Internet of Things 25 (2024) 101052

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

Exploring the solution space for adaptive curriculum sequencing: Study of a multi-objective approach

João Vítor de Castro Martins Ferreira Nogueira, Heder Soares Bernardino *, Jairo Francisco de Souza, Luciana Brugiolo Gonçalves, Stênio Sã Rosário Furtado Soares

Federal University of Juiz de Fora, Juiz de Fora, Minas Gerais, 36036-900, Brazil

ARTICLE INFO

Keywords:
Adaptive curriculum sequencing
Genetic algorithm
Multi-objective optimization
NSGA-II

ABSTRACT

Adaptive Curriculum Sequencing (ACS) is an important issue in personalized learning. In ACS problems, one desires the best sequence of learning materials that meet the profile of a given student. To do so, multiple features of the students and the materials used are necessary to generate good solutions. In fact, understanding the students' goals, motivation, and preferences is not an easy task and, consequently, different Internet of Things (IoT) approaches to gather this information during the learning process have been proposed. Actually, some works from the literature consider five objectives and, in this case, one has a many-objective optimization problem. Instead of solving the optimization problem considering the multiple objectives individually, the usual approach is to obtain solutions for a weighted sum of the objective values using search approaches for mono-objective optimization problems. However, this kind of approach may bias the search and limits the capacity of finding good results. Here, we solve the multi-objective ACS problem considering five objective functions. NSGA-II, a well-known Genetic Algorithm for multi-objective optimization problems, was used. In addition, the aggregation trees were employed to reduce the number of objectives to two and three due to the large number of objectives in the original problem. ACS problems from the literature were used to comparatively evaluate the proposed methods and the results obtained were compared to those found by the traditional approach of summing the objective values. According to these results, the best curriculum sequences were reached when using the proposal.

1. Introduction

Web-based learning environments have been facing an increase in demand in the latest years. That phenomenon occurs due to the advantages of distance learning modality compared to traditional classroom courses, such as its flexible schedule, affordable price range, and broad accessibility [1]. The effective use of e-learning systems solutions that aim for student satisfaction, as distance education modality suffers from the lack of commitment that may cause failures and drop-outs [2]. However, the growth in the use of educational technology and different formal and informal learning contents available on the internet create a huge amount of information that can lead students and instructors to cognitive overload and disorientation [3].

The Adaptive Learning and Intelligent Tutoring System (ITS) literature shows that the learning environment must be aware of learners' attributes, such as background, needs, intents, and preferences to provide easy and effective understanding [1,4,5].

* Corresponding author.
E-mail address: heder@ice.ufjf.br (H.S. Bernardino).

https://doi.org/10.1016/j.iot.2023.101052
Received 1 September 2023; Received in revised form 2 December 2023; Accepted 22 December 2023
Available online 8 January 2024

Adaptive Curriculum Sequencing (ACS) is considered a crucial issue in personalized learning as its purpose is to find the best sequence of learning materials that meet the student profile [6–8]. The challenge of ACS lies in automating the process, since unsuitable sequences may not offer learning materials that meet the student's learning goals and profile, leading to an increase in failure and dropout rates [9].

In [10], it is remarked on the need to consider multiple characteristics of the students and the materials used to generate good sequences. It may not be adequate to model the ACS problem using only one single objective due to the complexity of learning materials in the education system [11]. In fact, it is not trivial to understand student goals, motivation, or preferences, and researchers have proposed different IoT approaches to gather this information during the learning process [12,13]. This information is crucial for improving learning experiences, and ACS approaches make use of user profiles and a preference model to generate good personalized solutions. For instance, according to surveys in [14,15], personalization is important in IoT applications for learning, including personalized content, curricula, feedback, and learning strategies.

In the context of ACS approaches, each kind of student information is generally used as input for objective function to achieve a better-personalized solution. On the other side, most recent ACS approaches still perform single-objective optimization using an aggregated objective function. We argue that multi-objective ACS approaches can help both students and teachers by (i) reducing the cognitive workload when learning materials are recommended, (ii) allowing better personalizing when data from different sources – such as student, class, course, or learning material – can be modeled as a function and added to the meta-heuristic, and (iii) increasing motivation when personalized learning paths are created.

Given the previously mentioned problems, the contributions of this paper are as follows: we proposed the use of an aggregation tree to join the objectives, which lowers the complexity of the problem. The results of the aggregations are used in NSGA-II and compared with GA and between the aggregations in NSGA-II, considering the sum of the objectives, the domination of the solutions, the multi-objective metrics, and the Pareto front graphics, for each instance and student. The results of the aggregation tree show two good aggregation possibilities: considering aggregating the 5 objectives into two or three objectives. The results show that NSGA-II considering both the aggregations have better results than GA for a higher number of materials, and aggregating three objectives have better results than two objectives when evaluating three and five objectives.

The analysis of ACS approaches often overlooks the effect of individual objective functions in the searching process. Modern educational recommendation systems have been making use of more and more kinds of data, from system logs to sensor data, so handling multiple objective functions is mandatory in those systems. This study shed light on an important optimization issue with a great impact on decision-making in educational applications.

The remainder of this paper is organized as follows: Section 2 reviews related works; Section 3 describes the problem addressed by this work and the modeling that we adopted; Section 4 presents and describes the algorithms and the methods that were used; Section 5 shows the experiments carried out; Section 6 describes the results obtained; finally, Section 7 concludes the paper.

2. Related work

Since a single objective may not be enough to represent all the requirements needed by the student, the problem is often modeled as a multi-objective optimization problem [8]. Several works model sequencing as a problem of minimizing a set of objective functions [16–18]. Each function describes a different characteristic that the sequence needs to meet.

However, the problem is often simplified as a minimization problem of a combination of the functions [7,11,16,19–22]. In [11,16,20] is mentioned the importance of using multiple objectives, but each individual is evaluated using a sum of objectives. In [20], the authors use a combination of linear scalarizing approach and compromise programming to allow students to define objective weights as a means of expressing their preferences. In [21] the optimized objective is the product of the time requirement and the difficulty of the materials selected. Thus, despite considering multiple aspects of the generated sequences, the solutions obtained do not allow prioritizing which characteristic is most important to be attended to. Also, works rarely discuss how each objective function contributed to the quality of the final solution.

Some works propose the use of metaheuristics adapted to problems with multiple objectives [23,24]. In [23] it is proposed a variation of GA that divides the population into groups. Each group optimizes a single objective function and the diversity is maintained by a migration process. A version of PSO using an archive is proposed in [24]. The authors prove the convergence of the method and use it to address the sequencing problem, although the results are not presented. In each case, the authors do not compare their results with other methods. They also do not show how multi-objective methods compared with the simplest approaches that aggregate all the objectives. It is also not discussed a strategy to deal with a large number of objective functions that may be required to model the sequencing problem.

3. Adaptive curriculum sequencing

The Adaptive Curriculum Sequencing problem is usually modeled as a constraint satisfaction [25,26] or a multi-objective optimization problem [24,27]. Regarding the multi-objective optimization approach, the definition of the objective functions depends on how the problem is represented, and three elements are important to be modeled: the student, the learning material information, and the concept map. In [10], the authors present a mapping of the main characteristics used to describe students and materials for ACS and describe two main ways to model students' characteristics: using intrinsic or extrinsic parameters.

Intrinsic parameters are characteristics belonging to students and having no relation to external information. The main intrinsic parameters are preference, learning style (LS), intention, psychology profile, and attitude. Learning styles are the most used preference model in ACS approaches [28], such as Unified LS Model (ULSM) [29], Felder and Silverman LS Model (FSLSM) [30], and the model in [31]. Extrinsic parameters are external characteristics related to the information of an environment or knowledge domain. The main extrinsic parameters used by articles on ACS problems are knowledge level, competency, and time availability. The most used is knowledge level, which refers to the knowledge that a student has about some topic or domain. The most common way to obtain knowledge level is from pre-tests [24,32,33]. Other variables are usually used in ACS approaches, such as difficulty, type, format, competency, and learning time.

For the concept map, there are three main ways to define it: approximated from mathematical models; created by specialists; or generated from ontologies. The mathematical methods are used to approximate the concept map automatically and decentrally [23, 34–36]. However, it is necessary to evaluate the use of mathematical methods in the ACS problem because these methods ignore the relationships between concepts, potentially resulting in the creation of concept maps with illogical sequences [34]. The construction of concept maps based on expert experience is a widely recognized and widely utilized approach, but it still has some disadvantages since it is costly labor, depends on the experience of those involved, and is not flexible for students [37]. Finally, the ontology-based approach aims to associate semantics with the structure of concepts. It is important to note that the automatic construction of concept maps is a relevant issue within the field of adaptive learning [38].

