Summer Internship Project
Report

Multi Objective Optimization for Adaptive Recommender System Applied to Corporate MOOC

Submitted by

Selim Lakhdar
University of Lille

Under the guidance of

Laetitia Jourdan
Univerrsity of Lille
Mounir Hafsa
Mandarine Academy
Pamela Wattebled
Mandarine Academy

Abstract

This paper presents my summer internship at Mandarine Academy in the context of my Master's degree at University of Lille.

The first section includes a bibliographic search for recommender systems. Followed by a brief introduction to multi-objective optimization and its usage with recommender systems. In addition, the concept of Adaptive Learning is studied and common techniques are exposed. In the second part, we presented some graph representation and a trivial and initial multi-objective recommender system. Finally, we proposed an Adaptive Learning Recommender System model to respond to Mandarine needs.

Contents

Chapter 1

Introduction

Back to 2008, George Siemens and Stephen Downes at the University of Manitoba in Canada offered an online course (at a distance) called Connectivism and Connective Knowledge (CCK08) which was attended by 2.200 participants. They later referred to it as a MOOC (Massive Open Online Course), and a new era of internet-based learning began [1].

With the development of information technology, many e-learning platforms have emerged with rich online resources like videos, quizzes, presentations, online tests ... Quickly, those platforms face the problem of recommending the appropriate content for their users. Indeed, the variety of content is exponential and the user may get lost, which leads to an increased dropout rate.

For this purpose recommender systems are used to guide the user through the content. This field of machine learning is evolving quickly. Nowadays, a new generation of recommender systems for MOOCs is emerging. They include more flexibility and adaptability for recommending a learning path. They are able to adapt to user needs and goals.

The learning path search problem involves the courses with their dependencies, the learner preferences, and objectives. To be able to efficiently find the right content in this large search space, optimization is widely used.

This paper is presented as follows: Chapter 2 introduces the context of internship and explains the problem that will be treated. Chapter 3 introduces common notions about recommender systems and multi-objective optimization, with their application to MOOC, from the literature. Finally, it will showoff the proposed recommender system model, which is adaptive to user preferences.

Chapter 2

Context

2.1 Mandarine Academy

Mandarine Academy is a french Ed-Tech company founded in 2008 that specializes in innovative corporate training techniques such as personalized MOOCs, training logistics, web conferences, etc. Its first goal is to help companies with their digital transformation by facilitating the use of new technologies.

Mandarine accompanied more than 3000 clients and has more than 500K users all over the world. Its main product is the Office 365 MOOC which is the most accessed with more than 3.5K users per month.

They propose a large panel of tutorials, use cases, quizzes on Microsoft Office products like Word, Excel, Teams, organized into learning paths.

2.2 Problem Statement

As stated before, recommender systems are used to guide users by providing them with the right content to consume, which mitigates the dropout rate. Nevertheless, those systems suffer from a well-known problem, which is *One Size Fits All Problem* which is expressed by recommending content in the same way to users with different profiles. They mainly ignore the users' progress and changes during the learning process. Furthermore, they ignore the users' knowledge background and their ability to learn.

To face this problem, adaptive recommender systems are used to capture users' interests and preferences. Furthermore, multi-objective optimization is used to efficiently explore the large search space and provide the adapted content (with user-specific constraints), but they remain complex techniques, which are not easy to setup.

2.3 Objectives

In this context, my work is to do bibliographic research about recommender systems for e-learning content (ie: MOOC) in particular adaptive recommender systems. Furthermore, the use of multi-objective optimization have to be considered to efficiently explore the search space. Finally, we will propose a multi-objective adaptive recommender system model based on the advances in this field and test it with generic data.

Chapter 3

Work Done

3.1 Bibliographic Search

My first step was to study the literature to understand what is a recommender system and how it works. Furthermore, I had to focus on applying the recommendation on MOOCs (ie: learning content) for adaptive learning and explore the optimization techniques used for it.

3.1.1 Recommender Systems

Recommender systems are an ensemble of techniques that rely on data filtering and analysis to predict users' interest [2]. It's used in many ways like product recommendation (for online shopping: ie Amazon) or movie (video) recommendation in online platforms such as Youtube or Netflix. It aims to provide a personalized user experience to increase product/content consumption.

It relies on users feedback which can be categorized to two types [3];

- Explicit Feedback: which reflects user interest like asking user to rate a specific product.
- Implicit Feedback: which do not reflects user interest but give insights about users' preferences by monitoring the performed actions such as visits or total time spent.

In general Recommender Systems can be grouped in 3 categories as showed in Figure 3.1.

