

As input, the algorithm receives a list of required concepts R , the user's learning style U and also the list of learning objects L that in turn, contain at least one required concept in R . In the first part of the algorithm (Rank), we iterate over the elements of L calculating its rank. According to the line 3, the attribution cost of each LO x to the user U ($Cost = |\alpha_x - \alpha_U|$) is inversely amplified by the proportion of coverage of x with respect to R ($Count =$ the quantity of concepts of x that are present in R). The line 4 perform a sorted insertion of the element x into the list F , using the new property x_{rank} as the sorting key.

The second part (Selection) regards selecting LO's that will compose the solution. In this stage, (line 10) we iterate over the sorted set F , and whenever we found an LO x that adds a new required concept to the temporary concepts list t , we add it to the solution S . The algorithm stops (line 19) when the temporary concepts list t contains the same

Algorithm 1: The proposed heuristic	
	Data: L - LO's which covers at least one required concept
	Data: R - The list of required concepts
	Data: U - The learner learning style
	Result: A possible solution
	$F \leftarrow \emptyset$
	for $x \in L$ do
3	$x_{rank} \leftarrow \text{Cost}(x, U) \left(2 - \frac{\text{Count}(x_{concepts}, R)}{\text{Length}(R)}\right)$
4	InsertSort(F, x, x_{rank})
	end
	$t \leftarrow \emptyset$
	$S \leftarrow \emptyset$
	for $x \in F$ do
	$included \leftarrow \text{False}$
10	for $x_{concept} \in x_{concepts}$ do
	if not $x_{concept}$ in t and $x_{concept}$ in R then
	AddItem($x_{concept}, t$)
	$included \leftarrow \text{True}$
	end
	end
16	if $included$ is True then
	AddItem(x, S)
	end
19	if Length(t) = Length(R) then
	break
	end
	return S
	end

number of elements of the required concepts R . Notice that in practice, the algorithm tends to scan only the head of the ordered set F what improves its performance. The total cost of the final solution is given by Eq. 3 applied over the solution set S .

The proposed approach has an order of complexity close to $O(n)$ such that n is the number of LO's in L . Another important aspect of this approach is that it may be easily parallelized or distributed across multiple machines in a cluster. Its first step - when one compute the LO rank - calculations are performed independently from each other and, therefore, fits the *mapping* stage in an event stream.

Experiments and analysis of results

In our experiment, the proposed heuristic was evaluated over two dimensions: Performance level (time spent in milliseconds) and optimality. We compared its results with those achieved by a mixed-integer linear programming formulation, referred to in this work as the classical method. In this approach, if the problem formulation satisfies the requirements (convex and nonempty constraint region), then it is guaranteed to find an

optimal solution in one of its vertices. Thus, comparing the proposed heuristic to an optimal method enabled us to verify its suboptimality level safely.

In addition, at this time we implemented the a genetic algorithm using tournament selection, two-point crossover, one-flip bit mutation operator and elitism strategy. The genetic algorithm was compared to the proposed heuristics and the exact method. Unlike the approach considered in the work analyzed (Beliz 2018), we try to use the proposal of genetic algorithms in much larger search spaces, as proposed by this approach.

Both approaches were implemented in *Python* language. While the heuristic and genetic algorithm employed only pure *Python*, the mathematical programming approach was modeled using the *OR-Tools* library (Google 2018) which is an open source framework for modelling optimization problems such as linear optimization and constraint programming. The formulation was identical to the set covering problem described by Eqs. 1, 2, and 3. The solver, in this case, was the integer linear programming (MILP) *GLPK* (Makhorin 2008) which is a well-known tool for solving large-scale set covering problems.

We opted to create nine different LO datasets containing, respectively, 2k, 4k, 8k, 16k, 32k, 64k, 120k, 340k, 680k, and 1.360M registers. The generated LOs exhibit the same characteristics presented by *Belizário* (Beliz 2018). Each of them cover at least one and at most twelve concepts from a given universe of 50 possible concepts. Moreover, a random quality and learning style are assigned following the classification proposed by *Felder and Silverman* (Felder et al. 1988).

