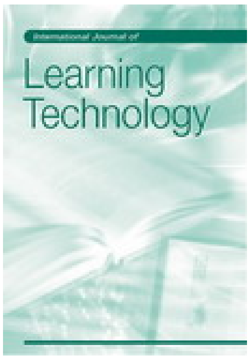


International Journal of Learning Technology

ISSN online: 1741-8119 - ISSN print: 1477-8386
https://www.inderscience.com/ijlt

Advances in personalised recommendation of learning objects based on the set covering problem using ontology

Clarivando Francisco Belizário Júnior, Fabiano Azevedo Dorça, Luciana Pereira de Assis, Alessandro Vivas Andrade

DOI: 10.1504/IJLT.2024.10063339

Article History:

Received:	10 June 2022
Last revised:	25 January 2023
Accepted:	15 March 2023
Published online:	08 April 2024

Advances in personalised recommendation of learning objects based on the set covering problem using ontology

Clarivando Francisco Belizário Júnior* and
Fabiano Azevedo Dorça

Faculty of Computer Science (FACOM),
Federal University of Uberlândia (UFU),
Campus Santa Mônica,
Bloco 1B, Sala 1B148, Av. João Naves de Ávila, 2.121,
Bairro Santa Mônica, Uberlândia, 38400-902, Minas Gerais, Brazil
Email: clarivando@ufu.br
Email: fabianodor@ufu.br
*Corresponding author

Luciana Pereira de Assis and
Alessandro Vivas Andrade

Department of Computing,
Federal University of Jequitinhonha and Mucuri Valleys (UFVJM),
Campus JK. Rodovia MGT 367 – Km 583, No. 5000,
Alto da Jacuba, Diamantina, 39100-000, Minas Gerais, Brazil
Email: lpassis@ufvjm.edu.br
Email: alessandrovivas@ufvjm.edu.br

Abstract: Loop-based intelligent tutoring systems (ITSs) support the learning process using a step-by-step problem-solving approach. A limitation of ITSs is that few contents are compatible with this approach. On the other hand, recommendation systems can recommend different types of content but ignore the fine-grained concepts typical of the step-by-step approach. This work contributes to the solution of this state-of-the-art challenge by proposing an approach for the recommendation of learning objects from different areas of knowledge, considering the refined concepts of ITSs. To deal with this challenge, we formulate the learning object recommendation problem as the set covering problem that belongs to the NP-hard class problems. An exact algorithm and a greedy heuristic were properly adapted, resulting in a promising approach to solve these problems, as shown by the results. This resulted in more personalised content for students using collaborative filtering and an ontology that models their knowledge, learning styles and search parameters.

Keywords: learning objects recommendation; personalised recommendation; collaborative filtering; ontology; set covering; learning styles; intelligent tutoring systems; ITSs.

Reference to this paper should be made as follows: Belizário Jr., C.F., Dorça, F.A., de Assis, L.P. and Andrade, A.V. (2024) 'Advances in personalised recommendation of learning objects based on the set covering problem using ontology', *Int. J. Learning Technology*, Vol. 19, No. 1, pp.25–57.

Biographical notes: Clarivando Francisco Belizário Júnior is a Doctoral student in Computer Science at the Federal University of Uberlândia (UFU), Minas Gerais, Brazil. He has completed his Master's and a degree in Computer Science and a degree in Letters (French) at the UFU. His research focuses on artificial intelligence applied to education, including the implementation of learning object recommendation systems using ontology and collaborative filtering techniques.

Fabiano Azevedo Dorça is a Professor of the Graduate Program in Computer Science at Federal University of Uberlândia. He served as a Coordinator of the Bachelor's in Computer Science at the Federal University of Uberlândia from 2013 to 2017. His research interests focus on intelligent tutoring systems, distance education, student modelling, learning styles, learning assessment, learning objects, adaptive hypermedia systems, machine learning, learning analytics, educational data mining, ontologies, semantic web, design smells and simulation.

Luciana Pereira de Assis is an Associate Professor at the Federal University of Jequitinhonha and Mucuri Valleys (UFVJM) where she teaches for the Information Systems course and the Graduate Program in Education. She has experience in optimisation, working mainly on the following topics: combinatorial optimisation, collection and delivery problems, heuristics and metaheuristics, vehicle routing problem, multiobjective optimisation, social media analysis and use of artificial intelligence and optimisation techniques in virtual learning environments.

Alessandro Vivas Andrade is an Associate Professor at the Federal University of Jequitinhonha and Mucuri Valleys (UFVJM). He works as a Professor in the Information Systems course where he teaches disciplines of Computer Networks and Distributed Systems. He has completed his Doctorate, Master's degree and Graduation in Electrical Engineering at the Federal University of Minas Gerais (UFMG) and a Technical Course in Electronics at Francisco Moreira da Costa School (ETEFMC). He is interested in the areas of machine learning, artificial intelligence and optimisation.

1 Introduction

Intelligent tutoring systems (ITSs) are computer systems that use artificial intelligence and cognitive psychology techniques to give feedback to students without human intervention (Bernacki et al., 2014). A challenge inherent in these systems is the personalised learning objects (LOs) recommendation. This problem, hereinafter referred to as LO recommendation problem (LORP), is treated in the literature by several techniques, the most used being content-based filtering (CBF) (Vanetti et al., 2010), collaborative filtering (CF) (De Medio et al., 2020) and the combination of two or more techniques (hybrid recommendation) (Burke, 2007; Barragáns-Martínez et al., 2010; Choi et al., 2012; Tarus et al., 2017). These techniques suffer from the rating sparsity (Zhao et al., 2015) and cold-start (Adomavicius and Tuzhilin, 2005) problems. In the context of e-learning, the rating sparsity problem occurs when few students have evaluated the same LO, and there is no overlap in the classification preferences. The cold-start problem occurs when it is not possible to make reliable recommendations due to the lack of initial assessments for new students or educational resources (Adomavicius and Tuzhilin, 2005).

ITSs are good at helping the students solve problems step-by-step by giving them feedback and hints on each step (VanLehn, 2006), but this approach is not compatible with most content (Soofi and Ahmed, 2019). On the other hand, recommendation systems (RSs) are able to recommend content from different areas, but they disregard the fine-grained concepts that the student needs to learn, which prevents a more personalised recommendation to the student's knowledge.

In this article, we define the LORP as a problem whose goal is to solve the previous drawbacks. We propose a hybrid recommendation approach that uses an ontology (Gruber, 1993) to model knowledge about students and educational resources, being able to recommend LOs from all areas of knowledge using fine-grained concepts, contributing to the state-of-the-art. Ontology is used by some works to model the knowledge about the students and learning resources (Bajenaru et al., 2015; Shishehchi et al., 2012; Moreno et al., 2013; Ruotsalo et al., 2013), and in our work, it is also used to model fine-grained LOs (called hints). In addition, the ontology stores the concepts that each LO covers, providing a fine-grained recommendation of LOs that cover the concepts that the student has not yet mastered, including subjects about which the student has doubts, for which hints will be recommended.

The main contributions of this work to the e-learning RSs are:

1. An approach that combines ontology-based recommendation – that reduces the rating sparsity and cold-start problems – and CF techniques for the recommendation of LOs based on concepts and the reuse of web content. The ontology models LOs and the students' knowledge level and profile, and it implements inference rules to aid the recommendation process.
2. The LORP is formalised as the set covering problem (SCP) (Garey and Johnson, 1979). We adapt two algorithms to solve this problem.

The rest of this paper is organised as follows. In Section 2, we present the background of this work. In Section 3, we discuss the related work relevant to this study. The proposed approach is detailed in Section 4. Section 5 is dedicated to the experiments and the results. Finally, the conclusions and future work are outlined in Section 6.

2 Background

This section presents the most relevant theories and concepts that ground this work. In Section 2.1, we present a summary about the semantic web (SW), a technology that uses ontologies to semantically represent the vast content of the traditional web. Ontologies structure entities and their relationships allowing the inference of new knowledge. In our approach, the ontology stores information about the students and LOs. The educational standards used to structure this information are presented in Section 2.2.

The main filtering and recommendation techniques that support RSs are presented in Section 2.3. Our approach uses the ontology-based recommendation technique, in which the domain model and the learner model are structured in an ontology. Our RS uses these models to solve the LORP. To understand the LORP as a covering problem, in Section 2.4, we describe the SCP and its usefulness in formalising real-world problems.

2.1 Semantic web

The SW (Berners-Lee et al., 2001), as the name suggests, extends the traditional web with semantic information described in eXtensible Markup Language (XML), resource definition framework (RDF) and Web Ontology Language (OWL). OWL is the language with the greatest potential for knowledge representation and is commonly used in the implementation of ontologies. Ontologies can be thought of as non-relational databases, which are consulted through queries in SPARQL, a language equivalent to SQL. A great advantage of ontologies is the possibility of discovering new knowledge through inference rules described in Semantic Web Rule Language (SWRL) (Horrocks et al., 2004).

2.2 Modelling LOs and students

A popular metadata standard for describing LOs is the IEEE-LOM (LTSC, 2002). For that, this standard uses nine categories, of which the general and educational categories stand out. Among the fields of the general category, the entry field serves to store the LO's link (for example, from a YouTube video or from a Wikipedia page), and the keyword field can store the concepts that the LO covers. The educational category describes pedagogical information about LO, such as its type, its degree of difficulty and its target audience. Not all fields of the IEEE-LOM standard are widely used. Also, some fields have a reduced vocabulary. One way to extend reduced vocabularies is to use some extension. The Customised Learning Experience Online (CLEO) (CLEOLab, 2003) extends the IEEE-LOM standard by expanding, for example, the vocabulary of *learning resources type* of the educational category.

In addition to educational content, the student also needs to be modelled by computer systems. One of the commonly modelled characteristics is the student's learning style. The most suitable model for use in modelling student's learning styles is the FSLSM proposed by Felder et al. (1988). The popularity of this model is due to the fact that it covers more psychological aspects than other models (Deborah et al., 2014). The model has four polar dimensions: input (visual and verbal), organisation (sequential and global), perception (sensitive and intuitive) and processing (active and reflective). One of the instruments used to assess student preferences in these four dimensions is the index of learning styles questionnaire (Soloman and Felder, 2005).

2.3 Filtering and recommendation techniques

In CBF (Vanetti et al., 2010), objects with content characteristics similar to those objects that the target user liked in the past are recommended. The disadvantage of CBF is that the students can only receive LOs similar to their past experience, whereas in CF (De Medio et al., 2020), the recommendation history of other students is considered in the recommendation of new LOs for the target student. CF uses object evaluations (see Figure 1) to calculate the similarity of users or objects and make the recommendation. Both techniques, unlike knowledge-based (KB) recommendation (Tarus et al., 2017), suffer from rating sparsity and cold-start problems. KB recommendation aggregates knowledge about the student and learning materials to apply them in the recommendation process. The ontology-based recommendation (Tarus et al., 2017) is a type of KB recommendation that uses ontology to represent this knowledge.

Figure 1 Rating matrix of CF and KB recommendation, (a) rating matrix of LOs for CF (b) rating matrix of LOs for KB recommendation

Learner	O_1	O_2	O_3
L_1	4	5	?
L_2	3	4	3
L_3	4	3	4
L_4	1	3	5
L_5	5	5	4

(a)

Learner	LO	Level	Rating
L_1	O_3	Beginner	2
L_2	O_3	Advanced	3
L_3	O_3	Beginner	2
L_1	O_3	Intermediate	?
L_3	O_3	Intermediate	5

(b)

Notice in Figure 1 that the rating matrix in CF takes into account only the ratings of the LOs, while KB recommendation considers the student's level, which can be beginner, intermediate or advanced. The target student is identified by L_1. In KB recommendation, the prediction of the score that L_1 would give the O_3 LO will depend on the grade that O_3 received from other students at the same level as L_1. This level is an example of contextual information or knowledge about the student. It can be said, therefore, that KB is a type of CF that aggregates contextual information about the student, helping to reduce the rating sparsity and cold-start problems in CF.