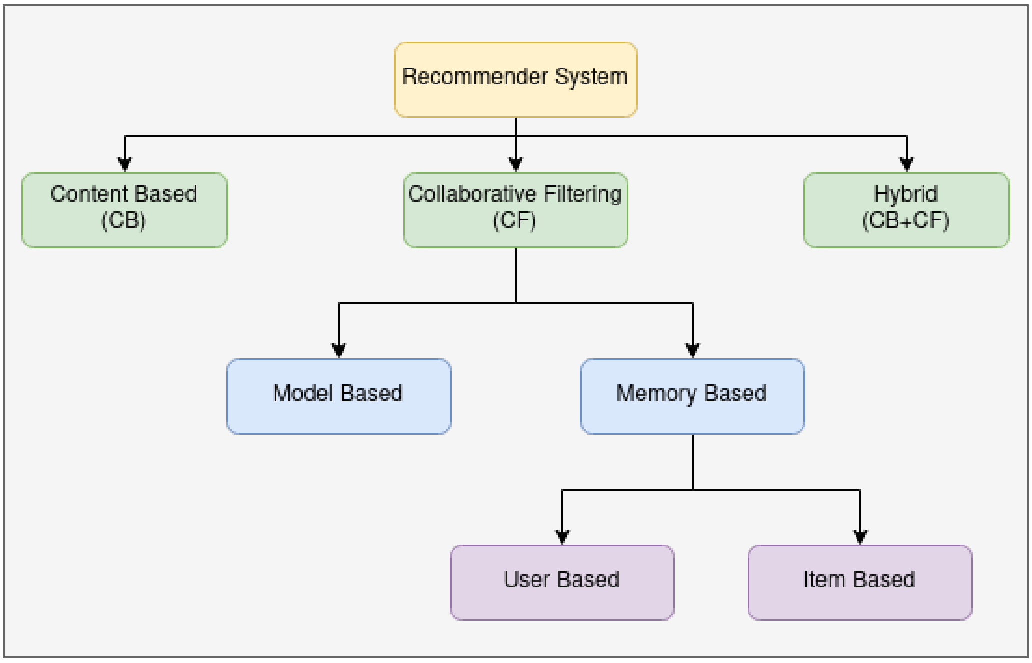


Figure 3.1: Recommender System Overview

3.1.1.1 Content Based

Content based recommender systems rely on content description and a profile of user preferences. These algorithms try to recommend items similar to those that a user liked in the past or is examining in the present. This approach assumes that if a user likes an item, they will also like items with similar characteristics. Lops et al presentend [4] a high-level architecture for those systems which is represented in Figure 3.2.

The main advantage of this technique is that it don't relies on users reviews. Indeed, similar characteristics between old and a new content can be used to rate it without users implication. In addition, it can capture niche interest based on specific user taste.

However, crafting the suitable features to describe the content may be a difficult task. Furthermore, those techniques don't assure the diversity or serendipity (ie: unplanned fortunate discovery).

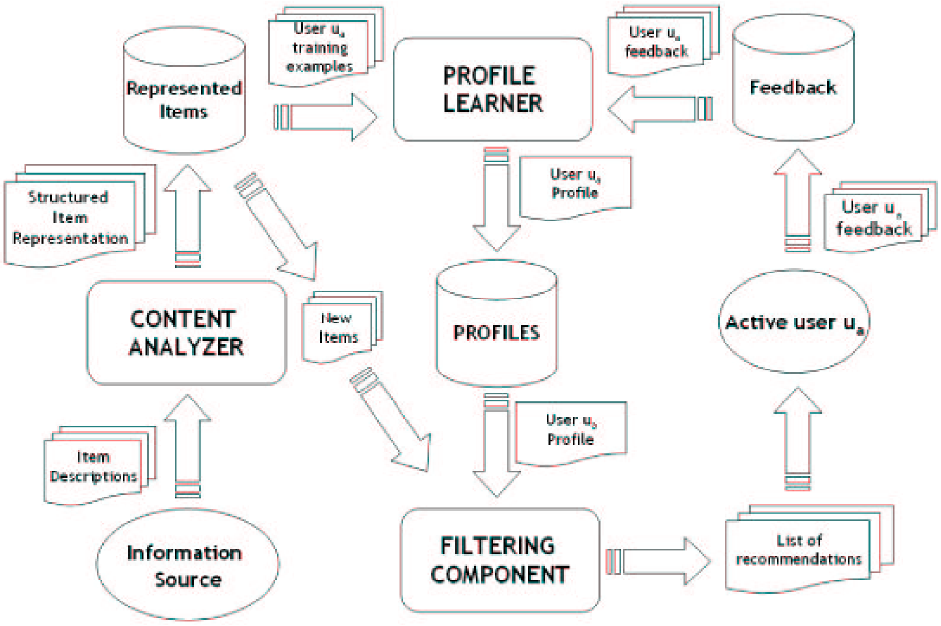


Figure 3.2: Content Based Recommender System High Level Architecture

3.1.1.2 Collaborative Filtering

Collaborative Filtering is a technique based on users rating. It not depends on the product themselves. This category is divided in two main categories.

3.1.1.2.1 Memory-based or neighborhood based methods are the most basic because they use no model at all. They assume that predictions can be made based solely on "memory" of past data and typically use a simple distance-measurement approach, such as the nearest neighbor.

It contains two methods:

- User-Based: Similar users which have similar ratings for similar items are found and then target user's rating for the item which target user has never interacted is predicted.
- Item-based: Find similar items to items which target user already rated or interacted.

Memory-based methods are simple and easy to explain and interpret, as well as easy to implement and apply. It doesn't require to craft suitable features to describe the item. However, this method suffer from sparsity (Arises from the phenomenon that users in general rate only a limited number of items) in ratings table and cold start (Inability to recommend for new users and items).