3.1. Modeling

In this work, the ACS problem is defined as a multi-objective optimization problem. A candidate solution to the ACS problem is a vector $\mathbf{x} = (x_1, x_2, \dots, x_{|M|})$, where M is the set of learning materials, $|M|$ the number of materials, and x_i is a binary variable that indicates if the ith learning material is part of the sequence generated for the student. The objectives used here are the same ones used in our previous work [37]: concept coverage (O_1), students ability (O_2), course duration (O_3), materials balancing between concepts (O_4), and adequacy to the learning style (O_5). These objectives were chosen as they include the main features present in other works [10] and are defined as

$$\begin{aligned} O_1(\mathbf{x}) =& (1-\rho)(|R(\mathbf{x})| - |R(\mathbf{x}) \cap E|) \\ &+ \rho(|E| - |R(\mathbf{x}) \cap E|) \end{aligned} \tag{1}$$

$$O_2(\mathbf{x}) = \sum_{i=1}^{|M|} x_i \left| D^{m_i} - \frac{\sum_{j=1}^{|C|} R_{c_j}^{m_i} E_{c_j} H_{c_j}}{\sum_{j=i}^{|C|} R_{c_j}^{m_i} E_{c_j}} \right| \frac{1}{\sum_{i=1}^{|M|} x_i} \tag{2}$$

$$\begin{aligned} O_3(\mathbf{x}) =& \max\left(T^{\downarrow} - \sum_{i=1}^{|M|} T^{m_i} x_i, 0\right) \\ &+ \max\left(0, \sum_{i=1}^{|M|} T^{m_i} x_i - T^{\uparrow}\right) \end{aligned} \tag{3}$$

$$O_4(\mathbf{x}) = \sum_{j=1}^{|C|} E_{c_j} \left| \sum_{i=1}^{|M|} x_i R_{c_j}^{m_i} - \frac{\sum_{i'=1}^{|M|} \sum_{j'=1}^{|C|} x_{i'} R_{c_{j'}}^{m_{i'}} E_{c_{j'}}}{\sum_{j'=1}^{|C|} E_{c_{j'}}} \right| \tag{4}$$

$$O_5(\mathbf{x}) = \sum_{k=1}^{4} \frac{\sum_{i=1}^{|M|} x_i \left| 3sgn(\theta_k^{m_i}) - L_k \right|}{4\sum_{i=1}^{|M|} x_i} \tag{5}$$

where C is the set of concepts of the course, $m_i \in M$ is the ith learning material, $c_j \in C$ is the jth concept, each material m_i has a set of concepts R^{m_i} formed by one or more concepts $R_{c_j}^{m_i} \in C$, and each student has a set of concepts E that are their learning goals. The set of concepts (C) is composed of all the concepts of the course, but each student can have different learning goals and, thus, each student has an individual set of concepts (E). Ideally, a student's learning sequence materials have materials that teach only those concepts that are part of that student's goals.

Objective $O_1(\mathbf{x})$ checks whether the course concepts covered by the selected sequence meet the student's learning goals. This function associates penalties to the number of spare concepts (first part of the equation) and the number of missing concepts (second part of the equation) according to the student's learning goals. $\rho \in [0, 1]$ is a parameter to choose whether to give more emphasis on covering every learning goal or not overloading the student. $R(\mathbf{x})$ represents the set of concepts covered by all learning materials present in vector $\mathbf{x}$ and can be defined as:

$$R(\mathbf{x}) = \bigcup_{i=1}^{|M|} \{R^{m_i} | x_i = 1\} \tag{6}$$

The learning material difficulty is compared in $O_2(\mathbf{x})$ to the mean of the student's ability in the concepts addressed by such learning material and that are included in the student's learning goals, where D^{m_i} represents the difficulty associated to a learning material m_i, $R_{c_j}^{m_i}$ indicates whether the learning material covers a concept c_j. Thus, $R_{c_j}^{m_i} = 1$ if the learning material cover the concept c_j and $R_{m_i}^{c_j} = 0$ otherwise. E_{c_j} indicates whether a concept c_j is in accordance with the student's learning goals. Thus, $E_{c_j} = 1$ if the concept c_j is expected by the student, and $E_{c_j} = 0$, otherwise. Finally, H_{c_j} represents the current ability level of the student concerning the concept c_j.

Objective $O_3(\mathbf{x})$ represents the deviation of the time estimated and that expected by the student for the course. Thus, $O_3(\mathbf{x})$ checks whether the total time of the course is between the lower and upper limits, where $T^{\downarrow}$ represents the lower and $T^{\uparrow}$ the upper bounds times expected by the student. Besides, T^{m_i} represents the estimated learning time of a learning material m_i (in hours).

The balancing of the selected learning materials is evaluated in $O_4(\mathbf{x})$. This function returns a low value as more concepts are covered by a similar amount of learning materials.

Finally, $O_5(\mathbf{x})$ measures the preference of the user for the material type where the preference is modeled as a 4-dimensional vector. The preference vectors can be calculated using some techniques such as questionnaires, and explicit or implicit feedback methods. In $O_5(\mathbf{x})$, for each dimension $k \in \{1, \ldots, 4\}$, $\theta_k^{m_i}$ indicates which preference is attended by the ith learning material, and $L_k \in [-3, 3]$ represents the student intensity for this dimension [7].

In this work, the student's preferences are defined by the student's learning style and it is linked to the characteristics of the learning materials according to the Felder and Silverman Learning Style Model (FSLSM) [30]. As described in our previous work [37], the learning style model was used as a proof of concept. The learning style is used only as an intrinsic student characteristic, describing a student's preferences. Other characteristics of the student could be used to replace this objective.

In this model, the student is characterized in four dimensions representing different aspects of how the student learns best, with values representing the intensity of the characteristics as follows: in the dimension Processing, the student may prefer materials that are more reflexive or more active; in the dimension Perception, it describes if the student is more intuitive or more sensing; in the dimension Input, the values represent if the student is more verbal or more visual; and, finally, the dimension Understanding specify if the students prefer to be explained using a global approach or a more sequential approach.

The multi-objective problem is then defined as the minimization of the values of the objective vector $f(x) = [O_1(x), O_2(x), O_3(x), O_4(x), O_5(x)]$. As the evaluated objectives can be conflicting, it is not possible to find a single answer to the problem. For example, one solution may be better suited to the student's time availability but it does not provide a good result to cover all the concepts he expected to learn, while another solution may cover a large number of concepts but it requires more time for the student. In this case, objective O_1 conflicts with objective O_3.

Thus, a set of solutions defined in terms of Pareto dominance should be considered. In a multi-objective optimization problem composed of n objectives to be minimized, a solution $\mathbf{x}'$ dominates a solution $\mathbf{x}''$ if $O_k(\mathbf{x}') \leq O_k(\mathbf{x}'')\ \forall k \in \{1, \ldots, n\}$ and $\exists l \in \{1, \ldots, n\} | O_k(\mathbf{x}') < O_k(\mathbf{x}'')$. In this case, the solution $\mathbf{x}''$ is dominated by $\mathbf{x}'$. If there is no other solution that dominates $\mathbf{x}'$, then $\mathbf{x}'$ is called a Pareto optimal solution. The set composed of all the Pareto optimal solutions is called the Pareto front.

3.2. Multi-objective approach

In a multi-objective optimization problem, one desires to find an approximation of the Pareto front. This approximation of the Pareto front is composed of non-dominated solutions.

Several heuristics used to approximate solutions to optimization problems are not able to deal with multi-objective problems. Most works that use evolutionary approaches are only able to deal with the problem considering a single objective [10]. In these cases, the most common way of adapting the problem is to consider the sum of all objectives as a single objective function defined as $f'(\mathbf{x}) = \sum_{i=1}^{n} \omega_i O_i(\mathbf{x})$, where n is the number of objectives of the original problem, and ω_i is the weight of the ith objective function.

The limitation of this approach is that considering a single set of weights ω, only a single solution is found. An alternative is solving the problem for several weight values, but that requires further changes in the algorithms used. Also, this approach is not able to find all solutions in a non-convex solution space.

Multi-objective problems having more than three objectives are referred to as many-objective optimization. These optimization problems are known to cause issues with evolutionary computation algorithms, such as NSGA-II [39], as the Pareto front size tends to grow exponentially with the number of objectives.

To address this problem, we reduced the number of objectives in the problem using the so-called aggregation trees [40]. As a result, search methods designed for multi-objective optimization problems can be used.

4. Methods

The goal of a multi-objective optimization problem is to find a set of solutions that approximates the Pareto front. Some methods use stochastic processes to calculate a single solution to the problem and try to find the Pareto front through multiple executions. With each new execution, a solution is found and, if it is not dominated by any solution already found, it is added to the final set. Any solutions that are dominated by the new solution are also removed from the set.

However, population-based metaheuristics such as GA can be adapted to solve multi-objective problems. This kind of metaheuristic can be changed to extract the non-dominated individuals from the final population and use this set as a result to the problem rather than returning only the best individual in the population.