2.4 Set covering problem

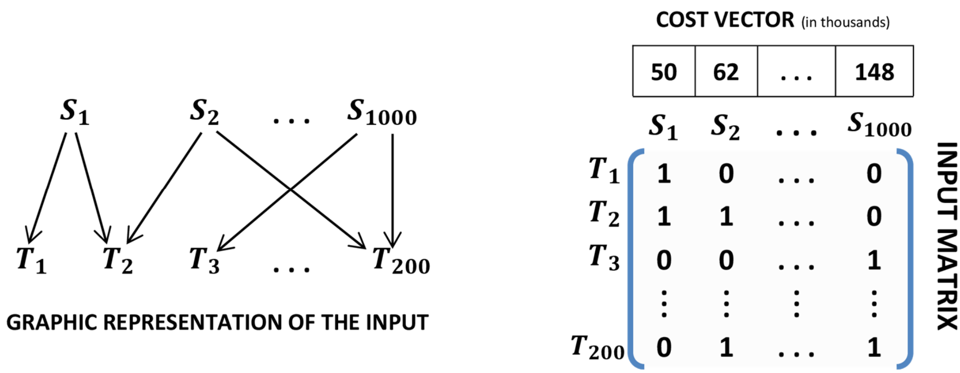
The SCP is a well-known combinatorial optimisation problem that has been applied to a wide range of applications (Lan et al., 2007), including crew scheduling in railway and airlines (Housos and Elmroth, 1997; Caprara et al., 1999), facility location problem (Vasko and Wilson, 1984) and industry production planning (Vasko et al., 1987).

The mathematical formulation of the SCP is as follows. Given m rows, n columns and an ($m \times n$) sparse matrix of zero-one elements a_{ij}, where $a_{ij} = 1$ if row i is covered by column j, and $a_{ij} = 0$ otherwise. Each column j covers at least one row from m rows and has an associated cost $c_j > 0$. The objective is to find a subset from n columns that covers all the rows of a_{ij} at a minimal cost. We define the LORP as the mathematical programming model of the SCP in Section 4.3.

To exemplify this problem in a real-life scenario, imagine that an automobile company needs to install repair shops in a state. More products require, in most cases, physical proximity between customers and suppliers, for the provision of technical assistance, if required. The problem is building stores across the state to meet all customer demand, which means serving every city (or town) in the state while spending as little money as possible. A repair shop can be strategically installed in a city to serve neighbouring cities, but depending on the installation location, the construction cost can be higher.

Given that this state has 200 cities and that the car company has 1,000 budgets with the cost of installing stores, following the formal definition of the SCP for this real-life problem, the objective is to find a subset from n = 1,000 repair shops that covers all the demand in the state, that is, it serves m = 200 cities at a minimum installation cost. The graphical representation of this instance of the problem is shown in Figure 2.

Figure 2 Input matrix and cost vector of the repair shop covering problem (see online version for colours)



Note that each column S (repair shop) has an associated cost (amount spent to install the store). Optimising this cost means reducing it as much as possible, respecting the SCP goal of covering all lines (cities). The sets S_1, S_2, …, S_{1000} have costs 50, 62, …, 148, respectively. Each set S is a repair shop that covers a set of cities. Repair shop S_1, for example, covers the cities T_1 and T_2, but does not cover the cities T_3 and T_{200}.

The SCP is NP-hard (Garey and Johnson, 1979) and exact algorithms (Balas and Carrera, 1996; Fisher and Kedia, 1990) are used to find its optimal solution, but these procedures are able to solve very limited size instances and are very time consuming, so exact algorithms are not practical for large-scale instances due to the computational complexity of SCP. For this reason, many researchers make a lot of efforts on developing metaheuristic algorithms (Bilal et al., 2013) based on constructive metaheuristics as ant colony optimisation (ACO) (Ren et al., 2008, 2010), evolutionary algorithms as genetic algorithm (GA) (Beasley and Chu, 1996; Solar et al., 2002; Wang and Okazaki, 2007) and local search (Musliu, 2006; Yagiura et al., 2006).

Exact and greedy algorithms can be good alternatives to solve the SCP when the instances are smaller, which is common in the educational context. In this work, we implement and compare two types of algorithms to solve the SCP (LORP): an exact algorithm and a greedy algorithm adapted from Golab et al. (2015). The simplest approach to solving the SCP is the greedy algorithm of Chvatal (1979). Greedy algorithms are fast, but they have a hard time finding the best solution, while exact algorithms are slower but find the best solution.

3 Related work

Much research combines recommendation techniques with ontologies and/or the web, including Wikipedia, for the recommendation of educational resources as shown in Table 1. For example, Limongelli et al. (2015) created a module in a system for a collaborative recommendation of Wiki pages used by teachers when creating their courses. The target teacher benefits from the recommendation made in the past to other teachers who have a teaching style similar to yours.

Table 1 Comparison of related literature with the proposal of this work

Reference	*Web content reuse*	*Ontology or semantic web technologies*	*LOs recommendation technique*	*LOs coverage using fine-grained concepts from different areas of knowledge*
Limongelli et al. (2015)	Yes	No	CBF and CF	No
Belizário and Dorça (2018)	Yes	Yes	GA	No
Falci et al. (2019)	Yes	Yes	Greedy alg.	No
Belizário et al. (2020)	Yes	Yes	CF, SWRL and PSO	No
Christudas et al. (2018)	No	No	CGA	No
Birjali et al. (2018)	Yes	No	ACO and GA	No
Ouf et al. (2017)	No	Yes	SWRL	No
Pereira et al. (2018)	Yes	Yes	SPARQL	No
Our proposal	*Yes*	*Yes*	*CF, SWRL, exact and greedy alg.*	*Yes*

Another approach that recommends Wiki content is presented in Belizário and Dorça (2018), but the content is recommended directly to the target student without using the teacher as an intermediary. This approach selects the best quality Wiki pages using the quality classes assigned to them by users. The sections (within these pages) that cover the concepts that the target student needs to learn are recommended. The approach uses an ontology for modelling students and LOs. The LORP is formalised as a SCP and is solved by a GA.

This same problem is solved by a faster algorithm considering a greedy heuristic, as shown in Falci et al. (2019). The intuition underlying heuristics is that LOs that meet the student's learning style, while covering more concepts, tend to deliver better candidates for the final solution. The algorithm that implements this heuristic is faster than GA, mainly for instances with thousands of LOs, for which GA can become impractical given the exponential search space and the high number of calculations of the fitness function.

This LORP defined as a covering problem is also solved in (Belizário et al., 2020) using CF, SWRL and particle swarm optimisation (PSO) (Kennedy and Eberhart, 1995). In the previous works (Belizário and Dorça, 2018; Falci et al., 2019), the authors considered only the user's search parameters when recommending LOs. In Belizário et al. (2020), in addition, the authors also consider the history of rating given to LOs by students with ratings similar to the student to whom the recommendation is directed.

GAs can be used to personalise the recommendation of LOs in contexts with many learning parameters. In Christudas et al. (2018), the authors proposed a compatible genetic algorithm (CGA) to the recommendation of LOs. The CGA forces compatibility of:

a the LO type in relation to the learning style of the student

b the LO complexity level with respect to the knowledge level of the student

c the interactivity level of the LO based on the satisfaction level of the student during the learning process.

In Birjali et al. (2018), the authors created an adaptive e-learning model based on big data that uses a MapReduce-based GA to determine the suitable educational objectives through the adequate student e-assessment method and an ACO algorithm to generate an adaptive learning path for each learner. After that, a MapReduce-based social networks analysis is performed to determine the learning motivation and social productivity in order to assign a specific learning rhythm to each student.

The SW, in addition to ontologies, also has technologies that have been explored by some authors for the recommendation of LOs. Ouf et al. (2017) developed a tool for an intelligent learning ecosystem using ontologies and rules in SWRL. Ontologies are used to model students and to tailor components of the learning process to students, such as LOs, preferred learning activities and relevant teaching-learning methods.

Pereira et al. (2018) created an infrastructure for the recommendation of educational resources based on information such as the user's profile and the educational context, extracted from the social network Facebook. SW technologies and information extraction techniques are used to extract, enrich and define the profiles and interests of users. The recommendation strategy is based on linked data, LO repositories and videos, benefiting from the time the user spends on the web.

In our previous paper (Belizário et al., 2020), we formulate the LORP as a covering problem, and in this work, we take advantage of this idea to define the LORP as the SCP, so the LORP becomes able to consider the concepts that the student needs to learn. It is noted that the works of the related literature use the web for the reuse of content (including LO repositories) and/or use SW technologies, but they do not combine the recommendation of fine-grained concepts typical of the step-by-step approach with the recommendation of content from different areas of knowledge.

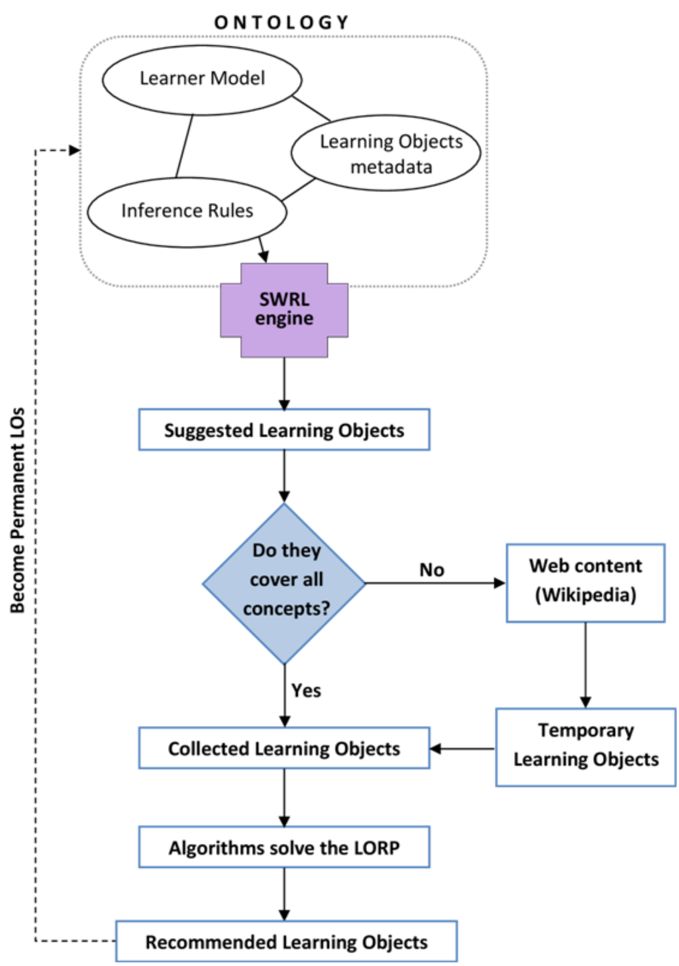
This work contributes to the solution of this state-of-the-art challenge by proposing an approach for the recommendation of LOs from different areas of knowledge, considering concepts with fine granularity. This is the main advance of our work in relation to the work initially proposed in Belizário et al. (2020). Our proposed approach is detailed in the next section.