Memory-based recommendation systems are not always as fast and scalable as we would like them to be, especially in the context of actual systems that generate real-time recommendations on the basis of very large datasets. To achieve these goals, model-based recommendation systems are used.

3.1.1.2.2 Model Based approach relies on Machine Learning Models. It presuppose some form of the underlying model and attempt to ensure that any predictions made fit the model properly. One common approach for model-based CF is Matrix factorization (ex: Singular Value Decomposition-SVD) that was used in the Netflix Challenge.

Model-Based approach have many advantages like;

- Scalability: Most models resulting from model-based algorithms are much smaller than the actual dataset, so that even for very large datasets, the model ends up being small enough to be used efficiently. This imparts scalability to the overall system.
- Prediction speed: Model-based systems are also likely to be faster, at least in comparison to memory-based systems because, the time required to query the model (as opposed to the whole dataset) is usually much smaller than that required to query the whole dataset.
- Avoidance of overfitting: If the dataset over which we build our model is representative enough of real-world data, it is easier to try to avoid overfitting with model-based systems.

In the other hand, it faces some drawbacks like;

- Inflexibility: Because building a model is often a time and resource-consuming process, it is usually more difficult to add data to model-based systems, making them inflexible.
- Quality of predictions: The fact that we are not using all the information, it is possible that with model-based systems, we don't get predictions as accurate as with memory-based systems.

3.1.1.3 Hybrid

Content-Based and Collaborative methods bring their own strengths and weaknesses, combining multiple methods together can achieve much better results while benefiting from all the advantages each method presents. Those combined methods are called Hybrid methods.

They are categorized into 3 general categories [5];

3.1.1.3.1 Monolithic: A single algorithm that integrates multiple approaches by pre-processing and combining several knowledge sources (Figure 3.3).

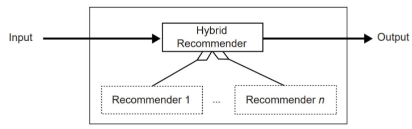


Figure 3.3: Monolithic Model [5]

3.1.1.3.2 Parallel: This method operates independently, each approach produces its own lists of recommendations which are later combined into a final set of solutions (Figure 3.4).

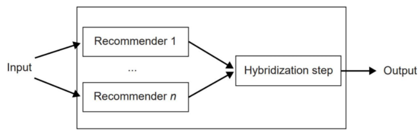


Figure 3.4: Parallel Model [5]

3.1.1.3.3 Pipeline: In this approach the output of one of engine becomes part of the input data of the next engine (Figure 3.5).

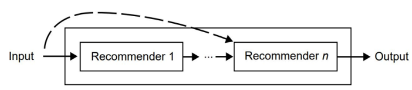


Figure 3.5: Pipeline Model [5]

3.1.2 Multi Objectives Optimization

Multi-Objectives Optimization is a branch of mathematics used in multiple criteria decision-making, which deals with optimization problems involving two or more objective function to be optimized simultaneously [12]. Maximizing learned knowledge and minimizing the total learning time is an example of multi-objective optimization problems with two objectives.

3.1.2.1 Pareto Front

Unlike single-objective optimization, results are presented as a Pareto set of optimal solutions where no improvement can be made for a particular objective without sacrificing another one [13]. Figure 3.6 shows an example of a Pareto front for a two objective optimization.

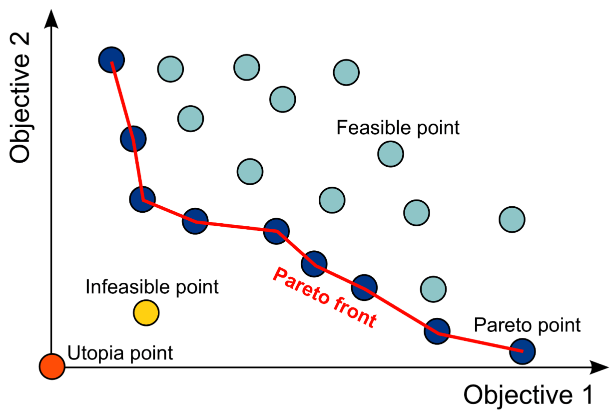


Figure 3.6: Example of a Pareto front in a minimization problem. [14]

3.1.2.2 Metaheuristics

Metaheuristics are strategies that "guide" the search process to find a near optimal solution of a large scale problem. The goal is to efficiently explore the search space and avoid being trapped in confined areas in order to give a solution in a feasible time [15].