NSGA-II is an algorithm proposed in [41] for solving multi-objective optimization problems. It is a GA variant that uses the concepts of dominance. Considering the discussion in Section 3.2, NSGA-II is not effective when the optimization problem contains a large number of objectives [39]. Here, the aggregation trees (Sections 3.1 and 4.3) are used to reduce the number of objectives. Thus, removing objectives is not necessary and search techniques for multi-objective optimization problems, such as NSGA-II, can be adopted.

Here, we propose using aggregation trees to generate a small number of objectives for the ACS problem by combining the original ones. Thus, search approaches for multi-objective optimization can be adopted. The proposed procedure is composed by

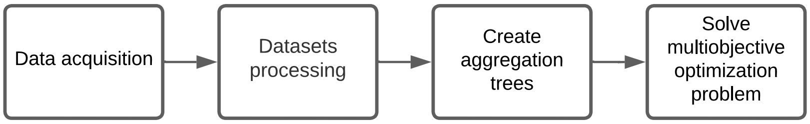

Fig. 1. Flowchart of the proposed procedure.

the following steps: (i) data acquisition, in which the data of the course, materials, and students are collected from IoT devices or other ways, depending on the application; (ii) datasets processing, where the data is handled to be used in the optimization process; (iii) creating of the aggregation trees, by combining the objectives of the original optimization problem into two or three objectives; and (iv) solving the multi-objective optimization problem. The flowchart of this procedure is illustrated in Fig. 1.

GA and NSGA-II are described in Sections 4.1 and 4.2, respectively. Also, we detail the aggregation tree approach in Section 4.3.

4.1. Genetic algorithm

Genetic Algorithm (GA) [42] is a search technique inspired by Darwin's theory of evolution. Evolutionary concepts, such as generation, crossover, and mutation were used to design a method for solving optimization problems. In GA, a population of (randomly initialized) candidate solutions evolves through the selection of the fittest individuals (best candidate solutions) and with crossover and mutation operators.

The process starts with a set of individuals (Population), where each individual is a candidate solution to the problem. The population size is defined here as PS. In the parent selection, a set of individuals (parents) are chosen to perform crossover. We used the roulette wheel procedure, in which the parents are randomly selected according to the proportion of their fitness regarding the fitness values of the population. The parent individuals are recombined to generate new ones. Here, a uniform crossover is used, where the swapping of each gene of the parents occurs according to a random value sampled from a uniform distribution. In the sequence, a mutation is applied to the generated offspring. A mutation is a small modification in the generated candidate solutions where one of the bits in the bit string is flipped according to a Mutation Probability (MP). Mutation is important to maintain the diversity of the population and prevent premature convergence.

The new candidate solutions are evaluated (objective function) and the current population is replaced. It is common to keep the best individuals in the population. We adopted the replacement strategy suggested in [37], in which a percentage of both the best (TS) and worst (PW) individuals is kept in the population. The algorithm ends when a stop criterion is met. We used in this work a predefined computational budget as a stop criterion (a maximum number of objective function evaluations). A pseudo-code of GA is presented in Algorithm 1.

Algorithm 1 Genetic Algorithm

```
Result: Sequence of learning materials
Population ← InitializePopulation(PS, |M|)
CalculateObjectiveFunction(Population)
while ¬StopCondition() do
    Parents ← Selection(Population)
    NewOffSpring ← Crossover(Parents)
    Offspring ← Mutation(NewOffSpring, MP)
    CalculateObjectiveFunction(Offspring)
    Population ← Replacement(Population, Offspring, PS, TS, PW)
end while
```

4.2. Non-dominated sorting genetic algorithm II

NSGA-II [41] is a GA variant to solve multi-objective optimization problems. NSGA-II starts with a set of randomly generated solutions, and the crossover and mutation operators used here are the same adopted in GA. In NSGA-II, the parent and survival selections are performed using a fast non-dominated sorting algorithm and crowding distance.

In the non-dominated sorting approach, the population is ranked by dominance. The non-dominated individuals are in the first rank. The second rank is composed of the non-dominated individuals of the population but without those from the first rank. The procedure is performed up to the entire population is ranked. In the sequence, the crowding distances of the individuals in each rank are calculated. Crowding distance is the semi-perimeter of a hypercube formed by nearest neighbors (in the objectives space) of each individual, or infinity for the extreme candidate solutions. Those individuals in the smallest ranks and with the largest Crowding distance values are preferable.

In the standard selection for survival of NSGA-II, both current and offspring populations are combined and the best individuals are chosen for the next generation. Here, a tournament is used for parent selection. A pseudo-code of NSGA-II is presented in Algorithm 2.

Algorithm 2 NSGA-II

```
Result: Set of non-dominated solutions
Population ← InitializePopulation(PS, |M|)
CalculateObjectiveFunction(Population)
NonDominatedSortAndCrowdingDistance(Population)
while ¬StopCondition() do
    Parents ← Selection(Population)
    NewOffspring ← Crossover(Parents)
    Offspring ← Mutation(NewOffspring, MP)
    CalculateObjectiveFunction(Offspring)
    NGeneration ← Offspring ∪ Population
    NonDominatedSortAndCrowdingDistance(NGeneration)
    Population ← Replacement(NGeneration, PS)
end while
```

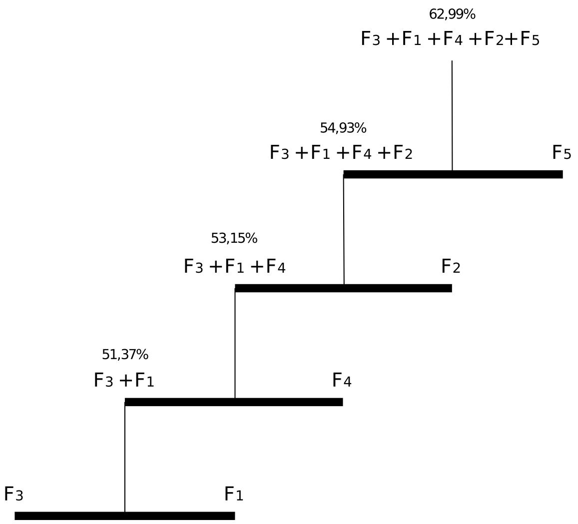

Fig. 2. Aggregation tree for a student in the 300-materials dataset.

4.3. Aggregation tree

Aggregation Trees [40] is an approach for dimension reduction of many-objective problems (concerning the objectives), such as the ACS problem solved here (the model is detailed in Section 3.1). The high dimensionality of such a type of problem makes it hard to represent the relationship between objectives and their solutions. Also, most search techniques from the literature are designed for optimization in lower dimensions. Thus, aggregation trees are appropriate when the number of objectives is large.

Aggregation trees iteratively combine sets of objectives such as an agglomerative hierarchical clustering technique. One can notice that pairs of objectives are analyzed at the first step (for forming a leaf node in the tree structure) and the procedure is concluded when a single set contains all the objectives (the root of the tree). Aggregation trees can use a harmony metric as the criterion for performing the aggregation of the sets of objectives. The Harmony metric is considered high when the solutions into two sets of objectives tend to have similar or proportional values. The aggregation is the summation of the values of the objectives in a given set.

This tree structure has the following properties: (i) leaf nodes represent objectives, where F_i is an objective function; and (ii) the remaining nodes represent a combination of objectives, which is the sum of the objectives in its child nodes. Fig. 2 illustrates the aggregation tree generated for the case of a student in the 300-materials dataset. In this example, $F3$ and $F1$ are aggregated and the parent node $F3+F1$ is generated. These objectives were selected as they are those with the highest conflict value between the pairs of objectives. One can observe that the internal nodes of Fig. 2 also contain these conflict values, that are conceptually the opposite of the harmony metric. Thus, $F3$ and $F1$ are aggregated with 51.37% of conflict, in this case. In addition, the values obtained by the summation of the objectives in each internal node are normalized for the evaluation of the harmony between the sets of objectives.

Aggregation trees make it possible for the search method to find good solutions when the number of objectives is large. However, one can observe that the aggregated objectives were not explored individually during the search.

5. Experiments

To conduct a comparative analysis of the outcomes attained through the use of NSGA-II for solving ACS problems, the findings derived from a prior study [37] are employed to contrast our results with those obtained when using single-objective methodologies. Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE), and Prey-Predator Algorithm (PPA) were evaluated in that work. Here, we compared our results with those obtained by the GA technique in [37] due to its similarity with NSGA-II. The experiments seek to examine the results concerning single- and multi-objective metrics.

Comparing different methods for ACS becomes a challenging task in the absence of established benchmarks. Unfortunately, many academic papers overlook comparisons with existing methods that have already been established in this research field. Additionally, most of the datasets are either private or lack sufficient descriptions, which makes it impossible to conduct controlled experiments that can be compared with each other.