4 Proposed approach

The proposed RS is based on a hybrid recommendation approach that combines CF and ontology-based recommendation techniques. The ontology is used to model students and store metadata of LOs. Our RS is able to recommend content from different areas of knowledge by reusing web content, especially Wikipedia pages. Web content is reused when the ontology LOs are not enough to cover all the concepts that the student needs to learn. In addition, we implemented the hint-type LO in the ontology for a more fine-grained recommendation for students.

Wikipedia is a great digital encyclopaedia with 6 million and 585 thousand articles in English and 1 million and 96 thousand articles in Portuguese, published as of 22nd January 2023. Wikipedia's content may be copied, modified and redistributed under a Creative Commons BY-SA licence. The Wikipedia community can assess the quality of articles with almost a dozen labels that serve to rank articles from the worst to the best. In Section 4.4, we show how these classes can be used as a user search parameter to ensure accuracy and confidence in creating LOs using Wiki sections.

Figure 3 Overall of the proposed RS (see online version for colours)



Users' search parameters, such as the concepts they need to learn, their preferences and their questions, are captured by the RS interface, which can be, for example, a chatbot. The advantage of the chatbot is its ability to understand human language, which can be exploited to extract concepts that the student has to learn or has doubts about. The ontology class in which these search parameters are stored is called ideal LO, as these are the ideal characteristics expected to be found in the recommended LOs. Once the ideal LO is filled in, the system is ready to work, regardless of the interface used to capture user input.

Based on the recommendation model in Figure 3, the ontology uses inference rules to suggest LOs. If the suggested LOs do not cover all concepts, then web content including Wikipedia pages dealing with the uncovered concepts are transformed into temporary LOs in the ontology, which are joined with the suggested LOs to form the set of collected LOs. The collected LOs are supposed to cover all concepts. These LOs and concepts are the input to the algorithms that solve the LORP, which is a cover problem that aims to recommend LOs that cover the concepts that the student needs to learn. An exact and a greedy algorithm are used to solve this problem. After solving it, the LOs of the best solution found are recommended to the student. The temporary LOs of this recommendation become permanent LOs in the ontology.

In the following sections, we present the ontology in Section 4.1 and the improvements made to it in Section 4.2. Section 4.3 shows how the SCP can be used to formally define the LORP, and Section 4.4 describes how the cost of LOs is calculated.

4.1 Ontology

The ontology used in this work was initially proposed in Belizário and Dorça (2018). It stores knowledge about students and specifies the LOs according to the IEEE-LOM standard and its CLEO extension. In addition, ontology implements SWRL rules with two different purposes. First, some rules are used to infer the types of LOs appropriate to the student's learning style based on the theory described by Graf et al. (2010), who address which types of LOs should be recommended for each type of student profile associated with the FSLSM. Second, other rules are used to perform the selection of LOs that are similar to the user's search parameters.

4.1.1 Domain model

The ontology does not contain the LOs, but their metadata. Each LO has the nine categories of the IEEE-LOM standard, which are represented in the ontology by the classes General_1, LifeCycle_2, MetaMetaData_3, Technical_4, Educational_5, Rights_6, Relation_7, Annotation_8 and Classification_9 (see Figure 4).

Properties whose range is a set of fixed values, such as the hasDifficulty property that has the range VeryEasy, Easy, Medium, Difficult and VeryDifficult, were implemented using the value partition pattern (Rector, 2005). The name of all classes that follow this pattern ends with ValuePartition, and their subclasses correspond to fixed values.

There are four types (subclasses) of LOs in the ontology (see Figure 4). They are:

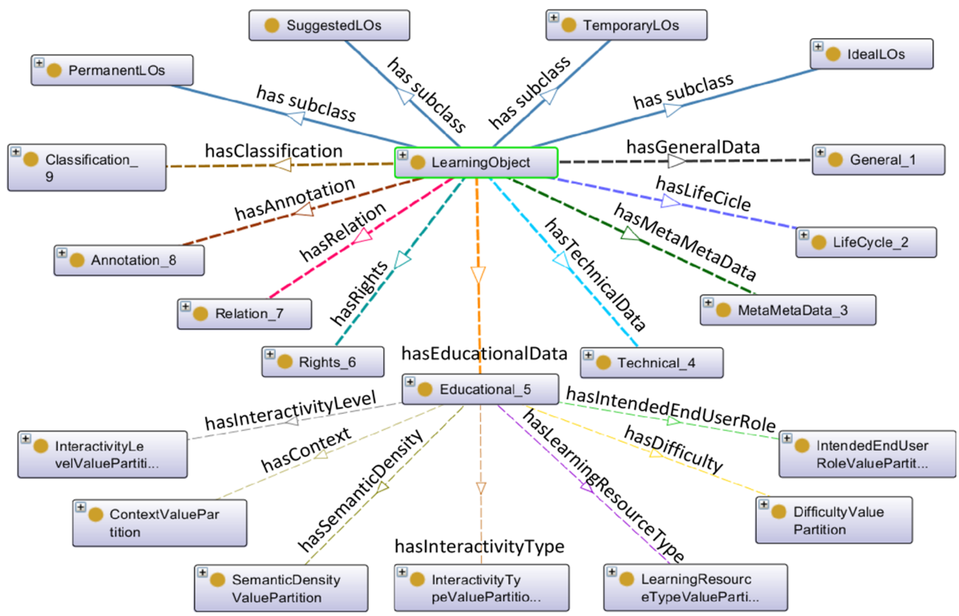
1 *Ideal LO:* It contains the user's search parameters, including the concepts that the student is expected to learn. The recommended LOs are expected to equal the ideal LO.

2 *Permanent LO:* Represent LOs already stored in the ontology, either because they have been created by the tutor or previously recommended.

3 *Suggested LO:* Contains inferred instances of permanent LOs that have some similarity to the ideal LO. These instances are LOs suggested by inference rules.

4 *Temporary LO:* Contains instances of LOs found on the web and temporarily stored in the ontology.

Figure 4 Domain model in the ontology (see online version for colours)



In Figure 4, the educational category is expanded to show the key educational or pedagogic characteristics in describing the content of LOs. These attributes have a set of fixed values. In addition, the educational category has four attributes characterised by primitive data types: typical age range, typical learning time, description and language.

4.1.2 Learner model

The ontology is open for the addition of new implementations – including student characteristics, such as name and knowledge level – according to the educational context in which it will be used. In this work, we consider the psychological aspects of the students that are structured in the ontology through the FSLSM as shown in Figure 5.

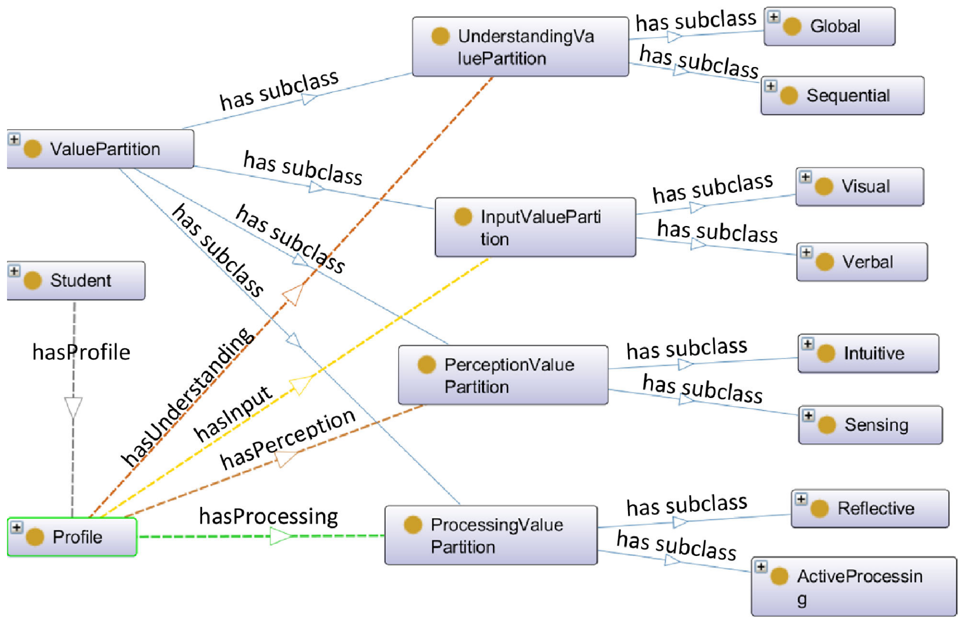
This model helps to recommend the most appropriate types of LOs for each type of student profile.

The student's learning style is given by the profile class, which has the four polar dimensions of the FSLSM. For example, in the input dimension, the student will be either visual or verbal. These dimensions are properly structured into classes and their fixed-value subclasses using the value partition pattern.

4.2 Ontology improvements

We incremented the ontology proposed in Belizário and Dorça (2018) to improve the LO recommendation process. In the teaching-learning process, the student naturally has doubts when studying content or solving exercises. These doubts may be related to concepts and LOs. In the work (Belizário and Dorça, 2018), the authors made the recommendation considering only the concepts that the student needs to learn. In this work, beyond concepts, we consider the student doubts associated with LOs. Thus, it is possible to recommend to the students hints related to the LO that they have doubts about.

Figure 5 Learner model in the ontology (see online version for colours)



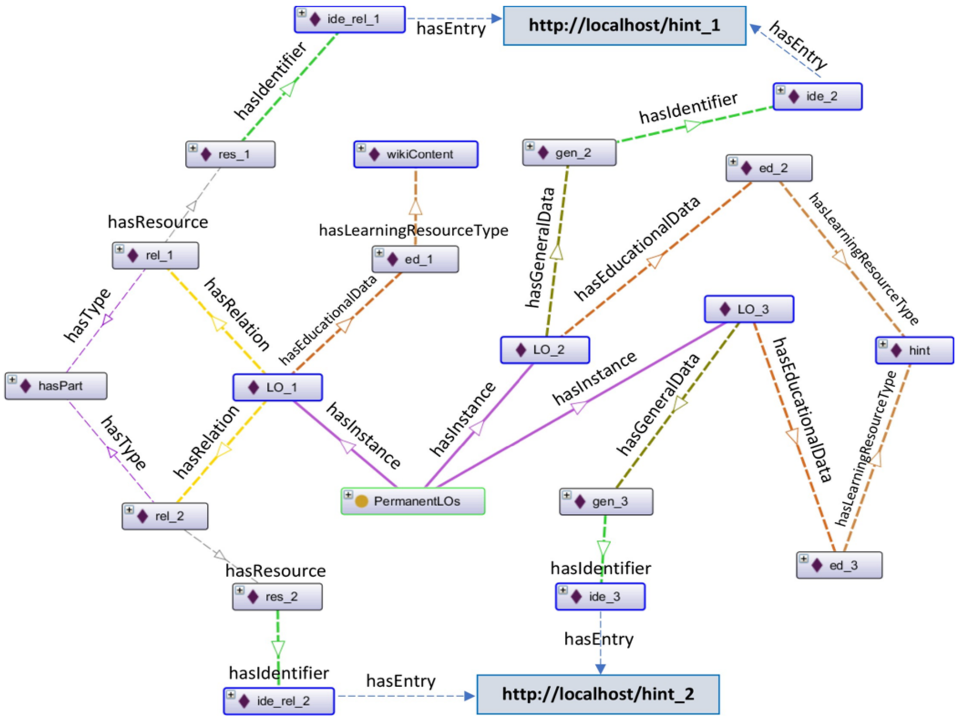
4.2.1 *Hint-type LOs*

To improve the LO recommendation process, we implement in the ontology the hint type and use the relation class to link the hint-type LOs to the LOs of other types (such as exercise and Wiki content).