Historically many variant has been proposed based on different inspiration. Some are made by analogy to other scientific domains as physics (simulated annealing), biology (ant colony and evolutionary algorithms), sociology (particle swarm optimization). In a general way two contradictory

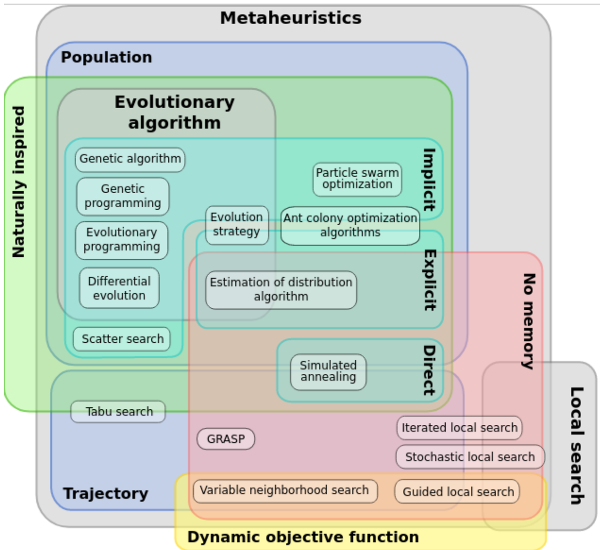


Figure 3.7: Euler diagram of the different classifications of metaheuristics

criteria must be taken into account: exploration (search in new areas) and exploitation (intensify the search in a specific area). Figure 3.7 shows an overview of metaheuristics.

3.1.2.2.1 Genetic Algorithms (GAs): are a family of computational models inspired by evolution. Firstly introduced by J. Holland in the 1970s [6] they are based on a simplified biological model and natural selection. These algorithms encode a potential solution to a specific problem on a simple chromosome like data structure and apply recombination operators to these structures so as to preserve critical information. They are commonly used to generate high-quality solutions for optimization problems and search problems [7].

Main operators of GAs are [7];

- Selection Operator: The idea is to give preference to the individuals with good fitness scores and allow them to pass their genes to successive generations. A common technique is the Roulette Wheel Selection.
- Crossover Operator: This represents mating between individuals. After selecting two individual using the Selection Operators the operation is a to exchange a sequence between the two candidate to create a completely new individual (offspring).

- Mutation: The idea is to insert random genes in offspring to preserve diversity int the population and avoid premature convergence.

3.1.2.2.2 Ant Colony Optimization (ACO): Ant Colony Optimization technique are purely inspired from the foraging behaviour of ant colonies. They are based on Ant Pheromone concentration which is used to guide ant colony to a food spot by the shortest way. It was first proposed by Alberto et al [8] and is used for solving combinatorial optimization problems such as the problem of travelling salesman, or vehicles routing and scheduling and many others.

The general idea behind this approach is to randomly disperse ants in the search space. Each ant will then try to seek a food point which is represented as a solution for the problem. For each step the ant leave a quantity of pheromone which will remain for a certain period due to the evaporation mechanism. The ants are guided by those pheromones and subsequently the best path will be more marked by pheromones.

Many implementation can be found in the literature, each implementation have its own hyper-parameter (ie:number of ants, pheromone dispersion rate, evaporation rate ...) to face a particular problem.

3.1.3 Adaptive Learning

Adaptive Learning notion evolved over time. In 2004 Mödritscher et al [9] list four aspects of adaptivity desired in an ideal e-learning system;

- Adaptive content aggregation: Locate and navigate in information hyperspace
- Adaptive presentation: Adapts the contents or display of a page according to the user's profile
- Adaptive navigation
- Adaptive collaboration support: Helps learners to find the most suitable helpers or collaborators)

More recently, in 2017 Deng et al [10] stated that adaptivity refers to approaches that generate learning paths considering the individual differences in learning preferences, goals, abilities, knowledge background, etc.

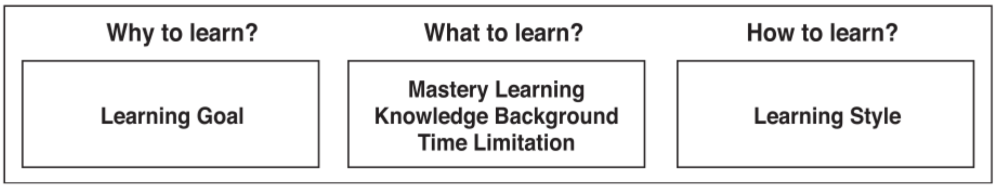


Figure 3.8: Adaptive Learning Parameters [10]

3.1.3.1 User Profile

To be able to adapt to user changes, many examples in the literature explored the ways to represent a user and his evolution.