The experiments were carried out using the datasets from [37], which are divided into two groups: real-life and synthetic data. The real-life dataset comprises 284 learning materials covering 30 Computer Science concepts across eight topics and 24 student profiles. The data were collected from a Computer Science Fundamentals course. This included course data, such as topics and concepts correlation, and goals; learning material data, such as concept coverage, duration, instructional media style, interactivity level, and content; and student data, such as access log, previous knowledge about each topic, and preference data. Student preference represents subjective information that can be retrieved from different data sources, such as monitoring student reactions to materials, stress level or emotion related to characteristics of the kind of media instructional design style, and more. In this dataset, a 4-dimensional vector is used for generalization. The synthetic datasets are useful for understanding the impact of the number of materials on ACS approaches, and they were created using the method proposed in [37] for generating synthetic data based on real-life data while maintaining the probability distribution of the original data. We used the synthetic datasets with 50, 300, and 1000 learning materials in the computational experiments.

NSGA-II has problems when solving many-objective optimization problems and the problem needs to be simplified to fewer objectives. The objective aggregation was performed by generating 1000 random solutions per instance pair, considering a given student and materials. For each pair, all the solutions were evaluated using the five objective functions, and the aggregation tree was generated. We have analyzed all the aggregation trees and selected the most frequent tree for all the cases analyzed in this study. In order to compare the impact of aggregation on solution quality, we conducted experiments using two and three objective functions.

A consequence observed when reducing the number of objectives in the original problem is that the employed methods exhibit a failure to effectively cover the original search space. To examine this aspect, the aggregation method outlined in Section 5.1 was employed. Therefore, the implementation of NSGA-II took into account the reduction of objectives to both 3 and 2. Furthermore, as a means of evaluating the consequences of reducing objectives, the Genetic Algorithm (GA) has been employed, which takes into account the aggregation of the initially proposed five objectives as a single one.

Given that the Pareto Boundary for the problem is currently unknown, the non-dominated solutions obtained from all algorithm executions for a specific instance were used to build the Pareto Boundary. Therefore, the Pareto Frontier of a given instance consists of the population that is not dominated by the results obtained by the Genetic Algorithm (GA), specifically the combination of NSGA-II with two objectives and NSGA-II with three objectives. Each algorithm was run 30 times, and the outcomes of all the executions were grouped to build the Pareto Frontier.

We compared the results obtained according to the sum of the objectives (Section 5.2), and considering the multiple objectives (Section 5.3). In the latter case, we evaluated metrics for multi-objective optimization, the proportion of the solutions of each algorithm into the Pareto set, the number of dominated solutions, and an illustration of the non-dominated solutions found by the search approaches.

All experiments were carried out on a computer with the Linux Operating System Ubuntu 18.04.6 LTS, with 11 GB and a processor with 8 cores using a thread per core. The following parameters were suggested in [37] for GA and used here: $PS = 10$, $TS = 0.15$, $MP = 0.01$, $PW = 0.1$, roulette wheel selection, uniform crossover, single bit mutation, and permissive replacement. In addition, the parameters adopted to NSGA-II for both aggregation cases were: $PS = 50$, $MP = 0.01$, tournament selection, uniform crossover, and single-bit mutation. For GA and NSGA-II techniques, a maximum number of objective function evaluations equal to 100,000 was allowed, and 30 independent runs were performed. All data, metaheuristics implementation and scripts used to run the experiments are publicly available.[1]

5.1. Aggregation tree

We used the Aggregation Tree method to identify the order of aggregation of objective functions. The algorithm was executed individually for each student and dataset of materials. Fig. 2 shows one of the generated aggregation trees. Although we have a tree for each instance, the tree for each pair is quite similar. Thus, Fig. 2 represents the common ground among all trees.

Two potential aggregations were found: a final aggregation with two objective functions and another with three objective functions. For two objectives the aggregation is $(O_5, O_1 + O_2 + O_3 + O_4)$, and for three objectives is $(O_5, O_2, O_1 + O_3 + O_4)$.

[1] https://github.com/lapic-ufjf/evolutionary-ACS-benchmark.

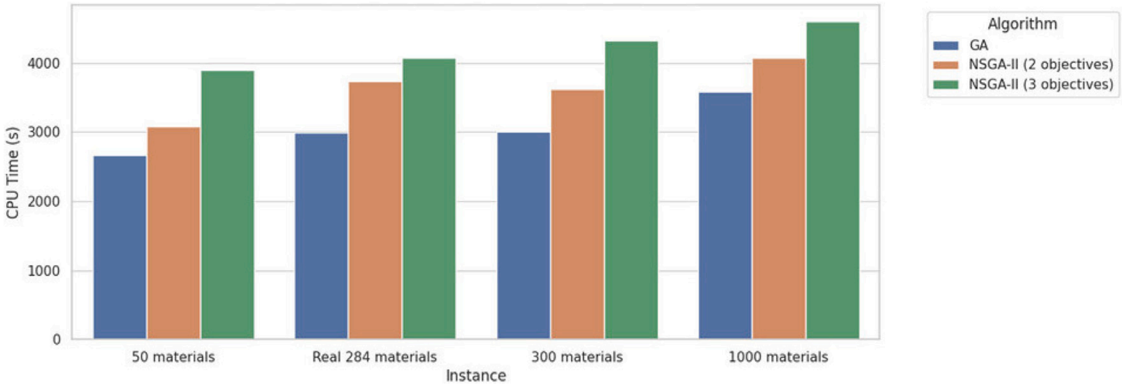

Fig. 3. CPU times (in seconds) for running GA, and NSGA-II for 2 and 3 objectives when solving the ACS problems considered here.

5.2. Comparison of the sum of objectives

A way to compare the various objectives is to face them as equally significant components of a larger macro objective, which is defined as the sum of all of these individual objectives. Here, results obtained by a GA, and NSGA-II optimizing the problem with two and three objectives are compared. To perform the comparison, the solutions with the highest sum of the objectives are considered in each independent run. The comparison is conducted about the number of NSGA-II executions that were able to find a superior solution when compared to GA.

The primary objective of this experiment is to assess the behavior of NSGA-II in a scenario that is highly favorable to a GA designed for single objective optimization problems. We investigate the impact of optimizing with multiple objectives on the efficiency when comparing the results with respect to the single aggregation of objectives.

Due to the utilization of its population for enhanced exploration of solution space, NSGA-II may encounter challenges in identifying solutions that strongly prioritize a singular approach for objective aggregation.

5.3. Comparison considering multiple objectives

The objective of the second experiment in this study is to evaluate the performance of the techniques regarding the multiple objectives considered here. This set of experiments considers two scenarios: the combination of objectives indicated in the Aggregation Tree method (NSGA-II for 2 or 3 objectives), and the solutions obtained through an evolutionary process that uses a fitness function calculated as the sum of these objectives (GA).

Initially, GA and NSGA-II variants are compared using metrics commonly adopted in multi-objective optimization. The following metrics were adopted: Inverted Generated Distance (IGD) [43,44] and Hypervolume [45]. The first metric measures the distance between the Pareto Front elements and the solutions obtained. The second performance metric measures the region between the solution set and a reference point. Thus, for Hypervolume, higher values indicate better results. While for the IGD, lower values indicate better results. Here, we defined the worst value for each objective in the solutions obtained by GA, NSGA-II with two objectives, and NSGA-II with three objectives as the reference point.

In the sequence, we checked the proportion of the Pareto set obtained by each search approach. This is important to evaluate the capacity of the algorithms to reach the Pareto set. The best methods are those that can find the larger number of individuals from the Pareto set.

Also, we analyze the dominant relationship between solutions produced by the NSGA-II approach. In this case, we evaluate the number of solutions obtained by a given search approach that dominates the solutions reached by another one. Finally, to illustrate this dominance, plots of the non-dominated solutions are presented.

6. Analysis of the results

We analyze in this section the results obtained by the search approaches considered here when solving four ACS problems. These analyses considered the sum of the objectives and metrics for multi-objective optimization problems. We also present plots of the non-dominated solutions obtained in a representative case in order to visually examine the results.

In addition, Fig. 3 shows the CPU times of the techniques used here when solving ACS problems with 50, 284 (real-world case), 300, and 1000 materials. One can observe that GA requires less CPU time to solve the problem and that there is no large variation in the CPU times when the size of the problem increases.

Table 1
Means and standard deviations of the percentage of solutions obtained by NSGA-II for both 2 and 3 objectives that are better than those achieved by GA concerning the sum of the five objectives.

Dataset	2 objectives	3 objectives
Real 284 materials	(37.78 ± 26.03)	(24.86 ± 19.05)
50 materials	(4.17 ± 9.49)	(9.31 ± 16.39)
300 materials	(71.53 ± 19.12)	(65.00 ± 34.22)
1000 materials	(98.33 ± 5.27)	(94.58 ± 8.54)

Table 2
Means and standard deviations of Hypervolume and IGD. These values were calculated using the best results found by GA. The reference points used to calculate the Hypervolumes are composed of the worst values in each objective considering the non-dominated solutions of the three approaches compared. NSGA-II 2 and NSGA-II 3 correspond to NSGA-II, respectively, for 2 and 3 objectives.