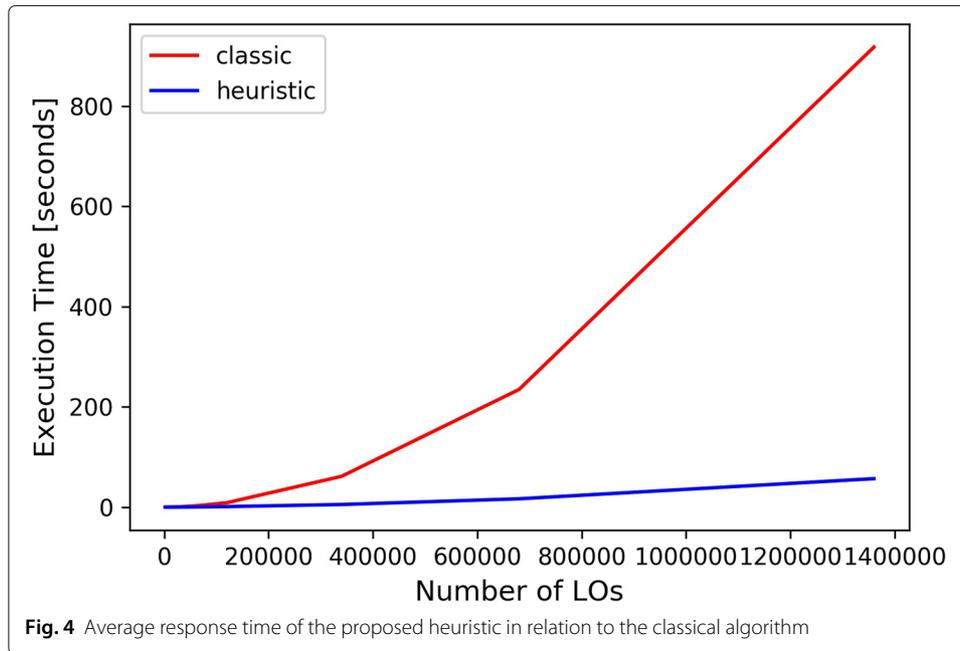
Six separated problem sets were also generated. They represent possible user queries and each of them contains a number of required concepts to which the system must find an optimal solution. Table 1 presents each problem and its required concepts. We also created a special problem set that was intended to capture situations where the system must return the whole content of a fictional course. This special case contained 70 randomly chosen concepts taken from the same universe of concepts described above.

For each dataset we executed all problem sets. At each experiment, the approach has to find the best LO subset that covers the set of concepts described on the problem set. We measure the approach performance with respect to its response time in milliseconds. Optimality is observed through the Mean Absolute Error between classical approach and the proposed heuristic ($MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$).

To validate the efficiency of the proposed approach, two items were evaluated. The first one concerns the response time of the heuristic in relation to the classical algorithm (see Fig. 4). Notice that the classical linear programming algorithm have an exponential behavior as the number of constraints (number of LOs to be evaluated) increases.

Table 1 Problem sets

Problem set	Required concepts
PS_1	$C_1, C_3, C_{12}, C_{17}, C_{24}$
PS_2	$C_{19}, C_{22}, C_{33}, C_{37}, C_{12}, C_{21}$
PS_3	C_{61}, C_{14}
PS_4	C_{13}, C_{14}, C_{15}
PS_5	$C_3, C_{21}, C_{33}, C_{45}, C_{47}, C_{52},$ $C_{55}, C_{17}, C_{18}, C_{54}, C_9, C_{27}, C_{69}$
PS_6	C_{22}



The genetic algorithm proposed by *Belizário*, despite being efficient in the context presented by him, proved to be inefficient when the search space is large as the situation proposed in this work. For a search space of 2000 learning objects (the smallest search space presented by this work) the execution time and cost were much higher than the other approaches as can be seen in the Table 2.