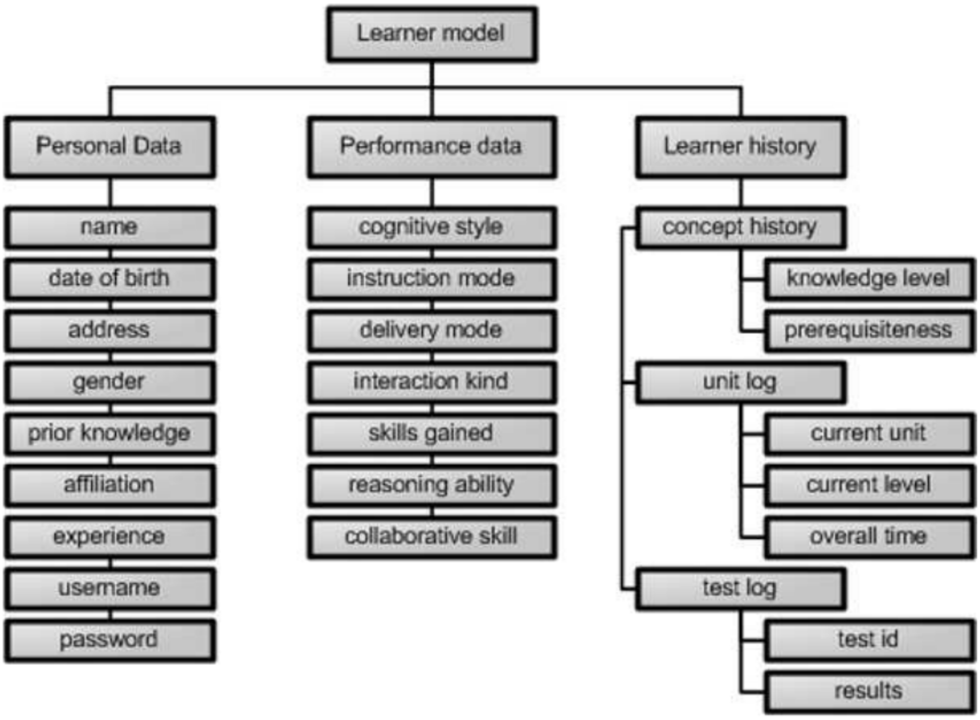


Figure 3.9: Learner Model [25]

Vesin et al [25] presented a 3-layer Learner Model to define and track user (Figure 3.9). It include;

- Objective information: which includes data supplied directly by the learner.

- Learner's performance: which includes data about level of knowledge of the subject domain, misconceptions, progress and the overall performance for a particular learner.
- Learning history: which includes information about lessons and tests learner has already studied, interaction with system, the assessments underwent, etc.

Kritikou et al [26] presented a 3-user-groups based on difficulty and update user profile according to evaluation parameters. Figure 3.10 represent the 3 groups: **Novice**, **Intermediate**, **Expert** with their different target parameters based on;

- Content Difficulty: influenced by the level of the user's knowledge on the subject, being. As the user takes more and more courses, the level of knowledge changes, from novice to intermediate and so on.
- Content Volume: refers to the amount of information the user wishes to explore in the lesson to study.
- Content Interactivity: refers to the level of interactivity wanted by user.
- Lesson Interface: is influenced by the user's learning styles and user preferences on content structure. (see Learning Style)

Setting the right target parameter isn't an easy task. To tackle this problem they set a number of evaluation parameters (Figure 3.10), which aid the system in predicting user's performance while progressing with the study. The evaluation parameters are;

- Lesson Duration: refers to the time that a user spends for completing a lesson. It's compared to pre-estimated threshold values.
- Test Duration: refers to the time that user spends for completing a test.
- Performance: By keeping the user's test scores, the system can estimate how well the user has comprehended the concepts of the course. This parameter is assumed to have four possible values: 'A', 'B', 'C' or 'D'.

Wan et al [27] defined a mixed concept map to to establish the association between learners and resources (Figure 3.11). Concept mapping can adaptively update learners' learning objective goals and adjust the candidate learning resources.

User groups—target parameters

	Novice	Intermediate	Expert
Content Difficulty	Low	Medium	High
Content Volume	Low	Medium	High
Interactivity	High	Medium	Medium/low
Lesson Interface	Simple	Normal/rich	Rich/advanced

User groups—evaluation parameters

	Novice	Intermediate	Expert
Lesson Duration	Very high, high	Medium	Low
Test Duration	Very high, high	Medium	Low
Performance	Low	Medium	Very high, high

Figure 3.10: Learner Model [26]

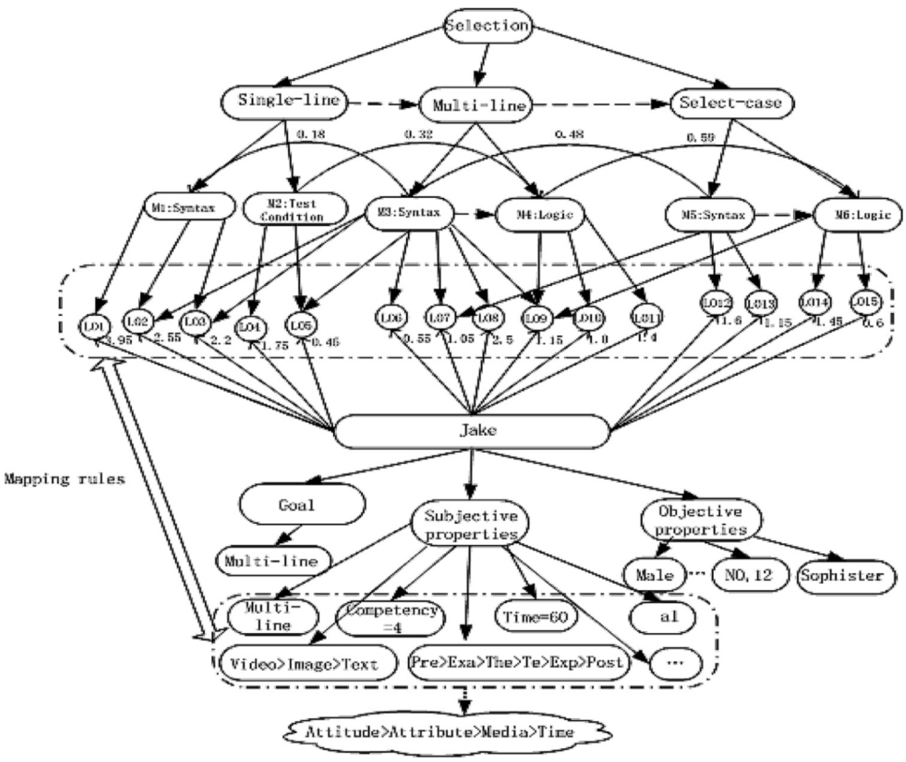


Figure 3.11: Mixed Concept Map [27]