In Figure 6, the LO_1 is a Wiki content that has two hints (LO_2 and LO_3), and the three LOs are instances of the PermanentLOs class. Note that LO_1 relates to hints via the Relation category.

The IEEE-LOM standard defines some values, such as haspart, hasversion and requires, for the type relation. The most appropriate value to relate a LO to its hints is haspart, because many hint-type LOs can be part of a LO. In Figure 6, for example, the LO_1 is a Wiki-type LO (wikiContent) and has two parts (hasPart) that correspond to the objects LO_2 and LO_3, which are of the hint type. Note that these relationships are done through the uniform resource identifier (URI) used to identify the location of LO_2 and LO_3. As many relationships (hasRelation) as necessary can be created for each LO, but the LOM standard sets a maximum of 100.

Figure 6 Relationship (hasPart) between wikiContent and hint LOs (see online version for colours)



4.2.2 Ideal LO

The user's search parameters are stored in the IdealLO class of the ontology. Considering that through the RS interface, for example, in a dialogue between the student and a chatbot, the LO in which the student has doubts has been identified, then the URI of this LO is stored in the ideal LO. For this, we use the relation metadata to store the URI. An example of this relationship is shown in Figure 7, in which LO_ideal_00001 has the identification of the LO (Wiki content) that generates the student's doubt.

The ideal LO (LO_ideal_00001) relates to a resource (res_ideal_00001) identified by the URI https://en.wikipedia.org/wiki/Mitosis. This relation (rel_ideal_00001) is of the haspart type, so this resource (Wiki content) is part of the ideal LO, i.e., it is part of the user's search parameters. This is how the URI of the LO about which the student has doubts is stored in the ideal LO.

Figure 7 shows that the ideal LO is recommended for the student_00001, who has a profile characterised by verbal, global, intuitive and active. From this semantic representation, we create inference rules to allow convenient hints for students to be chosen along with collected LOs. Table 2 shows an example of a SWRL rule and its meaning.

Using the URI of idealLO, the LO generating the student's doubt is identified, and all the hints associated with this LO are suggested, according to the inference rule in Table 2.

Figure 7 Ideal LO (see online version for colours)

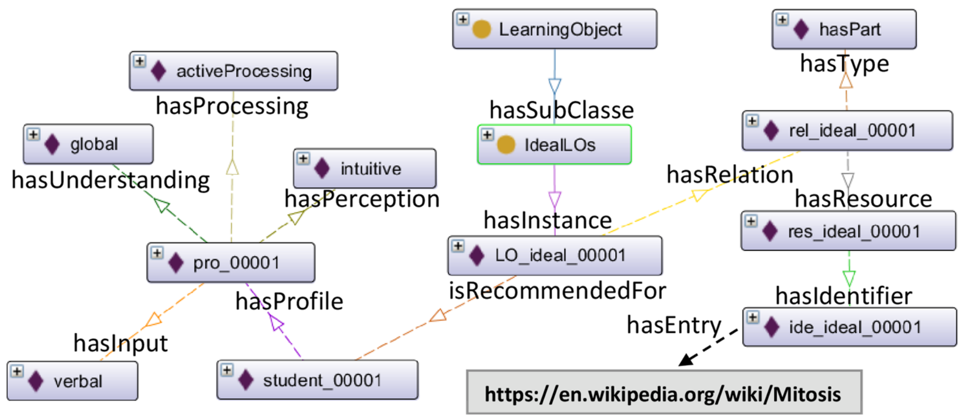


Table 2 SWRL rule

SWRL rule	*Meaning*
IdealLOs(*?idealLO*) ∧ Relation_7(?rel) ∧ hasRelation(?idealLO, ?rel) ∧ Resource(?res) ∧ hasResource(?rel, ?res) ∧ Identifier(?ideideal) ∧ hasIdentifier(?res, ?ideideal) ∧ hasEntry_(?ideideal, *?uri*)	*IF* there exists *?idealLO* such that *?idealLO* is an ideal LO, and *?idealLO* has relation with (searches for) a URI *?uri*
∧	*AND*
PermanentLOs(*?lo*) ∧ General_1(?gen) ∧ hasGeneralData(?lo, ?gen) ∧ Identifier(?ide) ∧ hasIdentifier(?gen, ?ide) ∧ hasEntry_(?ide, *?uri*)	There exists *?lo* such that *?lo* is a permanent LO, and *?lo* is addressed by *?uri*
∧	*AND*
Relation_7(?relat) ∧ hasRelation(*?lo*, ?relat) ∧ Resource(?resou) ∧ hasResource(?relat, ?resou) ∧ Identifier(?iderel) ∧ hasIdentifier(?resou, ?iderel) ∧ hasEntry_(?iderel, *?urihint*)	*?lo* has relation with a URI *?urihint*
∧	*AND*
PermanentLOs(*?lohint*) ∧ General_1(?genhint) ∧ hasGeneralData(?lohint, ?genhint) ∧ Identifier(?idehint) ∧ hasIdentifier(?genhint, ?idehint) ∧ hasEntry_(?idehint, *?urihint*) ∧ Educational_5(?edu) ∧ hasEducationalData(?lohint, ?edu) ∧ Hint_extended(?hinttype) ∧ hasLearningResourceType(?edu, ?hinttype)	There exists *?lohint* such that *?lohint* is a permanent LO, and *?lohint* is of the hint type, and *?lohint* is addressed by *?urihint*
→	*THEN*
SuggestedLOs(*?lohint*)	*?lohint* is a suggested LO

4.3 LORP defined as the SCP

In the context of teaching and learning, imagine a situation in which a student needs to learn four concepts belonging to the finite set $X = \{C_1, C_2, C_3, C_4\}$. Consider a collection of subsets of X given by $F = \{O_1, O_2, O_3, O_4\}$, where $O_1 = \{C_1, C_3\}$, $O_2 = \{C_3, C_4\}$, $O_3 = \{C_1\}$, and $O_4 = \{C_2, C_3\}$. The sets O_1, O_2, O_3 and O_4 have costs 5, 3, 2 and 2, respectively. Each element of F is a LO that covers a set of concepts. LO O_1, for example, covers the concepts C_1 and C_3. In this scenario, the objective is:

- Find a set of LOs that together cover all concepts (elements of X) at minimal cost.

This objective is equivalent to the SCP, and so the solution for the previous example is $\{O_2, O_3, O_4\}$ with cost 7. In the context of this work, the cost of O_j is inversely proportional to the importance that O_j has for the student. The lower the cost of the LO, the more it meets the student's knowledge and learning preferences, hence the importance of optimising the cost, that is, finding a set of LOs that has the lowest possible cost while covering all the concepts that the student needs to learn.

The formal definition of the SCP is as follows. Let a_{ij} be a zero-one matrix with m rows and n columns, the goal is to cover all rows using a subset of columns at minimal cost. Let $x_j = 1$ if the column j (with cost $c_j > 0$) is part of the solution, and $x_j = 0$ otherwise, then the SCP is formulated as:

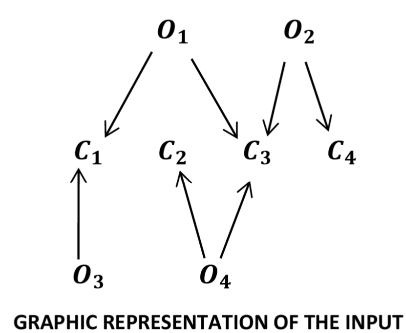
$$
\begin{aligned}
&\text{Minimise } \sum_{j=1}^{n} c_j x_j \\
&\text{subject to } 1 \le \sum_{j=1}^{n} a_{ij} x_{ij}, \quad i = 1, \ldots, m \qquad (1)\\
&\qquad x_j \in \{0, 1\}, \quad j = 1, \ldots, n
\end{aligned}
$$

The LORP, whose objective is to find a coverage of LOs that covers all concepts at minimum cost, corresponds to the SCP formalised by equation (1). The value c_j is calculated in Section 4.4.

The input matrix a_{ij} is filled with the user input concepts and with the LOs collected by the RS (i.e., the collected LOs shown in Figure 3). Each row i corresponds to an input concept C_i, and each column j is associated with an LO O_j resulting from the set of collected LOs. $a_{ij} = 1$, if O_j covers C_i, and $a_{ij} = 0$ otherwise.

Figure 8 shows the input matrix, the cost vector and the graphic representation of the previous example. The LOs O_1, O_2, O_3 and O_4 have costs 5, 3, 2 and 2, respectively, as shown in Figure 8. The solution for this example is $\{O_2, O_3, O_4\}$ with cost 7, as stated earlier.

Figure 8 Input matrix and cost vector of the LO covering problem



COST VECTOR

5	3	2	2

INPUT MATRIX

	O_1	O_2	O_3	O_4
C_1	1	0	1	0
C_2	0	0	0	1
C_3	1	1	0	1
C_4	0	1	0	0

4.4 Improvements in cost calculation

The cost of the column j (c_j) corresponds to the j^{th} value in the cost vector. The cost c_j is calculated by the dissimilarity between the user's search parameters (O_{ideal}) and O_j. The characteristics of each LO, such as the type (exercise, lecture notes, etc.) and the keywords, are compared with the equivalent search parameters configured in the O_{ideal} metadata of the ontology. The calculation of this cost was initially proposed in Belizário and Dorça (2018), where it was formally defined as:

$$c_j = diss\left(O_{ideal}, O_j\right) \tag{2}$$

The $diss(O_{ideal}, O_j)$ value is inversely proportional to the degree of similarity between O_{ideal} and O_j. The result of $diss(O_{ideal}, O_j)$ depends on the proximity between O_{ideal} and O_j. The parameters of O_j, such as degree of difficulty, semantic density and learning resource type, are compared with the corresponding parameters of O_{ideal} given by the user.

Formally, let α_i be the value of the i^{th} parameter. The calculation of the dissimilarity between O_j and the user's search parameters is given by equation (3):

$$diss\left(O_{ideal}, O_j\right) = \sum_{i=1}^{p}\left(\alpha_{i(ideal)} - \alpha_{i(j)}\right) \tag{3}$$

where p is the number of parameters, $\alpha_{i(ideal)}$ is the value of the i^{th} parameter of O_{ideal}, and $\alpha_{i(j)}$ is the value of the i^{th} parameter of O_j. In this work, we consider six parameters:

- *Title:* The titles are compared by the cosine similarity.
- *Interactivity type:* Each vocabulary term is mapped to a value (active = 0, mixed = 0.5, expositive = 1) that corresponds to the $\alpha_{i(j)}$ of equation (3).
- *Learning resource type:* Equation (3) results in 0 if the O_{ideal} and the O_j are the same resource type, and it results in 1 otherwise.
- *Interactivity level* and *semantic density:* Each vocabulary term is mapped to a value (*verylow* = 0, *low* = 0.25, *medium* = 0.5, *high* = 0.75, *veryhigh* = 1) that corresponds to the $\alpha_{i(j)}$ of equation (3).
- *Difficulty:* Each vocabulary term is mapped to a value (*veryeasy* = 0, *easy* = 0.25, *medium* = 0.5, *difficult* = 0.75, *verydifficult* = 1) that corresponds to the $\alpha_{i(j)}$ of equation (3).

In addition to these parameters, other parameters can be considered depending on the needs of each educational context. One could use, for example, a parameter for recommending higher-quality Wiki-type LOs. The amount of Wikipedia articles has grown exponentially, and associated with this, articles are edited all the time, which can change the quality of the article. To deal with this problem, several types of research have been developed for the automatic evaluation of Wiki articles.