		2 objectives		3 objectives		5 objectives	
		Hypervolume	IGD	Hypervolume	IGD	Hypervolume	IGD
Real 284 materials	GA	0.032 ± 0.047	0.563 ± 0.102	0.041 ± 0.049	0.602 ± 0.103	0.047 ± 0.053	0.794 ± 0.080
	NSGA-II 2	**0.400** ± 0.112	**0.030** ± 0.014	0.182 ± 0.158	0.321 ± 0.093	0.176 ± 0.142	0.371 ± 0.090
	NSGA-II 3	0.318 ± 0.124	0.098 ± 0.034	**0.372** ± 0.107	**0.087** ± 0.021	**0.293** ± 0.091	**0.138** ± 0.021
50 materials	GA	0.022 ± 0.038	0.329 ± 0.048	0.025 ± 0.023	0.390 ± 0.073	0.014 ± 0.013	0.501 ± 0.082
	NSGA-II 2	**0.143** ± 0.068	**0.014** ± 0.007	0.071 ± 0.031	0.176 ± 0.057	0.036 ± 0.016	0.219 ± 0.063
	NSGA-II 3	0.128 ± 0.061	0.027 ± 0.015	**0.122** ± 0.062	**0.044** ± 0.011	**0.066** ± 0.027	**0.089** ± 0.014
300 materials	GA	0.023 ± 0.045	0.600 ± 0.086	0.030 ± 0.041	0.640 ± 0.104	0.041 ± 0.053	0.845 ± 0.106
	NSGA-II 2	**0.442** ± 0.140	**0.077** ± 0.063	0.183 ± 0.170	0.375 ± 0.143	0.191 ± 0.176	0.449 ± 0.159
	NSGA-II 3	0.396 ± 0.168	0.108 ± 0.043	**0.406** ± 0.094	**0.086** ± 0.020	**0.319** ± 0.105	**0.141** ± 0.026
1000 materials	GA	0.004 ± 0.013	0.997 ± 0.313	0.006 ± 0.027	0.920 ± 0.165	0.007 ± 0.025	1.133 ± 0.212
	NSGA-II 2	**0.535** ± 0.153	**0.075** ± 0.113	0.146 ± 0.153	0.479 ± 0.163	0.172 ± 0.173	0.537 ± 0.171
	NSGA-II 3	0.466 ± 0.150	0.141 ± 0.067	**0.499** ± 0.073	**0.116** ± 0.032	**0.397** ± 0.087	**0.173** ± 0.034

6.1. Analyzing the sum of objective functions

While it is not feasible to directly compare approaches for mono and multi-objective optimization, this experiment aims to assess the quality of solutions obtained by the NSGA-II method for multi-objective problems. Specifically, the evaluation focused on the fitness function used in the mono-objective approach, namely GA. Given that the genetic algorithm (GA) evaluates the fitness of individuals in the population based on the sum of the five objectives, it was expected that the comparison between the solutions produced by the NSGA-II (with two or three objectives) and GA demonstrates the advantage of using GA.

Table 1 presents the means and standard deviations of the percentage of solutions found by NSGA-II for both 2 and 3 objectives that are better than those reached by GA regarding the sum of the five objectives. The means and standard deviations are calculated with respect to the 24 students, and these results are shown for a real scenario, with 284 materials, and the other artificial scenarios (with 50, 300, and 1000 materials). One can notice that, in all scenarios, NSGA-II with both combinations of objectives achieved better solutions than GA. The high percentage of solutions with better quality than those obtained by GA in two of the four scenarios suggests that there is, in fact, a bias in the relevance of the objectives. The linear combination of the 5 objectives used by GA makes it hard to escape from low-quality suboptimals.

6.2. Analyses considering multiple objectives

6.2.1. Multi-objective metrics

IGD and Hypervolume were calculated for each student (24), number of materials (50, 300, 1000, real base), algorithm (GA, NSGA-II for 2 objectives, NSGA-II for 3 objectives), number of objectives (2, 3 and 5) and seed (30). This procedure generated a total of 25,920 results that were previously normalized. Table 2 shows means and standard deviations of IGD and Hypervolume for each base, algorithm, and number of objectives. In this first analysis, we consider a single solution obtained by GA in each independent run. In the sequence, we also evaluate the performance metrics regarding the non-dominated solutions in the final population of GA.

Both NSGA-II for 2 and 3 objectives obtained better IGD and Hypervolume values than the GA in all datasets and number of objectives. NSGA-II for 2 objectives obtained better IGD than NSGA-II for 3 objectives and similar values of Hypervolume in the evaluation for 2 objectives. However, NSGA-II for 3 objectives obtained better IGD and Hypervolume than NSGA-II for 2 objectives when we analyzed using 3 and 5 objectives.

The results using the non-dominated solutions from the final population of GA are similar to those observed when only the best solutions of GA are considered, as one can see in Table 3. The change of values was proportional, with Hypervolume increasing and IGD decreasing.

Table 3
Means and standard deviations of Hypervolume and IGD. These values were calculated using the final populations reached by GA. The reference points used to calculate the Hypervolumes are composed of the worst values in each objective considering the non-dominated solutions of the three approaches compared. NSGA-II 2 and NSGA-II 3 correspond to NSGA-II, respectively, for 2 and 3 objectives.

		2 objectives		3 objectives		5 objectives	
		Hypervolume	IGD	Hypervolume	IGD	Hypervolume	IGD
Real 284 materials	GA	0.043 ± 0.051	0.477 ± 0.109	0.050 ± 0.056	0.520 ± 0.112	0.059 ± 0.060	0.646 ± 0.131
	NSGA-II 2	**0.412** ± 0.120	**0.031** ± 0.013	0.191 ± 0.172	0.316 ± 0.098	0.194 ± 0.164	0.361 ± 0.098
	NSGA-II 3	0.335 ± 0.145	0.096 ± 0.037	**0.395** ± 0.131	**0.083** ± 0.025	**0.322** ± 0.122	**0.128** ± 0.029
50 materials	GA	0.038 ± 0.042	0.245 ± 0.067	0.035 ± 0.030	0.280 ± 0.082	0.022 ± 0.020	0.342 ± 0.076
	NSGA-II 2	**0.180** ± 0.125	**0.014** ± 0.009	0.090 ± 0.050	0.169 ± 0.057	0.049 ± 0.025	0.194 ± 0.061
	NSGA-II 3	0.169 ± 0.124	0.025 ± 0.013	**0.186** ± 0.138	**0.040** ± 0.013	**0.094** ± 0.069	**0.074** ± 0.014
300 materials	GA	0.026 ± 0.047	0.546 ± 0.091	0.034 ± 0.044	0.568 ± 0.112	0.046 ± 0.055	0.699 ± 0.127
	NSGA-II 2	**0.457** ± 0.148	**0.076** ± 0.063	0.190 ± 0.169	0.371 ± 0.145	0.206 ± 0.187	0.437 ± 0.169
	NSGA-II 3	0.420 ± 0.183	0.104 ± 0.045	**0.447** ± 0.127	**0.080** ± 0.021	**0.347** ± 0.111	**0.129** ± 0.027
1000 materials	GA	0.008 ± 0.019	0.877 ± 0.243	0.009 ± 0.036	0.837 ± 0.143	0.012 ± 0.036	0.912 ± 0.191
	NSGA-II 2	**0.569** ± 0.100	**0.058** ± 0.060	0.154 ± 0.165	0.462 ± 0.158	0.185 ± 0.190	0.518 ± 0.185
	NSGA-II 3	0.505 ± 0.112	0.124 ± 0.047	**0.551** ± 0.097	**0.101** ± 0.039	**0.467** ± 0.114	**0.145** ± 0.048

Table 4
Proportion of the Pareto set. NSGA-II 2 and NSGA-II 3 correspond to NSGA-II, respectively, for 2 and 3 objectives.

		2 objectives (%)	3 objectives (%)	5 objectives (%)
Real 284 materials	GA	0.060 ± 0.046	0.137 ± 0.092	0.663 ± 0.316
	NSGA-II 2	**0.898** ± 0.053	0.591 ± 0.100	0.979 ± 0.032
	NSGA-II 3	0.366 ± 0.229	**0.950** ± 0.065	**1.000** ± 0.000
50 materials	GA	0.101 ± 0.044	0.362 ± 0.126	0.914 ± 0.107
	NSGA-II 2	**0.918** ± 0.069	0.666 ± 0.114	0.988 ± 0.015
	NSGA-II 3	0.781 ± 0.208	**0.997** ± 0.005	**0.999** ± 0.003
300 materials	GA	0.024 ± 0.027	0.088 ± 0.059	0.655 ± 0.337
	NSGA-II 2	**0.853** ± 0.138	0.498 ± 0.131	0.930 ± 0.099
	NSGA-II 3	0.383 ± 0.210	**0.937** ± 0.070	**0.999** ± 0.001
1000 materials	GA	0.001 ± 0.005	0.038 ± 0.049	0.636 ± 0.264
	NSGA-II 2	**0.903** ± 0.114	0.440 ± 0.175	0.881 ± 0.171
	NSGA-II 3	0.250 ± 0.171	**0.879** ± 0.101	**0.999** ± 0.002

6.2.2. *Pareto proportion*

The Pareto proportion for each algorithm was calculated concerning the non-dominated solutions obtained for each student, number of materials, and number of objectives. Table 4 presents the means and standard deviations of the results. To perform this calculation, the non-dominated solutions from the final population of GA were used, taking into account the number of objectives in each scenario.