Another approach based on professional social network (Linkedin) scrapping are also present in the literature. Urdaneta-Ponte et al [30] proposed an ontology based recommender system that incorporate endorsement of user Linkedin Skills for Lifelong Learning which allows modelling the data of knowledge areas and job performance sectors to represent professional skills of users obtained from social networks. Dai et al [31] proposed a framework for recommending MOOCs to lifelong learners based on user curricular information on their Linkedin profiles and by considering the job marrket needs. Alruwaili et al [32] presents MR-LI as a course recommendation approach that relies on clustering algorithms to group the users according to their LinkedIn skills and matching them with the right content from Coursera platform (Figure 3.12).

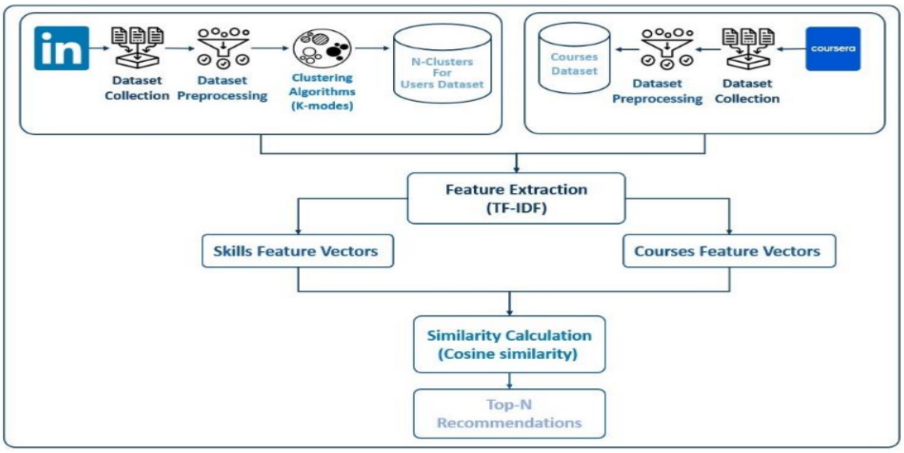


Figure 3.12: MR-LI Framework [32]

3.1.3.2 Techniques

Nabizadeh et al[11] made a state of the art survey on learning path personalization and recommendation methods where they summed up adaptive learning parameters into 3 categories as showed in Figure 3.8. Furthermore, they exposed 3 different personalization methods used in the literature;

3.1.3.2.1 Course Generation (CG) : These methods generate and recommend the entire path to a user in a single recommendation, and the learning assessment occurs only after completing the path. Ignoring the changes that occur during the learning process. (May increase dropout). Figure 3.13

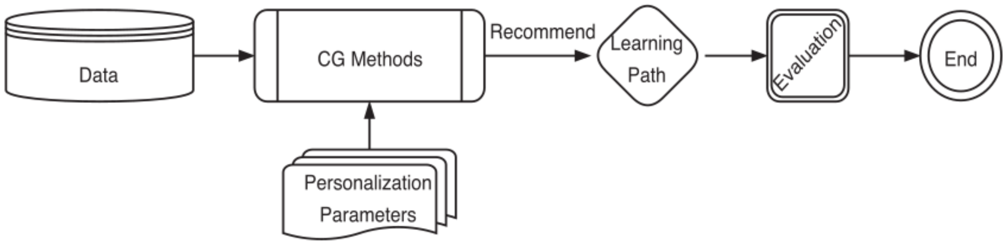


Figure 3.13: Course Generation Methods

3.1.3.2.2 Sequential Pattern Recognition (SPR) : are a subset of CG methods. Sequential pattern mining approaches are mainly applied to discover a learning path for a user from the transactions of similar users. Users are similar if they have similar initial states, preferences, goals, etc. Not frequently used. Figure 3.14

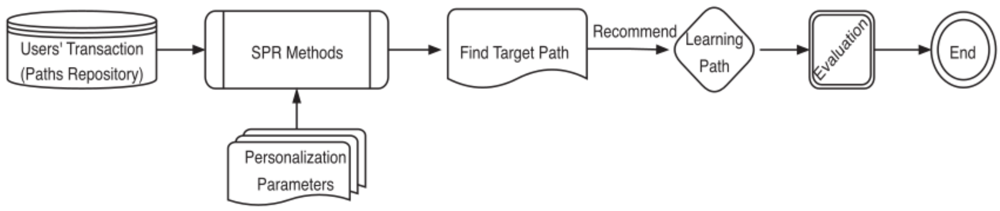


Figure 3.14: Sequential Pattern Recognition Methods

3.1.3.2.3 Course Sequence (CS) : These methods generate and recommend a path to a user step by step considering the user's progress, and the learning assessment occurs as the user proceeds on the path. User profile update is time consuming and might be unnecessary. Feature selection for update is critical. Figure 3.15