Warncke-Wang et al. (2013) used a classifier based on a decision tree to identify the quality of Wikipedia articles. The authors showed that by extracting only five features from Wiki pages, it is possible to obtain significant results. This classifier can classify Wiki articles into seven different classes: *FA*, *GA*, *A*, *B*, *C*, *Start* and *Stub*, all defined by

Wikipedia itself. The authors divided the articles into two large classes: *GoodEnough* articles (containing *FA*, *A* and *GA*) and *NeedsWork* (containing *B*, *C*, *Start* and *Stub*).

The quality of Wiki articles corresponds to one of the p parameters defined by equation (3). Let a_q be a quality parameter, and to use it in equation (3), the $a_{q_{ideal}}$ and a_{q_j} of the j^{th} LO compared to the ideal LO are defined below:

- $a_{q_{ideal}} = 1$ (the ideal LO certainly has the highest level of quality, belonging to the *FA* class)
- a_{q_j} is the value corresponding to the quality rating received by the j^{th} LO according to the following mapping: *FA* → 1.0, *A* → 0.85, *GA* → 0.70, *B* → 0.50, *C* → 0.30, *Start* → 0.15 and *Stub* → 0.

Later, the cost c_j was reformulated in Belizário et al. (2020) as:

$$c_j = diss\left(O_{ideal}, O_j\right) + \left(1 - P_j^L\right) \tag{4}$$

In this case, it considers the prediction P_j^L in addition to the degree of dissimilarity between O_{ideal} and O_j to calculate the cost c_j. This prediction P_j^L represents the relevance that O_j has for the target student L. This relevance is calculated using CF.

In this paper, we improve this cost calculation to make fine-grained LO recommendations using hint-type LOs for this. The new cost is formally defined as:

$$c_j = diss\left(O_{ideal}, O_j\right) + \left(1 - P_j^L + 1 - H_j\right) * \max_{j \in \{1,\ldots,n\}} diss\left(O_{ideal}, O_j\right) \tag{5}$$

where the max operator is a weight given to P_j^L and H_j to assign them the same importance as *diss*.

The recommendation process depends on the student's learning needs. The RS interface (the chatbot, for example) can identify three different student intentions, who can choose:

a hint-type LOs

b non-hint type LOs

c LOs of any type.

In this last case, $H_j = 1$, regardless of the type of O_j, which makes the term $(1 - H_j)$ neutral in the calculation of c_j.

The focus of this work is on the first two situations, in which the number of hints in the LORP solution is relevant for the student. In this case, the value of H_j depends on the recommendation mode used:

- *Recommendation mode 1* (the less hints the better): $H_j = 0$ if the LO O_j is of the hint type, and $H_j = 1$ otherwise.
- *Recommendation mode 2* (the more hints the better): $H_j = 1$ if the LO O_j is of the hint type, and $H_j = 0$ otherwise.

The RS has two recommendation modes. If the student has doubts when studying some content or solving an exercise, then the *more hints the better* to provide a more fine-grained recommendation. On the other hand, if the student has no doubts and needs to learn new concepts, then *the less hints the better* to recommend. In this case, the RS should recommend other types of LOs, such as lectures and exercises. Our two research questions derived from equation (5) that guide the validation of our approach are:

- *Research question 1:* Does the use of CF (variable P_j) contribute to the recommendation of the LOs with the best rating for the student?
- *Research question 2:* Does the use of H (hint: fine-grained LOs) in the calculation of c_j in the objective function (OF) improve the quality of LOs recommendation in relation to the number of hints expected by students?

In this work, the value P_j^L in equation (5) is the prediction of the rating the target student would give to the new O_j. This value represents the importance that the LO has for the student and is given in a real interval [0, 1], in which the higher its value, the greater the importance that the LO has for the student. This prediction is calculated using the k-nearest neighbours (kNN) (Adomavicius and Tuzhilin, 2005) approach proposed in Tarus et al. (2018); kNN is a simple algorithm, whose training phase corresponds to the simple storage of instances. It is the most used algorithm in CF (Adomavicius and Tuzhilin, 2005). It finds the k students, among those who evaluated the resource O_j, more similar to the target student. The goal is to predict the rating the target student would give O_j using the ratings that O_j received from other similar students (nearest neighbours).

KB recommendation aggregates knowledge about the student and learning materials to use them in the recommendation process. In this case, to predict P_j^L, the similarity calculation considers only students contextually similar to the target student L. For example, the similarity calculation takes into account only students who have a similar knowledge level or learning style as the target student.

It is not possible to make a reliable calculation of similarity when few students have evaluated the same LO (rating sparsity problem) or a new student has not rated any LOs (cold-start problem). In this case, information about students, such as their knowledge level and learning style, can be used in the similarity calculation to predict P_j^L. Thus, the KB recommendation contributes to reducing the rating sparsity and cold-start problems. It was possible to simplify the experimental tests without compromising them, using only the CF proposed in Tarus et al. (2018), disregarding the use of KB recommendation, which can be properly used in a real learning context.

5 Experimental analysis

The algorithms were implemented in Python, and the experiments were run on a notebook with AMD Quad-Core A10-9600P 2.40 GHz/8G and Windows 10 OS.

We have employed statistical tests designed to detect significant differences and to estimate their magnitude from the tests performed. The experiments were designed as a randomised complete block design (RCBD). By treating the problems as blocks, it was possible to model and remove the effects of different instances on the performance of the algorithm and obtain an overall performance difference across all test instances used

(Montgomery, 2012). The null hypotheses of absence of differences among the methods evaluated over all problems were considered against two-sided alternatives. To avoid the assumptions of normality, Wilcoxon test (non-parametric) was employed.

After testing for significance, least squares estimators of the block (instance) effects were obtained and subtracted from the samples, thus allowing a problem-independent estimation of the effect size for each algorithm (Montgomery, 2012). The estimations of effect size were calculated by Tukey's (1949) test for post-hoc analysis.

5.1 Benchmark instances and dataset

The dataset has 24 instances with symbolic data, as shown in Table 3. Note that there are six different values (2, 6, 10, 25, 40, 55) for the number of rows (concepts) and four values (100, 500, 2,000, 10,000) for the number of columns (LOs).

Table 3 The main features of the used benchmark instances

Inst.	*m*	*n*	*Density (%)*	*Inst.*	*m*	*n*	*Density (%)*
1	2	100	50	13	25	100	10
2	2	500	50	14	25	500	10
3	2	2,000	50	15	25	2,000	10
4	2	10,000	50	16	25	10,000	10
5	6	100	50	17	40	100	10
6	6	500	50	18	40	500	10
7	6	2,000	50	19	40	2,000	10
8	6	10,000	50	20	40	10,000	10
9	10	100	20	21	55	100	10
10	10	500	20	22	55	500	10
11	10	2,000	20	23	55	2,000	10
12	10	10,000	20	24	55	10,000	10

The implemented algorithms are deterministic, even so, each instance was solved ten times to find the average runtime, resulting in the average prediction and the average number of hints. The tests were executed in *recommendation mode 1* (the less hints the better) and *recommendation mode 2* (the more hints the better), which were explained in Section 4.4.

Table 3 shows features of a small dataset that we created to test the LORP. The benchmark instances simulate the educational context. The main characteristics of the instances are the name of instances (Inst), the number of rows (m), the number of columns (n) and the density. Instances were created to have densities equal to either 10%, 20% or 50%.

Each instance is composed of the input matrix and a cost vector (see Figure 8), but in our dataset, this vector is interpreted as a dissimilarity vector that is used to calculate the cost vector. The number of columns covering each row of the input matrix is defined by the density of the instance. For example, if the density is 10%, then for each row of the input matrix, 10% of the total number of columns is randomly chosen to cover it.

Table 4 Rating matrix used in testing

	O1	*O2*	*O3*	*O4*	*O5*	*O6*	*O7*	*O8*	*O9*	*O10*	*O11*	*O12*	*O13*	*O14*	*O15*	*O16*	*O17*	*O18*	*O19*	*O20*	*O21*	*O22*	*O23*	*O24*	*O25*	*O26*	*O27*	*O28*	*O29*	*O30*	*O31*	*O32*	*O33*	*O34*	*O35*	*O36*	*O37*	*O38*	*O39*	*O40*	*O41*	*O42*	*O43*	*O44*	*O45*	*O46*	*O47*	*O48*	*O49*	*O50*
Type	1	0	0	1	1	0	1	0	1	0	1	1	0	0	0	0	1	0	1	1	1	0	1	1	0	0	1	1	1	0	0	0	1	1	1	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0
L1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L4	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L5	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L6	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L7	0	2	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L8	0	2	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L9	0	2	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L10	0	1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L11	0	2	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L12	0	3	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L13	0	0	0	1	3	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	4	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L14	0	0	0	1	3	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L15	0	0	0	1	3	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4 Rating matrix used in testing (continued)

	O1	*O2*	*O3*	*O4*	*O5*	*O6*	*O7*	*O8*	*O9*	*O10*	*O11*	*O12*	*O13*	*O14*	*O15*	*O16*	*O17*	*O18*	*O19*	*O20*	*O21*	*O22*	*O23*	*O24*	*O25*
L16	0	0	0	2	3	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L17	0	0	0	1	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L18	0	0	0	1	5	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L19	0	0	0	0	0	0	1	2	4	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L20	0	0	0	0	0	0	1	2	4	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L21	0	0	0	0	0	0	1	2	4	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L22	0	0	0	0	0	0	1	2	3	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L23	0	0	0	0	0	0	1	3	3	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L24	0	0	0	0	0	0	2	3	4	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L25	0	0	0	0	0	0	0	0	0	0	1	2	3	4	5	0	0	0	0	0	0	0	0	0	0
L26	0	0	0	0	0	0	0	0	0	0	1	2	3	4	5	0	0	0	0	0	0	0	0	0	0
L27	0	0	0	0	0	0	0	0	0	0	1	2	3	4	5	2	3	1	5	4	0	0	0	0	0
L28	0	0	0	0	0	0	0	0	0	0	2	2	3	5	5	1	2	2	4	4	0	0	0	0	0
L29	0	0	0	0	0	0	0	0	0	0	3	2	3	3	4	0	0	0	0	0	5	1	4	5	1
L30	0	0	0	0	0	0	0	0	0	0	1	1	2	2	4	0	0	0	0	0	3	2	5	5	3

	O26	*O27*	*O28*	*O29*	*O30*	*O31*	*O32*	*O33*	*O34*	*O35*	*O36*	*O37*	*O38*	*O39*	*O40*	*O41*	*O42*	*O43*	*O44*	*O45*	*O46*	*O47*	*O48*	*O49*	*O50*
L16	0	0	1	2	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L17	0	0	0	0	0	4	3	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L18	0	0	0	0	0	2	4	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L19	0	0	0	0	0	0	0	0	3	2	3	5	0	0	0	0	0	0	0	0	0	0	0	0	0
L20	0	0	0	0	0	0	0	0	2	3	3	4	0	0	0	0	0	0	0	0	0	0	0	0	0
L21	0	0	0	0	0	0	0	0	0	0	0	0	1	3	4	3	0	0	0	0	0	0	0	0	0
L22	0	0	0	0	0	0	0	0	0	0	0	0	1	2	2	3	0	0	0	0	0	0	0	0	0
L23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	3	3	5	0	0	0	0	0
L24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	5	1	4	0	0	0	0	0
L25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	3	4	2
L26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	5	3	1
L27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The input matrix and the dissimilarity vector of the instances were created to simulate a real-world scenario. Dissimilarity [*diss* in equation (5)] is an integer value ranging from 1 to 10% of the number of columns in the instance. For example, in 500-column instances, the columns from 1 to 10 have *diss* = 1, columns from 11 to 20 have *diss* = 2, and so on. Note that the last ten columns have *diss* = 50 (10% of 500 columns).