The results indicate a similar pattern when compared with the metrics. For 2 objectives, the aggregation of NSGA II for 2 objectives displays a higher Pareto frontier proportion. For 3 objectives, the aggregation of NSGA II for 3 objectives shows a superior Pareto frontier proportion. Also, similarly to the results of the multi-objective metrics, NSGA II for 3 objectives achieves slightly better results than NSGA II for 2 objectives when all 5 objectives are considered in the comparisons. Notably, all individuals within the Pareto frontier were obtained by the NSGA II for 3 objectives in the real base for 5 objectives, and almost all (99.9%) in the other instances of materials for 5 objectives. Both NSGA II aggregations outperform GA in all scenarios.

6.2.3. *Comparison of the number of dominated individuals*

After observing the ability of the NSGA-II to obtain good solutions even when the performance is evaluated considering the sum of the 5 objectives (mono-objective), this experiment analyzes the solutions found according to their dominance. For this, 30 executions of each algorithm were performed, for a total of 24 students, for each material dataset. The NSGA-II used a population of 50 individuals, totaling a max of 36 000 individuals per instance.

The non-dominated solutions obtained for each independent run in each instance were used in this analysis in Tables 5–8. One can observe that the number of individuals for each instance is smaller than 36 000 as only the non-dominated solutions were considered in this analysis.

The dominance relations for the NSGA-II with two objectives and for the NSGA-II with three objectives are presented in Tables 5 and 6, respectively. As anticipated, NSGA-II typically reached a higher proportion of solutions that dominate the solutions obtained by GA than the solutions of GA that dominate those found by NSGA-II. However, it should be noted that the results obtained by NSGA-II for the instance with 50 materials (18 non-dominated solutions for NSGA-II with two objectives and 22 for NSGA-II with three objectives) indicate that the grouping suggested by the aggregation tree did not imply a significant gain in the ratio of

Table 5
Comparison regarding dominance between GA and NSGA-II with 2 objectives.

Instance	GA	Neither	NSGA-II	Total
Real 284 materials	1 (0.00%)	25,372 (97.40%)	675 (2.59%)	26,048
50 materials	6 (0.03%)	19,745 (99.88%)	18 (0.09%)	19,769
300 materials	1 (0.00%)	18,193 (89.98%)	2025 (10.02%)	20,219
1000 materials	0 (0.00%)	2705 (17.25%)	12,974 (82.75%)	15,679

Table 6
Comparison regarding dominance between GA and NSGA-II with 3 objectives.

Instance	GA	Neither	NSGA-II	Total
Real 284 materials	2 (0.01%)	35,610 (99.08%)	329 (0.92%)	35,941
50 materials	59 (0.17%)	34,092 (99.76%)	22 (0.06%)	34,173
300 materials	0 (0.00%)	34,640 (96.37%)	1303 (3.63%)	35,943
1000 materials	0 (0.00%)	14,055 (39.09%)	21,901 (60.91%)	35,956

Table 7
Dominance considering the solutions of NSGA-II (version with 2 objectives) and the sum of the 5 objectives.

Instance	GA	Neither	NSGA-II	Total
Real 284 materials	0 (0.00%)	26,051 (99.78%)	57 (0.22%)	26,108
50 materials	5 (0.03%)	19,739 (99.81%)	33 (0.17%)	19,777
300 materials	1 (0.00%)	20,147 (98.76%)	251 (1.23%)	20,399
1000 materials	0 (0.00%)	17,915 (91.99%)	1561 (8.01%)	19,476

Table 8
Dominance considering the solutions of NSGA-II (version with 3 objectives) and the sum of the 5 objectives.

Instance	GA	Neither	NSGA-II	Total
Real 284 materials	1 (0.00%)	35,861 (99.78%)	79 (0.22%)	35,941
50 materials	46 (0.11%)	34,104 (99.80%)	24 (0.07%)	34,174
300 materials	0 (0.00%)	35,696 (99.31%)	247 (0.69%)	35,943
1000 materials	0 (0.00%)	35,480 (98.66%)	481 (1.34%)	35,961

Table 9
Results for dominance comparing the solutions of NSGA-II for 2 objectives with those found by GA considering its final populations. Here, the dominance regards the 2 objectives of NSGA-II.

Instance	GA	Neither	NSGA-II	Total
Real 284 materials	1 (0.00%)	27,657 (63.31%)	16,030 (36.69%)	43,688
50 materials	7 (0.02%)	20,524 (65.57%)	10,770 (34.41%)	31,301
300 materials	1 (0.00%)	17,354 (47.70%)	19,028 (52.30%)	36,383
1000 materials	0 (0.00%)	4207 (13.10%)	27,908 (86.90%)	32,115

dominance over the solutions provided by GA. This may be related to the low amount of materials in these instances. On the other hand, the advantage of using NSGA-II with the aggregation trees increases with the number of materials. In the cases with 1000 materials, NSGA-II reached 82.75% and 60.91% when compared with 2 and 3 objectives, respectively, and 8.01% and 1.34% when all the 5 objectives are considered in the analysis.

In addition, we compared the dominance ratio while taking the original five objectives into account. In other words, the optimization process was carried out using the NSGA-II algorithm by aiming at the aggregate form of the initial problem with 2 and 3 objectives. In this case, the 5 objectives were then used to identify the solutions that dominate those found by the other technique in the comparison. Table 7 displays the dominance relations for the NSGA-II with two objectives, while Table 8 displays the dominance relations for the NSGA-II with three objectives. The results are consistent with the prior experiment as NSGA-II exhibits a higher proportion of solutions that dominate the GA solutions in instances involving the most materials. However, it should be observed that in this experiment, these percentages are substantially smaller, proving that the aggregation suggested by the Aggregation Tree algorithm successfully detects those related objectives.

When the non-dominated solutions of the final population of GA are considered, the proportion of NSGA II solutions that dominate the GA solutions is higher than in the previous case. These results are presented in Tables 9, 10, 11, and 12. This result may indicate that the non-dominated solutions of the final population of GA have similar objective function values. In this scenario, the maximum number of solutions is 360,000, which occurs if the final population of GA and NSGA II do not contain any duplicated or dominated solution.

6.2.4. Non-dominated solutions

To assist in the analysis of the results, plots were generated from the Pareto Front for each student and the instance of available materials. As a lot of graphics were generated, totaling 288, we chose only two of them, both of one student for the instance of 300

Table 10
Results for dominance comparing the solutions of NSGA-II for 3 objectives with those found by GA considering its final populations. Here, the dominance regards the 3 objectives of NSGA-II.

Instance	GA	Neither	NSGA-II	Total
Real 284 materials	1 (0.00%)	80,793 (73.46%)	29,181 (26.53%)	109,975
50 materials	40 (0.05%)	67,477 (80.09%)	16,732 (19.86%)	84,249
300 materials	0 (0.00%)	70,922 (66.44%)	35,827 (33.56%)	106,749
1000 materials	0 (0.00%)	33,673 (24.00%)	106,652 (76.00%)	140,325

Table 11
Results for dominance comparing the solutions of NSGA-II for 2 objectives with those found by GA considering its final populations. Here, the dominance regards all 5 objectives.

Instance	GA	Neither	NSGA-II	Total
Real 284 materials	0 (0.00%)	118,810 (97.17%)	3462 (2.83%)	122,272
50 materials	8 (0.01%)	93,794 (96.65%)	3242 (3.34%)	97,044
300 materials	1 (0.00%)	87,526 (94.19%)	5401 (5.81%)	92,928
1000 materials	0 (0.00%)	88,750 (78.40%)	24,454 (21.60%)	113,204

Table 12
Results for dominance comparing the solutions of NSGA-II for 3 objectives with those found by GA considering its final populations. Here, the dominance regards all 5 objectives.