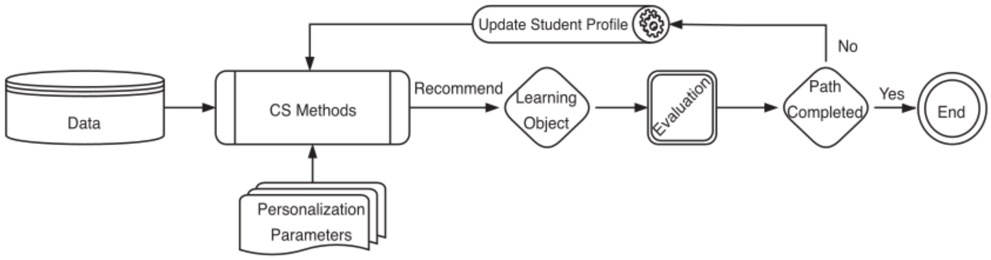


Figure 3.15: Course Sequence Methods

3.1.3.3 Learning Style

The notion of Learning Style is strongly related to the notion of Adaptive Learning. It is described by Cury [16] as *"characteristic cognitive, effective, and psychosocial behaviors that serve as relatively stable indicators of how learners perceive, interact with, and respond to the learning environment"*. It refers to how the learner learn by representing a set of characteristics, mostly related to personality and attitude. they are gathered by questionnaire approaches.

Historically, many approaches was introduced by the most common ones are Kolb's and Felder-Silverman models. The later one is the most used in the literature and can be expressed by a 4-dimensions model (Figure 3.16) described as:

- Perception dimension (Intuitive/Sensing poles): deals with how individuals prefer to perceive information. Sensing learners choose to deal with concrete facts whereas an intuitive learner works based on abstract concepts and ideas.
- Processing dimension (Active/Reflective poles): cites the ability of a learner to be able to process information. While an active learner learns best by practising and working in groups, a reflective learner prefers learning by thinking things through and working individually.
- The Input modality dimension (Visual/Verbal poles): is related to an individual's preference on how information is presented. While visual learners process information better when presented through graphics and presentations, verbal learners perform better by listening to verbal instructions and lectures.

- Understanding dimension (Sequential/Global poles): express a learner's ability to understand information. Generally, a sequential learner inclines towards thinking in an orderly manner, in sequential steps, a global learner prefers learning by integrating various ideas and in large leaps.

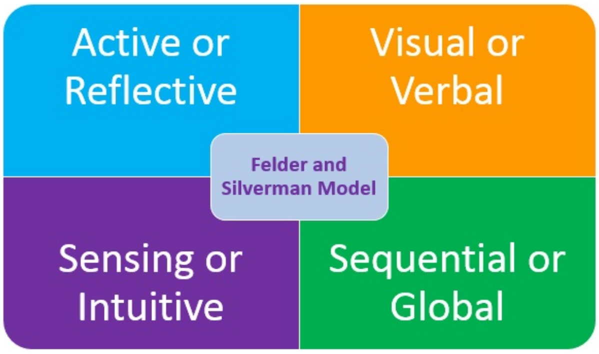


Figure 3.16: Felder-Silverman Learning Style

The Felder-Silverman Model is expressed by a questionnaire with 44 questions grouped in 4 categories, each one containing 11 questions. The final score express the preferences of a user for each category as showed in Figure 3.17.

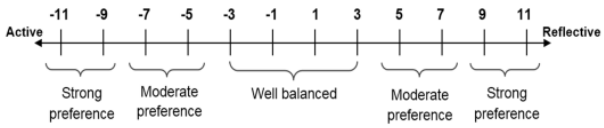


Figure 3.17: Felder-Silverman Learning Style [20]

In their work Nafea et al [18] stated that the Felder-Silverman model is the most appropriate to capture user learning style. Indeed this model integrate the work of others proposed models and cover all categories. Furthermore, the present version of the LS questionnaire is reckoned to be valid, reliable and appropriate to aid in identifying Learning Styles. Several studies conducted in the past related to the reliability and the validity of this questionnaire set forth by Felder and Silverman have emerged out to be quite positive. Furthermore they presented a dynamic Learning Style questionnaire based on the Felder-Silverman questionnaire which tries to reduce the answer time by skipping questions when the majority of answers are grouped in the same category (Figure 3.18).

	Participants							
	Clara		Emma		Bob		Oliver	
D1 (Information Processing)	Active	Reflective	Active	Reflective	Active	Reflective	Active	Reflective
Questions ULEARN Sequence	a	b	a	b	a	b	a	b
Q1	✓		✓			✓	✓	
Q2	✓			✓		✓		✓
Q3	✓		✓		✓			✓
Q4	✓			✓	✓			✓
Q5	✓		✓			✓		✓
Q6	✓		✓		✓			✓
Q7	✓			✓		✓		✓
Q8	Skipped the rest of questions within D1		✓			✓	✓	
Q9				✓	✓			✓
Q10			✓		✓		Skipped the rest of questions within D1	
Q11				✓		✓		
Total	#a	#b	#a	#b	#a	#b	#a	#b
	7	0	6	5	5	6	2	7

Figure 3.18: Dynamic Felder-Silverman Learning Style Questionnaire [18]

Hmedna et al [20] proposed a neural network approach to identify and track user Learning Style. Then provide them the appropriate resources, activities, etc. through adaptive recommendation system.