Over 2000 learning objects the use of the genetic algorithm becomes impractical making its comparison with other algorithms in large search spaces impossible.

As expected, the behavior of the classical algorithm and heuristic tested showed to be similar at 16k learning objects. From this point on, it is observed that the classical approach exponentially increases its average response time as the volume of LO's increases. At 64k, one might notice that the difference is already in several orders of magnitude. The proposed heuristic on the other hand was linear, with a predictable response time variation. Another important aspect the heuristic is that at the selection stage (Algorithm 1), the algorithm scanned, on average, only the first eight elements of the ranked list to generate its solutions.

Table 3 shows the response time of the proposed heuristic in each set of learning objects in each of the six problem sets analyzed. The table also shows the average response time

Table 2 Comparison in cost and time (seconds) of the 3 approaches

Problem set	Classical algorithm		Proposed heuristic		Genetic algorithm	
	Cost	Time	Cost	Time	Cost	Time
PS ₁	3	13.25x10 ⁻³	3	2.47x10 ⁻³	21	425.67
PS ₂	3	19.44x10 ⁻³	4	3.38x10 ⁻³	16	451.48
PS ₃	1	9.89x10 ⁻³	1	2.49x10 ⁻³	5	237.87
PS ₄	2	14.18x10 ⁻³	2	3.30x10 ⁻³	9	296.59
PS ₅	5	25.50x10 ⁻³	9	4.38x10 ⁻³	33	679.94
PS ₆	1	6.18x10 ⁻³	1	2.48x10 ⁻³	2	86.49

Table 3 Response time of the proposed heuristic for each set of LO

LO set	Time (seconds)						Average Time
	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	
2000	0.008	0.009	0.005	0.006	0.017	0.004	0.008
4000	0.02	0.02	0.01	0.01	0.04	0.01	0.017
8000	0.04	0.04	0.02	0.03	0.08	0.02	0.0389
16000	0.09	0.09	0.05	0.07	0.18	0.03	0.086
32000	0.23	0.22	0.11	0.14	0.46	0.08	0.205
64000	0.50	0.52	0.26	0.33	1.17	0.16	0.49
120000	1.16	1.22	0.52	0.75	2.92	0.35	1.15
340000	5.34	5.41	2.00	3.0	15.05	1.12	5.32
680000	16.31	16.00	5.30	8.6	51.46	2.63	16.71
1360000	54.73	53.83	15.55	25.93	184.53	6.73	56.88

in each LO set. Table 4 presents the response time values of the classic algorithm as well as the mean response time in each problem analyzed in a given LO set.

For all problem sets, considering the variable response time, the heuristics obtained the shortest average times. The classical approach in turn, obtained larger values, according to the Tables 3 and 4

The second evaluation regards the optimality of the heuristic. The Table 5 presents the experimental results. One may observe that the proposed heuristic presented variations, demonstrating that, although sometimes the optimal value is not found, a good alternative is offered with a much lower computational cost. The results of the heuristic show that its performance worsens when we increase the number of concepts in an instance. Finally, it can be noticed that the two evaluated items (performance and optimality) follow the expectations, corroborating with the hypothesis of heuristic efficiency and is usable in real world scenarios, with millions of LOs to be assessed.

In this section experiments and analysis of results were presented in details. Based on simulation, the mathematical model has been properly tested, and it has proven to be more effective than other previously proposed approaches. This kind of experimental analysis considering simulated environment has proven to be very practical and useful to validate this kind of approach. It is very difficult (or impossible), and highly cost, to validate this kind of approach in real learning scenarios, considering real students, due to the need resources and time consuming.