Instance	GA	Neither	NSGA-II	Total
Real 284 materials	1 (0.00%)	160,627 (95.43%)	7695 (4.57%)	168,323
50 materials	73 (0.04%)	162,531 (96.92%)	5085 (3.03%)	167,689
300 materials	0 (0.00%)	154,846 (94.57%)	8894 (5.43%)	163,740
1000 materials	0 (0.00%)	181,568 (86.87%)	27,455 (13.13%)	209,023

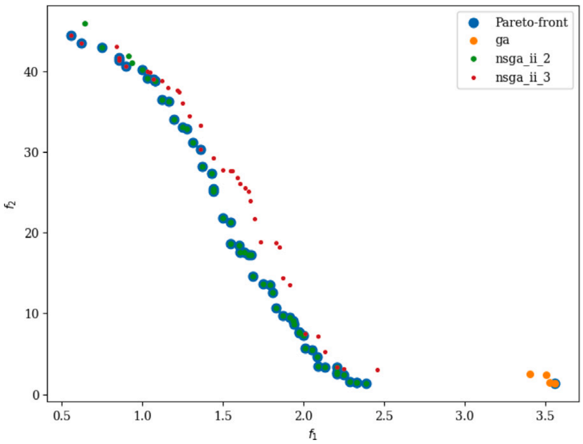

Fig. 4. Pareto fronts concerning 2 objectives for a student of the 300 materials dataset.

materials, one considering 2 objectives (Fig. 4) and the other considering 3 objectives (Fig. 5). These graphics are representative of the behavior that is also present in the other students in other instances and the number of objectives studied. In both graphics, there are the results of the algorithm NSGA-II for the aggregation of 2, represented in green, and of 3 objectives, represented in red, as well as the GA in orange and the Pareto Front calculated from these algorithms in blue.

It is noted that for the graphic considering 2 objectives (Fig. 4), the bigger part of the Pareto Front is from the results of the NSGA-II for 2 objectives, even if in some instances the GA and NSGA-II for 3 objectives also compose the Pareto Front. Meanwhile, in the graphic considering 3 objectives (Fig. 5), the biggest part of the Pareto Front is composed of the results of the NSGA-II for 3 objectives.

In addition, we also present the plots considering the non-dominated solutions of the final population of GA. These plots are in Figs. 6 and 7 and they reinforce the affirmative presented in Section 6.2.3: the non-dominated solutions of the final population of GA are similar. One also can observe that the number of non-dominated solutions of GA increases with the number of objectives (the number of solutions in Fig. 7 is larger than that in Fig. 6).

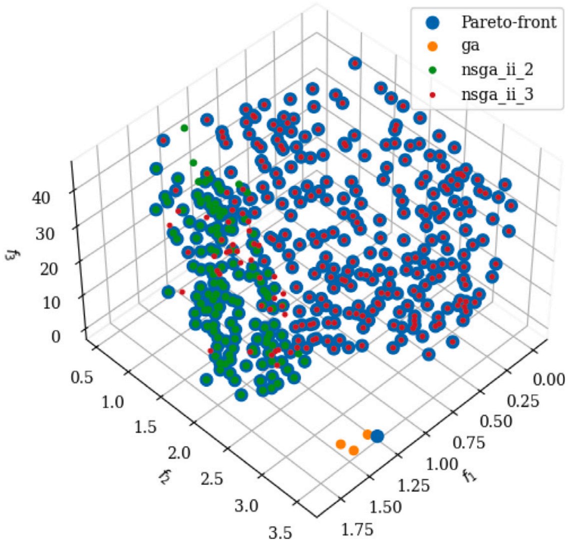

Fig. 5. Pareto fronts concerning 3 objectives for a student of the 300 materials dataset.

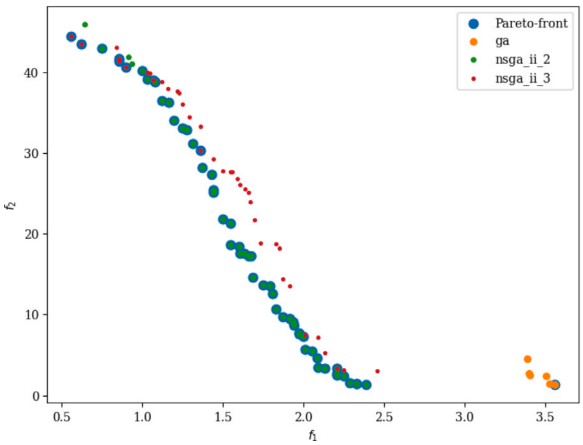

Fig. 6. Pareto fronts concerning 3 objectives for a student of the 300 materials dataset, using the last non-dominated population of GA.

7. Conclusion

Adaptive Curriculum Sequencing (ACS) is important in education due to its personalized learning capacity. Student preference can be obtained from different data sources, such as monitoring student reactions to materials, stress level or emotion related to features of the type of media, and other IoT devices.

In this paper, we solve the multi-objective ACS problem considering five objective functions, which characterizes a many-objective optimization problem. To lead this problem we employed NSGA-II, a well-known Genetic Algorithm for multi-objective optimization problems. Due to NSGA-II has problems when solving many-objective optimization problems, the aggregation trees were employed to reduce the number of objectives to two and three.

For a higher number of materials, the proposal of combining objectives in NSGA-II using aggregation trees has better results in general. When the number of materials is small, GA and NSGA-II reached similar results.

In addition, when the number of objectives used in NSGA-II is evaluated, the aggregation for 3 objectives in NSGA-II obtained better results than aggregating for 2 objectives. In particular, the IGD and Hypervolume values are better when using 3 objectives considering the performance metrics with 3 (as expected) and 5 objectives.

The proposal provide a set of non-dominated solutions. In the sequence, a multi-criteria decision-making process is required to select the most appropriate solution. In this regard, a smaller number of objectives may help in this task, mainly visually. A tool for visualization may provide interactive access to the solutions, mainly when 3 objective functions are used. Also, the decision maker can be the student, the professor, a manager, and so on, and this choice depends on the application. Thus, future works may investigate this decision-making process.

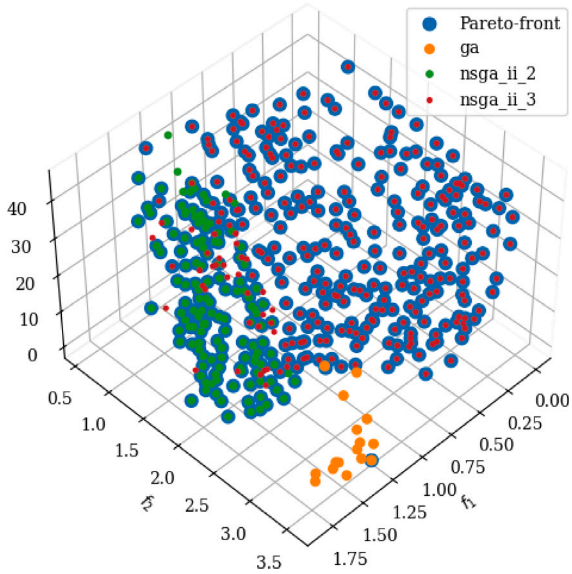

Fig. 7. Pareto fronts concerning 3 objectives for a student of the 300 materials dataset, using the last non-dominated population of GA.

NSGA-II was used to evaluate the proposal, but additional techniques can be adopted. Thus, the analysis of other metaheuristics for multi- and many-objective optimization when solving adaptive curriculum sequence problems is an interesting research avenue.

The multi-objective optimization of the curricula is strongly related to the personalization pursued by some IoT for education researchers. Here, we used real-world data in the computational experiments and both, the dataset and proposal, can be used by other researchers. However, finding data to support this type of research is a hard task. Thus, we observe that making better datasets public is an important challenge for researchers in IoT for education.

CRediT authorship contribution statement

João Vítor de Castro Martins Ferreira Nogueira: Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Heder Soares Bernardino:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing,. **Jairo Francisco de Souza:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Luciana Brugiolo Gonçalves:** Formal analysis, Investigation, Methodology, Project administration, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Stênio Sã Rosário Furtado Soares:** Formal analysis, Investigation, Methodology, Project administration, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

We thank André Ferreira Martins for the programming support, which helped us to perform the computational experiments and the analysis of their results. We also thank the financial support provided by FAPEMIG, Brazil, Capes, Brazil, CNPq, Brazil, and UFJF, Brazil.