In addition to *diss*, it is necessary to determine P_j^L and H_j to calculate the cost using equation (5). For that, we create a rating matrix to simulate a real-life scenario with the ratings the students gave for LOs they evaluated (see Table 4). It has 30 students (each one in a row) and 50 LOs, each in a column; they correspond to the first 50 LOs of the instances. The value of row i and column j corresponds to the grade, in an integer interval [1, 5], that student i gave to LO j. If this grade is 0, then the student i has not rated LO j. The first row of the rating matrix is used to identify the type of each LO. The value of the first row and column j is 1 if the LO j is of the hint type. Otherwise, the value is 0. Half of the LOs (25) were randomly chosen to be of the hint type. The other LOs of each instance (O_{51}, O_{52}, …, O_n) have $H_j = 0.5$ in both recommendation modes (the less/more hints the better).

For the first 50 LOs of each instance, it is possible to predict P_j^L. The other LOs of each instance (O_{51}, O_{52}, …, O_n) were not rated by any student (cold-start problem), so the prediction value assigned to them is given by the arithmetic mean of the ratings of the LOs evaluated by the target student L.

For the execution of the tests, the target student $L11$ was chosen in Table 4. The *k*NN algorithm used to calculate the prediction is set to $k = 3$. It finds the three students, among those who evaluated the resource O_j, more similar to the learner $L11$.

5.2 Selected algorithms and parameters

In this work, an exact algorithm and a greedy heuristic named concise weighted set cover (CWSC) were considered for the LORP solution. The exact algorithm belongs to the Pulp Library (Mitchell et al., 2011), which is an open-source package written in Python to express linear programming models in a way similar to the conventional mathematical notations.

The CWSC algorithm results from the authors' motivation in Golab et al. (2015) to find a generalisation for the weighted set cover (Garey and Johnson, 1979) and maximum coverage (McGregor and Vu, 2019) problems. The CWSC input is a set of n elements, a collection of weights for those elements, an integer size constraint value k and a minimum coverage fraction s. The output is a subset of up to k sets whose union contains at least sn elements and whose sum of weights is minimal.

The CWSC algorithm was adapted for the solution of the LORP. The goal is to cover all the concepts ($s = 1$) using a number of columns at most equal to the number of lines ($k = m$) of the input matrix.

5.3 Comparison of the versions of each algorithm with and without prediction

To evaluate the CF (prediction) implemented in the proposed approach, we use the adapted CWSC and the exact algorithm to solve the LORP. Two versions of each algorithm were implemented. The difference between them relates to how the c_j cost is calculated. In the first, the cost is calculated by equation (6), which does not use the variable P (prediction), while in the second, the cost is calculated by equation (5), which uses the prediction.

$$c_j = diss\left(O_{ideal}, O_j\right) + \left(1 - H_j\right) * \max_{j \in \{1,\ldots,n\}} diss\left(O_{ideal}, O_j\right) \tag{6}$$

The two versions of each algorithm are compared in Table 5. In the no columns, c_j is calculated by equation (6), and in the yes columns, c_j is calculated by equation (5). Each LO in the solution has a rating that is either given in Table 4 or predicted by calculating P_j. The results obtained for the experimental comparison are summarised in Table 5, which reports the mean values of the ten runs (replications). These values represent the importance that the LO has for the student. The closer to 1, the greater the importance that the LO has for the student.

Table 5 Comparison of the average ratings in the solutions with (yes) and without (no) the P variable in solving the LORP

	Recommendation mode 1				*Recommendation mode 2*			
Instance	*Exact*		*CWSC*		*Exact*		*CWSC*	
	No	*Yes*	*No*	*Yes*	*No*	*Yes*	*No*	*Yes*
1	0.697	0.730	0.730	0.730	0.697	1.000	0.697	1.000
2	0.697	0.809	0.730	0.809	0.697	0.995	0.697	0.995
3	0.596	0.809	0.596	0.809	0.697	0.995	0.697	0.995
4	0.697	0.809	0.697	0.809	0.697	0.929	0.697	0.929
5	0.734	0.753	0.734	0.703	0.697	0.929	0.697	0.929
6	0.697	0.753	0.697	0.753	0.601	0.929	0.697	0.929
7	0.663	0.753	0.663	0.714	0.697	1.000	0.697	1.000
8	0.697	0.787	0.674	0.787	0.843	0.856	0.794	0.900
9	0.734	0.734	0.716	0.734	0.625	0.723	0.661	0.810
10	0.717	0.753	0.718	0.756	0.697	0.810	0.697	0.813
11	0.746	0.746	0.683	0.746	0.810	0.813	0.742	0.813
12	0.697	0.701	0.683	0.697	0.697	0.798	0.697	0.848
13	0.709	0.735	0.701	0.731	0.729	0.731	0.737	0.755
14	0.698	0.698	0.699	0.698	0.725	0.734	0.729	0.748
15	0.722	0.715	0.706	0.709	0.781	0.783	0.760	0.772
16	0.706	0.713	0.651	0.713	0.728	0.765	0.720	0.783
17	0.692	0.709	0.710	0.701	0.721	0.721	0.733	0.733
18	0.703	0.726	0.704	0.719	0.720	0.764	0.734	0.754
19	0.639	0.709	0.661	0.708	0.749	0.734	0.731	0.770
20	0.695	0.719	0.711	0.728	0.731	0.774	0.729	0.774
21	0.713	0.714	0.711	0.712	0.744	0.744	0.724	0.732
22	0.658	0.706	0.658	0.715	0.711	0.735	0.710	0.727
23	0.658	0.715	0.670	0.724	0.734	0.734	0.710	0.753
24	0.719	0.711	0.700	0.708	0.665	0.763	0.681	0.756
Median	0.697	*0.728*	0.700	*0.722*	0.715	*0.778*	0.710	*0.796*

From Table 5, it can be seen that the yes variables have a higher median value than no variables in the two recommendation modes (the less/more hints the better). This means that the use of CF (variable P_j) contributes to the recommendation of the LOs with the best rating for the student.

For each algorithm analysed, the differences between the yes and no variables (using or not using the predictive variable in the cost function) are statistically significant (*p-value* < 0.05). Table 6 summarises the results of the statistical analysis (*p-value*) and the magnitude of the statistically significant differences (*magnitude diff*).

Table 6 Estimated difference in average performance between the prediction variables (no/yes)

	Recommendation mode 1		*Recommendation mode 2*	
	Exact	*CWSC*	*Exact*	*CWSC*
p-value (Wilcoxon)	< 0.001	< 0.001	< 0.001	< 0.001
Magnitude diff (Tukey)	0.0428	0.0420	0.1069	0.1189

Note: If *p-value* < 0.05, then there is a statistically significant difference between the variables.

The gain in using the CF is greater with recommendation mode 2 (the more hints the better), with the magnitudes of the differences reaching the maximum value of 0.1, approximately. This difference demonstrates that the use of P (CF) in the calculation of the OF (c_j) increases on average 0.1 the average rate of the LORP solution, improving the quality of the LOs recommended to the learners. In recommendation mode 1 (the less hints the better), the increase was on average 0.04 and 0.05 in the average rate of the LORP solutions, relatively more modest gains.

Figure 9 Boxplot to compare the versions of each algorithm with (yes) and without (no) prediction (see online version for colours)

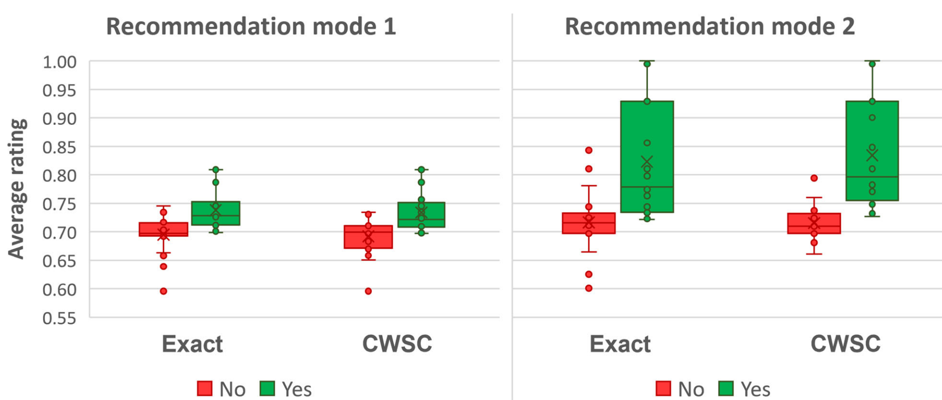


Figure 9 shows a boxplot of the average ratings of the solutions of the LORP. The red boxes represent the average rating without using the P (prediction) variable, i.e., the cost c_j is calculated by equation (6). On the other hand, green boxes represent the average rating using the P variable. In this case, the cost is calculated by equation (5). The graphs corroborate the data presented in Table 6, demonstrating more significant gains in recommendation mode 2. However, improvements can be seen in both recommendation modes and in the two algorithms evaluated with the use of CF. We can see that the algorithms implemented to solve the LORP using the prediction find solutions composed of LOs with better ratings, while not using the prediction decreases the quality of the solutions in relation to the rating.

5.4 Comparison of the versions of each algorithm with and without hint variable

To evaluate the *H* (hint) variable implemented in the proposed approach, we use the same strategy presented in Section 5.3. Two versions of each algorithm were implemented, and the difference between them relates to how the c_j cost is calculated. In the first, the cost is calculated by equation (7), which does not use the *H* variable, while in the second, the cost is calculated by equation (5), which uses the hint variable.

$$c_j = diss\left(O_{ideal}, O_j\right) + \left(1 - P_j^L\right) * \max_{j \in \{1,\ldots,n\}} diss\left(O_{ideal}, O_j\right) \tag{7}$$

The two versions of each algorithm are compared in Table 7. In the no columns, c_j is calculated by equation (7), and in the yes columns, c_j is calculated by equation (5). In Table 7, each value corresponds to an average number of hints of the ten runs (replications).

From Table 7, it can be seen that the yes variables have a lower median value than no variables in all algorithms when the less hints the better. On the other hand, yes variables have a higher median value than no variables in all algorithms when the more hints the better. These differences are statistically significant, as shown in Table 8. Table 8 shows the *p-value* and magnitude of the differences between the analysed samples.

In the recommendation model 1 (less hints the better), the magnitudes are negative, demonstrating that the number of hints returned when *H* is applied to the OF is smaller (which is expected in this modality). In the two algorithms, on average, approximately three less hints are presented to the learner when *H* is applied in the OF.

In the recommendation model 2 (more hints the better), the magnitudes are positive, demonstrating that the number of hints returned when *H* is applied to the OF is higher, as expected in this modality. In the two algorithms, on average, approximately three more hints are presented to the learner when *H* is applied in the OF.

The use of *H* (hint: fine-grained LOs) in the calculation of c_j in the OF improves the quality of LOs recommendation in relation to the expected number of hints. Thus, the results demonstrate that the RS will adapt better to the learners' needs, given the modality to be used at that moment, by returning more or less hints.

Figure 10 shows a boxplot of the average number of hints in the solution of the LORP. The red boxes represent the average number of hints without using the *H* (hint) variable, i.e., the cost c_j is calculated by equation (7). On the other hand, green boxes represent the average number of hints using the *H* variable. In this case, the cost is calculated by equation (5).