Table 4 Response time of the classic algorithm for each set of LO

LO set	Time (seconds)						Average Time
	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	
2000	0.03	0.04	0.02	0.03	0.08	0.02	0.04
4000	0.06	0.06	0.03	0.05	0.14	0.021	0.06
8000	0.14	0.13	0.07	0.10	0.32	0.046	0.14
16000	0.36	0.38	0.16	0.25	0.74	0.11	0.33
32000	0.99	1.12	0.50	0.67	1.97	0.31	0.93
64000	3.51	3.19	1.66	2.20	6.12	0.97	2.94
120000	10.68	10.44	4.84	6.70	17.53	2.94	8.86
340000	74.78	74.17	35.38	47.20	117.45	20.11	61.52
680000	284.29	285.79	132.87	183.54	444.50	77.95	234.82
1360000	1103.89	1124.32	534.63	711.28	1727.24	305.41	917.79

Table 5 Accuracy of the heuristic in relation to the classical algorithm for each set of LO

	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	MAE
2000	1	1	0	0	2	0	0.67
4000	1	1	0	0	1	0	0.50
8000	1	0	0	0	1	0	0.33
16000	1	0	0	0	3	0	0.67
32000	1	0	0	0	3	0	0.67
64000	1	0	0	0	2	0	0.50
120000	0	0	0	0	1	0	0.17
340000	0	0	0	0	3	0	0.50
680000	0	0	0	0	3	0	0.50
1360000	0	0	0	0	3	0	0.50

Simulated experimental analysis of mathematical models and algorithms like this are important and valuable tool to allow the advancing of the state-of-the-art in our research field. Thus, our next step is to use the proposed approach, already validated, to perform controlled experiments with large groups of students in a real learning scenarios, evaluating other aspects related to the quality and efficiency of the proposed approach.

Conclusion

According to the obtained results, it can be concluded that the approach proposed in this work was superior to the method that uses a classical mixed-integer linear programming formulation and the genetic algorithm to solve the set covering problem of learning objects recommendation. It presented an almost linear time to find the solution to the proposed problems. In addition, its accuracy proved to be close to that of the classical approach, validating the hypothesis that the proposed heuristic is feasible to solve this type of problem.

In this way, as real-world learning objects repositories exhibit a large number of records, the proposed approach becomes particularly relevant. It may find an adequate set of learning objects faster than classical algorithms. Moreover, techniques such based on evolutionary algorithms are not able to achieve the same performance, highlighting the importance of new complementary studies to this work.

As future works, our next step is to use the proposed approach, already validated, to perform controlled experiments with large groups of students in a real learning scenarios, evaluating other aspects related to the quality and efficiency of the proposed approach. In this case, control groups and experimental groups should be tested and statistically validated considering different aspects.

Acknowledgements

No one but the authors was involved with this work

Authors' contributions

Samuel Henrique Falci and Daniel Henrique Mourão Falci conducted the research and extracted data from other works for comparison. He also participated in the implementation of all the algorithms and performed the tests. Fabiano Azevedo Dorça and Alessandro Vivas Andrade guided the research stipulating their methods and evaluated the results. The author(s) read and approved the final manuscript.

Funding

Not Applicable.

Availability of data and materials

All the data used in this work were computationally simulated and its replication can be easily done through information contained in the paper itself.

Competing interests

The authors declare that they have no competing interests.

Author details

¹Universidade Federal de Uberlândia, Avenue João Naves de Ávila, 2121, Uberlândia, MG, Brazil. ²Universidade Federal dos Vales do Jequitinhonha e Mucuri, Road MGT 367, km 583, 5000, Diamantina, MG, Brazil. ³Universidade Fumec, Street Cobre, 200, Belo Horizonte, MG, Brazil.