References

[1] P. Phobun, J. Vicheanpanya, Adaptive intelligent tutoring systems for e-learning systems, Procedia-Soc. Behav. Sci. 2 (2) (2010) 4064–4069.
[2] D. Davis, G. Chen, C. Hauff, G.-J. Houben, Gauging MOOC learners' adherence to the designed learning path, in: EDM '16: 9th International Conference on Educational Data Mining, 2016, pp. 54–61.
[3] A. Debbah, Y.M.B. Ali, Solving the curriculum sequencing problem with dna computing approach, Int. J. Distance Educ. Technol. (IJDET) 12 (4) (2014) 1–18.
[4] A.S. Rathore, S.K. Arjaria, Intelligent tutoring system, in: Utilizing Educational Data Mining Techniques for Improved Learning: Emerging Research and Opportunities, IGI global, 2020, pp. 121–144.
[5] R.C. Silva, A.I. Direne, D. Marczal, A.C. Borille, P.R.B. Guimarães, A. da Silva Cabral, B.F. Camargo, Adaptability of learning objects using calibration and adaptive sequencing of exercises, Braz. J. Comput. Educ. 26 (01) (2018) 70.
[6] M. Niknam, P. Thulasiraman, LPR: A bio-inspired intelligent learning path recommendation system based on meaningful learning theory, Educ. Inf. Technol. (2020) 1–23.
[7] M.d.O.C. Machado, E. Barrére, J. Souza, Solving the adaptive curriculum sequencing problem with prey-predator algorithm, Int. J. Distance Educ. Technol. (IJDET) 17 (4) (2019) 71–93.
[8] A. Muhammad, Q. Zhou, G. Beydoun, D. Xu, J. Shen, Learning path adaptation in online learning systems, in: Computer Supported Cooperative Work in Design (CSCWD), 2016 IEEE 20th International Conference on, IEEE, 2016, pp. 421–426.
[9] H. Xie, D. Zou, F.L. Wang, T.-L. Wong, Y. Rao, S.H. Wang, Discover learning path for group users: A profile-based approach, Neurocomputing 254 (2017) 59–70.
[10] M.d.O.C. Machado, N.F.S. Bravo, A.F. Martins, H.S. Bernardino, E. Barrere, J.F. de Souza, Metaheuristic-based adaptive curriculum sequencing approaches: a systematic review and mapping of the literature, Artif. Intell. Rev. (2020) 1–44.
[11] X. Sheng, K. Lan, X. Jiang, J. Yang, Adaptive curriculum sequencing and education management system via Group-Theoretic Particle Swarm Optimization, Systems 11 (1) (2023) 34.
[12] A. Badshah, A. Ghani, A. Daud, A. Jalal, M. Bilal, J. Crowcroft, Towards smart education through the internet of things: A review, 2023, arXiv preprint arXiv:2304.12851.
[13] Y. Zhang, Y. Ning, B. Li, Y. Liu, An innovative classroom teaching technology assisted by artificial intelligence of things, in: 2021 2nd International Conference on Information Science and Education (ICISE-IE), IEEE, 2021, pp. 1661–1664.
[14] M. Kassab, J. DeFranco, P. Laplante, A systematic literature review on internet of things in education: Benefits and challenges, J. Comput. Assist. Learn. 36 (2) (2020) 115–127.
[15] M. Al-Emran, S.I. Malik, M.N. Al-Kabi, A survey of Internet of Things (IoT) in education: Opportunities and challenges, in: Toward Social Internet of Things (SIoT): Enabling Technologies, Architectures and Applications: Emerging Technologies for Connected and Smart Social Objects, Springer, 2020, pp. 197–209.
[16] B. Cun-Ling, W. De-Liang, L. Shi-Yu, L. Wei-Gang, D. Jun-Yu, Adaptive learning path recommendation based on graph theory and an improved immune algorithm, KSII Trans. Internet Inf. Syst. 13 (5) (2019).
[17] B. Hssina, M. Erritali, A personalized pedagogical objectives based on a genetic algorithm in an adaptive learning system, Procedia Comput. Sci. 151 (2019) 1152–1157.
[18] B.C.L. Christudas, E. Kirubakaran, P.R.J. Thangaiah, An evolutionary approach for personalization of content delivery in e-learning systems based on learner behavior forcing compatibility of learning materials, Telemat. Inform. 35 (3) (2018) 520–533.
[19] D.J. Almeida, M.A. Fernandes, N.T. da Costa, Sequencing and recommending pedagogical activities from bloom's taxonomy using RASI and multi-objective PSO, in: Proceedings of the 14th International Conference on Computer Supported Education – CSEDU, Vol. 2, 2022, pp. 105–116.
[20] N.T. Son, J. Jaafar, I.A. Aziz, B.N. Anh, Meta-heuristic algorithms for learning path recommender at MOOC, IEEE Access 9 (2021) 59093–59107.
[21] V. Vanitha, P. Krishnan, R. Elakkiya, Collaborative optimization algorithm for learning path construction in E-learning, Comput. Electr. Eng. 77 (2019) 325–338.
[22] A. Essayah, M. Youssfi, O. Bouattane, E. Illoussamen, QoS-based semantic micro services discovery and composition using ACO Algorithm, Int. J. Adv. Comput. Sci. Appl. (IJACSA) 10 (6) (2019).
[23] K. Seki, T. Matsui, T. Okamoto, An adaptive sequencing method of the learning objects for the e-learning environment, Electron. Commun. Japan (Part III: Fundam. Electron. Sci.) 88 (3) (2005) 54–71.
[24] Y. Gao, L. Peng, F. Li, W. Li, et al., A multi-objective PSO with Pareto archive for personalized E-Course composition in moodle learning system, in: 2015 8th International Symposium on Computational Intelligence and Design (ISCID), Vol. 2, IEEE, 2015, pp. 21–24.
[25] L. de Marcos, J.-J. Martínez, J.-A. Gutiérrez, Swarm intelligence in e-learning: a learning object sequencing agent based on competencies, in: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, 2008, pp. 17–24.
[26] L. de Marcos, R. Barchino, J. Martínez, J. Gutiérrez, A new method for domain independent curriculum sequencing: a case study in a web engineering master program, Int. J. Eng. Educ. 25 (4) (2009) 632.
[27] C.-P. Chu, Y.-C. Chang, C.-C. Tsai, PC 2 PSO: personalized e-course composition based on Particle Swarm Optimization, Appl. Intell. 34 (2011) 141–154.
[28] N.S. Raj, V. Renumol, A systematic literature review on adaptive content recommenders in personalized learning environments from 2015 to 2020, J. Comput. Educ. 9 (1) (2022) 113–148.
[29] E. Popescu, P. Trigano, C. Badica, B. Butoi, M. Duica, A course authoring tool for WELSA adaptive educational system, in: Proc. ICCC 2008, 2008, pp. 531–534.
[30] R.M. Felder, L.K. Silverman, et al., Learning and teaching styles in engineering education, Eng. Educ. 78 (7) (1988) 674–681.
[31] D.A. Kolb, R.E. Boyatzis, C. Mainemelis, Experiential learning theory: Previous research and new directions, in: Perspectives on Thinking, Learning, and Cognitive Styles, Routledge, 2014, pp. 227–248.
[32] C.-C. Hsu, C.-C. Ho, The design and implementation of a competency-based intelligent mobile learning system, Expert Syst. Appl. 39 (9) (2012) 8030–8043.
[33] S. Wan, Z. Niu, A learner oriented learning recommendation approach based on mixed concept mapping and immune algorithm, Knowl.-Based Syst. 103 (2016) 28–40.
[34] C.-M. Chen, Intelligent web-based learning system with personalized learning path guidance, Comput. Educ. 51 (2) (2008) 787–814.
[35] M.-J. Huang, H.-S. Huang, M.-Y. Chen, Constructing a personalized e-learning system based on genetic algorithm and case-based reasoning approach, Expert Syst. Appl. 33 (3) (2007) 551–564.
[36] Q. Guo, M. Zhang, Implement web learning environment based on data mining, Knowl.-Based Syst. 22 (6) (2009) 439–442.
[37] A.F. Martins, M. Machado, H.S. Bernardino, J.F. de Souza, A comparative analysis of metaheuristics applied to adaptive curriculum sequencing, Soft Comput. (2021) 1–16.
[38] S. Gutiérrez, B. Pardo, Sequencing in web-based education: approaches, standards and future trends, in: Evolution of Teaching and Learning Paradigms in Intelligent Environment, Springer, 2007, pp. 83–117.
[39] P.J. Fleming, R.C. Purshouse, R.J. Lygoe, Many-objective optimization: An engineering design perspective, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer, 2005, pp. 14–32.

[40] A.R. de Freitas, P.J. Fleming, F.G. Guimaraes, Aggregation trees for visualization and dimension reduction in many-objective optimization, Inform. Sci. 298 (2015) 288–314.

[41] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.

[42] J.H. Holland, Genetic algorithms, Sci. Am. 267 (1) (1992) 66–72.

[43] M.R. Sierra, C.C. Coello, A New Multi-Objective Particle Swarm Optimizer with Improved Selection and Diversity Mechanisms, Technical Report of CINVESTAV-IPN, 2004.

[44] C.A. Coello Coello, M. Reyes Sierra, A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm, in: MICAI 2004: Advances in Artificial Intelligence: Third Mexican International Conference on Artificial Intelligence, Mexico City, Mexico, April 26-30, 2004. Proceedings 3, Springer, 2004, pp. 688–697.

[45] E. Zitzler, S. Künzli, Indicator-based selection in multiobjective search, in: International Conference on Parallel Problem Solving from Nature, Springer, 2004, pp. 832–842.