From boxplot in Figure 10, we can see that in the less hints the better mode, the algorithms tend to present no hints. The algorithms implemented to solve the LORP using the *H* variable find solutions with more hint-type LOs when the more hints the better, and they find solutions with less hints when the less hints the better, while not using the *H* variable decreases the quality of the solutions in relation to the expected number of hints.

Therefore, the variables P and H used in equation (5) contribute to the recommendation of solutions with the best rated LOs and with an appropriate number of hints (according to the selected recommendation mode), respectively.

Table 7 Comparison of the average number of hints in the solutions with (yes) and without (no) the H variable in solving the LORP

Instance	*Recommendation mode 1*				*Recommendation mode 2*			
	Exact		*CWSC*		*Exact*		*CWSC*	
	No	*Yes*	*No*	*Yes*	*No*	*Yes*	*No*	*Yes*
1	1	0	1	0	1	1	1	1
2	2	0	2	0	2	2	2	2
3	2	0	2	0	2	2	2	2
4	2	0	2	0	2	2	2	2
5	2	0	2	0	2	2	2	2
6	2	0	2	0	2	2	2	2
7	1	0	1	0	1	1	1	1
8	2	0	2	0	2	2	2	3
9	3	0	3	0	3	3	3	4
10	2	0	3	0	2	4	3	4
11	1	0	1	0	1	4	1	4
12	3	0	2	0	3	3	2	5
13	2	1	4	1	2	8	4	8
14	2	0	3	0	2	6	3	7
15	5	0	5	0	5	8	5	9
16	2	0	2	0	2	8	2	7
17	6	0	6	2	6	10	6	11
18	7	0	5	0	7	10	5	12
19	3	0	3	0	3	5	3	7
20	3	0	4	0	3	9	4	9
21	5	2	6	1	5	9	6	13
22	3	1	4	0	3	9	4	12
23	4	0	4	0	4	11	4	12
24	3	0	3	0	3	5	3	9
Median	2.0	*0.0*	3.0	*0.0*	2.0	*4.5*	3.0	*6.0*

Table 8 Estimated difference in average performance between the hint variables (no/yes)

	Recommendation mode 1		*Recommendation mode 2*	
	Exact	*CWSC*	*Exact*	*CWSC*
p-value (Wilcoxon)	< 0.001	< 0.001	0.001	< 0.001
Magnitude diff (Tukey)	–2.6667	–2.8333	2.4167	3.1667

Note: If *p-value* < 0.05, then there is a statistically significant difference between the variables.

Figure 10 Boxplot to compare the versions of each algorithm with (yes) and without (no) the *H* (hint) variable (see online version for colours)

5.5 Comparison between algorithms

The algorithms are compared based on the OF in equation (1), and the c_j is calculated in terms of the variables *H* and *P*, according to equation (5).

Data under analysis from Table 9 do not have a normal distribution, so we used a median value to compare the exact and CWSC algorithms. The bottom of Table 9 shows that the OF values of the exact and CWSC algorithms are almost equal and equal in recommendation modes 1 and 2, respectively. However, the difference between the algorithms is statistically significant. Table 10 summarises the results of the statistical analysis (*p-value*) and the magnitude of the statistically significant differences (*magnitude diff*). The performance of the CWSC algorithm is 26.97% better in recommendation mode 1 (*magnitude diff* = 55.5991) compared to mode 2 (*magnitude diff* = 76.1392).

Based on the data, the exact algorithm is better than the CWSC, but the CWSC is much faster than the exact algorithm, so the choice of one or the other will depend on the needs of the educational context in which they will be used.

Figure 11 Comparison of algorithm runtimes (see online version for colours)

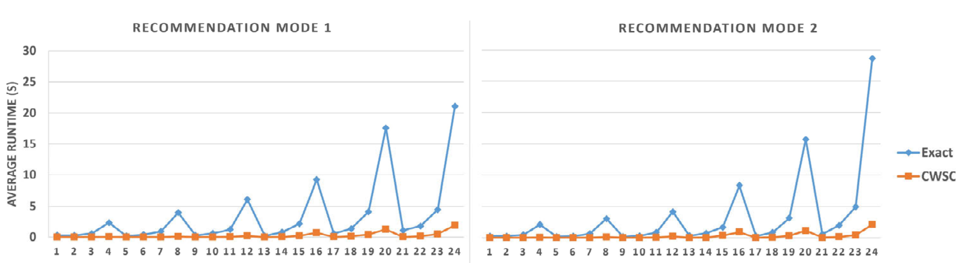


Figure 11 shows that CWSC has better average runtime than the exact algorithm in all instances. The downside of the exact algorithm is that it is very time consuming for larger instances. Therefore, the exact algorithm would be the best option for solving small instances of LORP, but if the shortest time is the priority, then CWSC is the best algorithm.

Table 9 Comparison between the OFs of the exact and CWSC algorithms

Instance	*Recommendation mode 1*		*Recommendation mode 2*	
	Objective function		*Objective function*	
	Exact	*CWSC*	*Exact*	*CWSC*
1	3.70	3.70	3.00	3.00
2	11.54	11.54	5.54	5.54
3	40.14	40.14	7.16	7.16
4	192.72	192.72	146.96	146.96
5	8.94	9.94	6.42	6.42
6	27.70	27.70	12.10	12.10
7	104.78	119.54	3.00	3.00
8	430.62	430.62	293.79	306.59
9	16.97	16.97	14.32	15.59
10	59.39	59.76	45.96	46.42
11	158.69	158.69	158.67	158.67
12	905.04	917.58	613.39	768.15
13	48.22	53.25	55.21	64.50
14	192.55	222.87	191.46	215.22
15	625.47	717.39	515.31	585.02
16	2,660.89	2,660.89	2,698.76	3,146.53
17	82.16	98.88	64.65	68.07
18	226.99	252.46	186.03	217.58
19	916.26	1,176.03	835.85	923.48
20	3,590.66	4,134.67	2,895.72	2,895.72
21	96.07	101.14	90.74	112.86
22	263.29	300.30	298.24	335.71
23	926.44	1,021.48	775.24	825.40
24	4145.79	4,341.13	4,349.17	5,224.31
Median	175.62	175.70	152.81	152.81

Table 10 Estimated difference in the values of the OF of the exact and CWSC algorithms

	Recommendation mode 1	*Recommendation mode 2*
	Objective function	*Objective function*
p-value (Wilcoxon)	< 0.001	< 0.001
Magnitude diff. (Tukey)	55.5991	76.1392

Note: If *p-value* < 0.05, then there is a statistically significant difference between the variables.

5.6 Discussion

The results presented in Section 5.3 show that the hypothesis derived from research question 1 is true, i.e., the use of CF (variable P_j) contributes to the recommendation of the LOs with the best rating for students confirming that the use of a track record of ratings as in Belizário et al. (2020) implies a recommendation of LOs that meet students' learning needs.

In addition, the findings presented in Section 5.4 confirm the hypothesis derived from research question 2, i.e., the use of H (hint: fine-grained LOs) in the calculation of c_j in the OF improves the quality of LOs recommendation in relation to the number of hints expected by students. These results demonstrate that our approach outperforms recommendation strategies that consider only the user's search parameters when recommending LOs (Belizário and Dorça, 2018; Falci et al., 2019) and those that combine the user's search parameters with CF (Belizário et al., 2020).

The results presented in Section 5.5 suggest that the use of exact and greedy algorithms to solve the LORP can be a good alternative in relation to the evolutionary algorithms implemented in previous works (Belizário and Dorça, 2018; Christudas et al., 2018; Birjali et al., 2018), corroborating the results presented by Falci et al. (2019), who created a greedy heuristic that is faster than GA, mainly for instances with thousands of LOs.

The algorithms implemented in our approach are used for the recommendation of online learning resources and LOs created by the teacher. E-learning resources that can be recommended include, among others, hints, lecture notes, exercises and tutorial videos. Even in the case of students having many generic resources available, in addition to the specific and fine-grained ones such as hints, the proposed recommendation algorithms recommend the LOs correctly by using two recommendation modes.

6 Conclusions and future work

In this paper, we propose a RS for recommending e-learning resources and fine-grained LOs (called hints) based on the student's learning style, knowledge and search parameters, which are modelled by an ontology. The challenge is the recommendation of LOs from different areas of knowledge, considering the refined concepts of ITSs. We faced this challenge by formulating the LORP as the SCP that belongs to the NP-hard class problems. Thus, the recommendation of LOs takes into account the concepts that the student needs to learn. In addition, we implement the hint type in an ontology for a more fine-grained recommendation of LOs, which is combined with the reuse of web content to overcome the low content diversity of ITSs and the lack of refined concepts in RSs.

Experiments were executed on a set of 24 benchmark instances created to simulate a real scenario. Experimental results showed that when the RS considers the variables P (CF) and H (fine-grained LOs) in calculating the cost of the solutions, the quality of the solutions improves in terms of the average rating and the number of expected hints. The exact and CWSC algorithms are good strategies to find these solutions for instances that simulate the educational context. If the best runtime is the priority, then the best algorithm is CWSC, but it finds the best solution in only 37.5% of instances.

This work has three limitations that we present below as challenges for future work. First, validation of the proposed approach in a real educational scenario. Experiments conducted with students are complex because they involve several variables, which, if properly controlled, can help to validate the proposed approach in the classroom context. Second, the proposed recommendation approach does not employ some recommendation strategies from related works that seem promising, such as sequential pattern mining for purposes of prediction. This can be a good strategy to further refine the objects delivered to students based on patterns found in the history of past educational resource recommendations. Third, the LORP is an NP-hard covering problem, so in addition to the greedy and exact algorithms presented in this paper, other algorithms based on constructive metaheuristics such as ACO, evolutionary algorithms as GA and local search should be explored by works to implement algorithms that are faster than exact and with better solutions than CWSC.

It is plausible to think that future work can overcome these limitations and challenges, given that the experimental results confirm the feasibility of our approach to be implemented in a real scenario. Our future research will focus on integrating the proposed RS into a learning environment such as Moodle for a more refined recommendation of LOs similar to what occurs in ITSs, considering content from different areas of knowledge. A practical application of the proposed RS is as a tool to solve frequent doubts of students from basic education to higher education in the different domains of knowledge, while teachers will be able to focus their time on solving the unusual questions that can only be answered by human creativity.

Acknowledgements

This study was financed in part by the Coordination for the Improvement of Higher Education Personnel – Brazil (CAPES) – Finance Code 001, and it was supported by the National Council for Scientific and Technological Development (CNPq; Grant 402431/2021-9).

References

Adomavicius, G. and Tuzhilin, A. (2005) 'Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 17, No. 6, pp.734–749.

Bajenaru, L., Borozan, A-M. and Smeureanu, I. (2015) 'Using ontologies for the e-learning system in healthcare human resources management', *Informatica Economica*, Vol. 19, No. 2, pp.15–24.

Balas, E. and Carrera, M.C. (1996) 'A dynamic subgradient-based branch-and-bound procedure for set covering', *Operations Research*, Vol. 44, No. 6, pp.875–890.

Barragáns-Martínez, A.B., Costa-Montenegro, E., Burguillo, J.C., Rey-López, M., Mikic-Fonte, F.A. and Peleteiro, A. (2010) 'A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition', *Information Sciences*, Vol. 180, No. 22, pp.4290–4311.

Beasley, J.E. and Chu, P.C. (1996) 'A genetic algorithm for the set covering problem', *European Journal of Operational Research*, Vol. 94, No. 2, pp.392–404.