Received: 3 April 2020 Accepted: 25 August 2020

Published online: 11 September 2020

References

- Araújo, R.D. (2017). Uma Arquitetura Computacional para Autoria e Personalização de Objetos de Aprendizagem em Ambientes Educacionais Ubíquos, PhD thesis, Universidade Federal de Uberlândia.
- Araújo, R., Brant-Ribeiro, T., Ferreira, H., Dorça, F., Cattelan, R. (2018a). A semiautomatic and probabilistic approach for student modeling in ubiquitous learning environments, In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, vol. 29 (p. 1313). Ceará: Sociedade Brasileira de Computação - SBC.
- Araújo, R.D., Brant-Ribeiro, T., Mendonça, I.E., Mendes, M.M., Dorça, F.A., Cattelan, R.G. (2017a). Social and collaborative interactions for educational content enrichment in ules. *Journal of Educational Technology & Society*, 20(3), 133–144.
- Araújo, R.D., Cattelan, R.G., Dorça, F.A. (2017b). Towards an adaptive and ubiquitous learning architecture, In *2017 IEEE 17th International Conference on Advanced Learning Technologies (ICALT)* (pp. 539–541). Taiwan: IEEE.
- Araújo, R.D., Ferreira, H.N., Cattelan, R.G., Dorça, F.A. (2018b). A hybrid architecture for adaptive, intelligent, and ubiquitous educational systems, In *Digital Technologies and Instructional Design for Personalized Learning* (pp. 120–144). University of Utah: IGI Global.
- Beliz, C.F. (2018). Reúso de conteúdo da Web na Recomendação Personalizada de Objetos de Aprendizagem: uma abordagem baseada em um Algoritmo Genético, Tecnologias da Web Semântica e uma Ontologia. PhD thesis, Universidade Federal de Uberlândia.
- Boticario, J.G., Rodriguez-Ascaso, A., Santos, O.C., Raffenne, E., Montandon, L., Roldán Martínez, D., Buendía García, F. (2012). Accessible lifelong learning at higher education: outcomes and lessons learned at two different pilotsites in the eu4all project. *Journal of Universal Computer Science*, 18(1), 62–85.
- Chen, N.-S., Cheng, I.-L., Chew, S.W., et al (2016). Evolution is not enough: Revolutionizing current learning environments to smart learning environments. *International Journal of Artificial Intelligence in Education*, 26(2), 561–581.
- Christalin, B., Kirubakaran, E., Jeba Thangaiyah, R. (2017). An evolutionary approach for personalization of content delivery in e-learning systems based on learner behavior forcing compatibility of learning materials. *Telematics and Informatics*, 35. <https://doi.org/10.1016/j.tele.2017.02.004>.
- Crockett, K., Latham, A., Mclean, D., Bandar, Z., O'Shea, J. (2011). On predicting learning styles in conversational intelligent tutoring systems using fuzzy classification trees, In *Fuzzy Systems (FUZZ), 2011 IEEE International Conference On* (pp. 2481–2488). Taiwan: IEEE.
- Deborah, L.J., Baskaran, R., Kannan, A. (2014). Learning styles assessment and theoretical origin in an e-learning scenario: a survey. *Artificial Intelligence Review*, 42(4), 801–819.
- Dorça, F., Araújo, R., Carvalho, V., T. Resende, D., Cattelan, R. (2016). An automatic and dynamic approach for personalized recommendation of learning objects considering students learning styles: An experimental analysis. *Informatics in Education*, 15, 45–62. <https://doi.org/10.15388/infedu.2016.03>.
- Dorça, F.A., Carvalho, V.C., Mendes, M.M., Araújo, R.D., Ferreira, H.N., Cattelan, R.G. (2017). An approach for automatic and dynamic analysis of learning objects repositories through ontologies and data mining techniques for supporting personalized recommendation of content in adaptive and intelligent educational systems, In *2017 IEEE 17th International Conference on Advanced Learning Technologies (ICALT)* (pp. 514–516). New York City: IEEE.
- Dorça, F.A., Lima, L.V., Fernandes, M.A., Lopes, C.R. (2013). Comparing strategies for modeling students learning styles through reinforcement learning in adaptive and intelligent educational systems: An experimental analysis. *Expert Systems with Applications*, 40(6), 2092–2101.
- Felder, R.M., Silverman, L.K., et al (1988). Learning and teaching styles in engineering education. *Engineering education*, 78(7), 674–681.
- Ferreira, H., Araújo, R.D., Dorça, F., Cattelan, R. (2016). Uma Abordagem Híbrida para Acompanhamento da Aprendizagem do Estudante Baseada em Ontologias e Redes Bayesianas em Sistemas Adaptativos para Educação, PhD thesis, Universidade Federal de Uberlândia. <https://doi.org/10.5753/cbie.wcbie.2016.447>. <http://br-ie.org/pub/index.php/wcbie/article/view/6966>.
- Ferreira, H., de Oliveira, G.P., Araújo, R., Dorça, F., Cattelan, R. (2019a). Technology-enhanced assessment visualization for smart learning environments. *Smart Learning Environments*, 6(1), 14.
- Ferreira, H., Oliveira, G., Araújo, R., Dorça, F., Cattelan, R. (2019b). An open model for student assessment visualization, In *2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT)*, vol. 2161 (pp. 375–379): IEEE.
- Gaeta, M., Miranda, S., Orciuoli, F., Paolozzi, S., Poce, A. (2013). An approach to personalized e-learning. *Journal of Education, Informatics & Cybernetics*, 11(1), 15–21.
- Goldberg, D.E. (2013). *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*, (p. 272). US: Springer.
- Google, I. (2018). OR-Tools - Google Optimization Tools. *GitHub*.
- Graf, S., Liu, T.-C., et al. (2008). Identifying learning styles in learning management systems by using indications from students' behaviour, In *Advanced Learning Technologies, 2008. ICALT'08. Eighth IEEE International Conference On* (pp. 482–486). New York City: IEEE.
- Gruber, T.R. (1993). A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2), 199–220.
- Hwang, G.-J., Kuo, F.-R., Yin, P.-Y., Chuang, K.-H. (2010). A heuristic algorithm for planning personalized learning paths for context-aware ubiquitous learning. *Computers & Education*, 54(2), 404–415.