Belizário Jr., C.F. and Dorça, F. (2018) 'Uma abordagem para a criação e recomendação de objetos de aprendizagem usando um algoritmo genético, tecnologias da web semântica e uma ontologia', in *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, SBC, Fortaleza, CE, Vol. 29, pp.1533–1542.

Belizário Jr., C.F., Dorça, F., Andrade, A.V. and Assis, L.P. (2020) 'Avanços na recomendação personalizada de objetos de aprendizagem através da utilização de meta-heurísticas clássicas associadas aos Problemas de Cobertura de Conjuntos e de Máxima Cobertura: Uma análise experimental', in *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, SBC, Natal, RN, Vol. 31, pp.1383–1392.

Bernacki, M.L., Aleven, V. and Nokes-Malach, T.J. (2014) 'Stability and change in adolescents' task-specific achievement goals and implications for learning mathematics with intelligent tutors', *Computers in Human Behavior*, Vol. 37, No. 1, pp.73–80.

Berners-Lee, T., Hendler, J. and Lassila, O. (2001) 'The semantic web', *Scientific American*, Vol. 284, No. 5, pp.34–43.

Bilal, N., Galinier, P. and Guibault, F. (2013) 'A new formulation of the set covering problem for metaheuristic approaches', *ISRN Operations Research*, Vol. 2013, No. 5, pp.1–10.

Birjali, M., Beni-Hssane, A. and Erritali, M. (2018) 'A novel adaptive e-learning model based on big data by using competence-based knowledge and social learner activities', *Applied Soft Computing*, Vol. 69, No. 1, pp.14–32.

Burke, R. (2007) 'Hybrid web recommender systems', *The Adaptive Web*, Vol. 4321, No. 1, pp.377–408.

Caprara, A., Fischetti, M. and Toth, P. (1999) 'A heuristic method for the set covering problem', *Operations Research*, Vol. 47, No. 5, pp.730–743.

Choi, K., Yoo, D., Kim, G. and Suh, Y. (2012) 'A hybrid online-product recommendation system: combining implicit rating-based collaborative filtering and sequential pattern analysis', *Electronic Commerce Research and Applications*, Vol. 11, No. 4, pp.309–317.

Christudas, B.C.L., Kirubakaran, E. and Thangaiah, P.R.J. (2018) 'An evolutionary approach for personalization of content delivery in e-learning systems based on learner behavior forcing compatibility of learning materials', *Telematics and Informatics*, Vol. 35, No. 3, pp.520–533.

Chvatal, V. (1979) 'A greedy heuristic for the set-covering problem', *Mathematics of Operations Research*, Vol. 4, No. 3, pp.233–235.

CLEOLab (2003) *CLEO Extensions to the IEEE Learning Object Metadata*, CLEO White Paper, CLEO Collaborative Partners – Cisco Systems, Inc., IBM Corporation, Microsoft Corporation, Thomson NETg.

De Medio, C., Limongelli, C., Sciarrone, F. and Temperini, M. (2020) 'MoodleREC: a recommendation system for creating courses using the moodle e-learning platform', *Computers in Human Behavior*, Vol. 104, No. 1, p.106168.

Deborah, L.J., Baskaran, R. and Kannan, A. (2014) 'Learning styles assessment and theoretical origin in an e-learning scenario: a survey', *Artificial Intelligence Review*, Vol. 42, No. 4, pp.801–819.

Falci, S.H., Dorça, F.A., Falci, D.H.M. and Vivas, A. (2019) 'A low complexity heuristic to solve a learning objects recommendation problem', in *2019 IEEE 19th ICALT*, IEEE, Vol. 19, pp.49–53.

Felder, R.M., Silverman, L.K. et al. (1988) 'Learning and teaching styles in engineering education', *Engineering Education*, Vol. 78, No. 7, pp.674–681.

Fisher, M.L. and Kedia, P. (1990) 'Optimal solution of set covering/partitioning problems using dual heuristics', *Management Science*, Vol. 36, No. 6, pp.674–688.

Garey, M.R. and Johnson, D.S. (1979) *Computers and Intractability. Series of Books in the Mathematical Sciences*, p.340, W.H. Freeman & Co., New York.

Golab, L., Korn, F., Li, F., Saha, B. and Srivastava, D. (2015) 'Size-constrained weighted set cover', in *2015 IEEE 31st Intern. Conf. on Data Engineering*, IEEE, pp.879–890.

Graf, S., Ives, C. et al. (2010) 'A flexible mechanism for providing adaptivity based on learning styles in learning management systems', in *2010 10th IEEE International Conference on Advanced Learning Technologies*, IEEE, pp.30–34.

Gruber, T.R. (1993) 'A translation approach to portable ontology specifications', *Knowledge Acquisition*, Vol. 5, No. 2, pp.199–220.

Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M. et al. (2004) 'SWRL: a Semantic Web Rule Language combining OWL and RuleML', *W3C Member Submission*, Vol. 21, No. 79, pp.1–31.

Housos, E. and Elmroth, T. (1997) 'Automatic optimization of subproblems in scheduling airline crews', *Interfaces*, Vol. 27, No. 5, pp.68–77.

Kennedy, J. and Eberhart, R. (1995) 'Particle swarm optimization', in *Proceedings of ICNN'95 – International Conference on Neural Networks*, IEEE, Vol. 4, pp.1942–1948.

Lan, G., DePuy, G.W. and Whitehouse, G.E. (2007) 'An effective and simple heuristic for the set covering problem', *European Journal of Operational Research*, Vol. 176, No. 3, pp.1387–1403.

Limongelli, C., Gasparetti, F. and 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)*, IEEE, pp.1–6.

LTSC (2002) *Standard for Learning Object Metadata (IEEE 1484.12.1)*, Learning Technology Standards Committee [online] https://webpages.uidaho.edu/fritz/edui/LOM_1484_12_1_v1_Final_Draft.pdf (accessed 2 May 2022).

McGregor, A. and Vu, H.T. (2019) 'Better streaming algorithms for the maximum coverage problem', *Theory of Computing Systems*, Vol. 63, No. 7, pp.1595–1619.

Mitchell, S., OSullivan, M. and Dunning, I. (2011) *PuLP: A Linear Programming Toolkit for Python*, Vol. 65, The University of Auckland, Auckland, New Zealand.

Montgomery, D.C. (2012) *Design and Analysis of Experiments*, p.730, John Wiley & Sons, New Jersey, USA.

Moreno, A., Valls, A., Isern, D., Marin, L. and Borràs, J. (2013) 'Sigtur/e-destination: ontology-based personalized recommendation of tourism and leisure activities', *Engineering Applications of Artificial Intelligence*, Vol. 26, No. 1, pp.633–651.

Musliu, N. (2006) 'Local search algorithm for unicost set covering problem', in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, Springer, pp.302–311.

Ouf, S., Ellatif, M.A., Salama, S.E. and Helmy, Y. (2017) 'A proposed paradigm for smart learning environment based on semantic web', *Computers in Human Behavior*, Vol. 72, pp.796–818 [online] https://doi.org/10.1016/j.chb.2016.08.030.

Pereira, C.K., Campos, F., Ströele, V., David, J.M.N. and Braga, R. (2018) 'BROAD-RSI – educational recommender system using social networks interactions and linked data', *Journal of Internet Services and Applications*, Vol. 9, No. 7, pp.1–28.

Rector, A. (2005) *Representing Specified Values in OWL: 'Value Partitions' and 'Value Sets'*, Technical Report Note 17, May, W3C, Semantic Web Best Practices and Deployment Working Group [online] http://www. w3.org/TR/swbp-specified-values/ (accessed 2 May 2022).

Ren, Z., Feng, Z., Ke, L. and Chang, H. (2008) 'A fast and efficient ant colony optimization approach for the set covering problem', in *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, IEEE, pp.1839–1844.

Ren, Z-G., Feng, Z-R., Ke, L-J. and Zhang, Z-J. (2010) 'New ideas for applying ant colony optimization to the set covering problem', *Computers & Industrial Engineering*, Vol. 58, No. 4, pp.774–784.

Ruotsalo, T., Haav, K., Stoyanov, A., Roche, S., Fani, E., Deliai, R., Mäkelä, E., Kauppinen, T. and Hyvönen, E. (2013) 'SMARTMUSEUM: a mobile recommender system for the web of data', *Journal of Web Semantics*, Vol. 20, No. 1, pp.50–67.

Shishehchi, S., Banihashem, S.Y., Zin, N.A.M., Noah, S.A.M. and Malaysia, K. (2012) 'Ontological approach in knowledge based recommender system to develop the quality of e-learning system', *Australian Journal of Basic and Applied Sciences*, Vol. 6, No. 2, pp.115–123.

Solar, M., Parada, V. and Urrutia, R. (2002) 'A parallel genetic algorithm to solve the set-covering problem', *Computers & Operations Research*, Vol. 29, No. 9, pp.1221–1235.

Soloman, B.A. and Felder, R.M. (2005) *Index of Learning Styles Questionnaire*, Vol. 70, NC State University [online] http://www.engr.ncsu.edu/learningstyles/ilsweb.html (accessed 14 May 2010).

Soofi, A.A. and Ahmed, M.U. (2019) 'A systematic review of domains, techniques, delivery modes and validation methods for intelligent tutoring systems', *International Journal of Advanced Computer Science and Applications*, Vol. 10, No. 3, pp.99–107.

Tarus, J.K., Niu, Z. and Kalui, D. (2018) 'A hybrid recommender system for e-learning based on context awareness and sequential pattern mining', *Soft Computing*, Vol. 22, No. 8, pp.2449–2461.

Tarus, J.K., Niu, Z. and Yousif, A. (2017) 'A hybrid knowledge-based recommender system for e-learning based on ontology and sequential pattern mining', *Future Generation Computer Systems*, Vol. 72, No. 1, pp.37–48.

Tukey, J.W. (1949) 'Comparing individual means in the analysis of variance', *Biometrics*, Vol. 5, No. 2, pp.99–114.

Vanetti, M., Binaghi, E., Carminati, B., Carullo, M. and Ferrari, E. (2010) 'Content-based filtering in on-line social networks', in *International Workshop on Privacy and Security Issues in Data Mining and Machine Learning*, Springer, pp.127–140.

VanLehn, K. (2006) 'The behavior of tutoring systems', *International Journal of Artificial Intelligence in Education*, Vol. 16, No. 3, pp.227–265.

Vasko, F.J. and Wilson, G.R. (1984) 'Using a facility location algorithm to solve large set covering problems', *Operations Research Letters*, Vol. 3, No. 2, pp.85–90.

Vasko, F.J., Wolf, F.E. and Stott, K.L. (1987) 'Optimal selection of ingot sizes via set covering', *Operations Research*, Vol. 35, No. 3, pp.346–353.

Wang, R-L. and Okazaki, K. (2007) 'An improved genetic algorithm with conditional genetic operators and its application to set-covering problem', *Soft Computing*, Vol. 11, No. 7, pp.687–694.

Warncke-Wang, M., Cosley, D. and Riedl, J. (2013) 'Tell me more: an actionable quality model for Wikipedia', in *Proceedings… International Symposium on Open Collaboration*, ACM, New York [online] https://doi.org/10.1145/2491055.2491063.

Yagiura, M., Kishida, M. and Ibaraki, T. (2006) 'A 3-flip neighborhood local search for the set covering problem', *European Journal of Operational Research*, Vol. 172, No. 2, pp.472–499.

Zhao, X., Niu, Z., Chen, W., Shi, C., Niu, K. and Liu, D. (2015) 'A hybrid approach of topic model and matrix factorization based on two-step recommendation framework', *Journal of Intelligent Information Systems*, Vol. 44, No. 3, pp.335–353.