- Júnior, L.J., Neto, F.M.M., da Silva, L.C.N. (2012). Uma abordagem baseada em algoritmo genético para recomendação de objetos de aprendizagem sensível ao contexto do estudante, In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, vol. 23. Rio de Janeiro: Sociedade Brasileira de Computação - SBC.
- Limongelli, C., Gasparetti, F., Sciarrone, F. (2015). Wiki course builder: a system for retrieving and sequencing didactic materials from wikipedia, In *2015 International Conference on Information Technology Based Higher Education and Training (ITHET)* (pp. 1–6). New York City: IEEE.
- Makhorin, A. (2008). Glpk (gnu linear programming kit). <http://www.gnu.org/s/glpk/glpk.html>. Accessed 9 Sep 2020.
- Markets (2019). Markets: Lms market by component (solution and services), delivery mode (distance learning, instructor-led training and blended learning), deployment type, user type (academic and corporate), and region - global forecast to 2023. Technical report, Markets and Markets. <https://www.marketsandmarkets.com/Market-Reports/learning-management-systems-market-1266.html>. Accessed 9 Sep 2020.
- Mendes, M., Carvalho, V., Araújo, R., Dorça, F., Cattelan, R. (2017). Clustering learning objects in the ieeee-lom standard considering learning styles to support customized recommendation systems in educational environments. <https://doi.org/10.1109/LACLO.2017.8120898>.
- Menolli, A., Malucelli, A., Reinehr, S. (2012). Criação semi-automática de objetos de aprendizagem a partir de conteúdos da wiki, In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, vol. 1. Rio de Janeiro: Sociedade Brasileira de Computação - SBC.
- RISK, U. (2002). Draft standard for learning object metadata. *IEEE standard*, 1484(1), 1–40.
- Stathacopoulou, R., Magoulas, G.D., Grigoriadou, M., Samarakou, M. (2005). Neuro-fuzzy knowledge processing in intelligent learning environments for improved student diagnosis. *Information Sciences*, 170(2–4), 273–307.
- Taliesin, B. (2003). CLEO Extensions to the IEEE Learning Object Metadata. *Version*.
- Tsai, K.H., Chiu, T.K., Lee, M.C., Wang, T.I. (2006). A learning objects recommendation model based on the preference and ontological approaches, In *Advanced Learning Technologies, 2006. Sixth International Conference On* (pp. 36–40). New York City: IEEE.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