Villaverde et al [21] proposed a feed-forward neural networks to infer the learning styles of students automatically. The proposed algorithm uses the recent history of system usage so that systems using this approach can recognize changes in learning styles or some of their dimensions over time.

3.1.4 Recommender Systems Optimization for Adaptive Learning

Optimization techniques are widely used with recommender systems to provide the right content with respect to some pre-defined constraints. Those constraints are objectives that will be minimized or maximized.

Wong et al [17] proposed a novel way of modeling learning pathways that combines rule-based prescriptive planning, which could be found in many of the classic Intelligent Tutoring Systems, and Ant Colony Optimization-based inductive planning, for recommending learning paths by stochastically computing past learners' traveled paths and their performances.

Sharma et al [28] proposed an ant-based algorithm Adaptive Content Sequencing in eLearning (ACSeL) for providing learning content to the online learners. The content is categorized into various concept-groups. ACSeL takes as input this information combined with the existing knowledge levels of the individual learners and utilizes it to recommend appropriate levels of concepts to them. The proposed system captures learners' behaviours, updates their knowledge levels and fine-tunes its strategies to recommend the next concept accordingly. The ant-based nature of the system takes into account decisions of the preceding learners and draws them together to provide solutions to the current learners in the system.

Zhao et al [19] proposed an improved ant colony algorithm to optimize the micro-learning path. The improved algorithm is used to recommend the micro-learning units to target learners according to their transitions, and generated a dynamic learning path during the learning process. In the late of algorithm running, the improved algorithm tends to fall into local optimum. This problem could be solved by adjusting the parameters of algorithm in the future, such as the coefficient of pheromone evaporation.

Son et al [22] developed a multi-objective optimization model as a knowledge-based recommender for MOOC's learning. The model operates based on the knowledge-graph and offers recommendations close to the user's goal, but the knowledge-graph has not been easily constructed. They proposed 5 objectives expressed as;

- F1: Minimizing the critical learning path.
- F2: Minimize the learning time.
- F3: Prioritize courses with higher ratings.
- F4: Prioritize courses with higher number of recent enrollments.
- F5: Prioritize lower-cost courses.

Furthermore, they developed a Genetic Algorithm (GA) and an Ant Colony Optimization Algorithm (ACO), to solve the proposed model.

Zhu et al [23] proposed a novel approach to overcome the different learning path preferences of e-learners in different learning scenarios. Based on a knowledge map, the recommended learning path is generated by considering the combination of the domain knowledge structure and cognitive structure of the learners.

Bian et al [24] proposes an efficient approach based on graph theory and an Improved Immune Algorithm (IIA). By introducing the ideas of prey and follow from the AFSA, the performance of the standard IA is significantly improved. Firstly, a learner-centred concept map is created according to the relationship between the learner model and prior knowledge. Then, a traversal algorithm is applied to form a linear concept sequence to maintain the didactical precedence relationships. Next, each concept in sequence is provided with an optimal LO to generate an adaptive learning path, which is converted to a multi-objective combinatorial optimization problem. The IIA is applied to facilitate the search for the near-optimal solution. They used the Felder-Silverman Model to capture user learning style and tagged their content with a 2-tuple to epitomize the Learning Objects characteristics with Concept, Learning Style. Figure 3.19 shows the chromosome frame encoding which is a sequence of learning objects grouped by concepts. They incorporate the Learning Style as an objective that should be met during the search process.

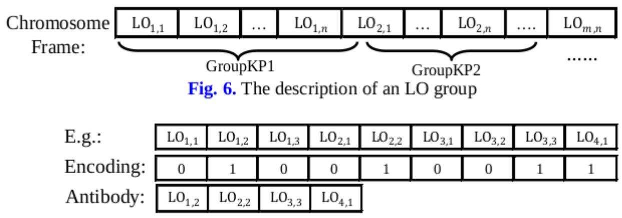


Figure 3.19: Chromosome Encoding [24]

Wan et al [27] proposed a learner oriented recommendation approach based on mixed concept mapping (Figure 3.11) and immune algorithm (IA). They build universal models for learners and Learning Objects respectively, then apply mixed concept mapping to assimilate their attributes. They model the learner oriented recommendation as a constraint satisfaction problem (CSP) which aims to minimize the penalty function of unsatisfied indexes. Finally they exposed an advanced IA which takes the inherent characteristics of personalized recommendation into consideration like learner goal, sentiment, competency, attitude and current knowledge (by a pretest).

Another approach based on gathering information from social network (Linkedin) was also explored. Piao et al [29] proposed a comparison between 3 user modeling strategies based on Job Titles, Education Fields and Skill from a Linkedin profile for personalized MOOC recommendation.

3.2 Graph Representation

As we saw is previous section, graph representation is widely used in the literature. Indeed learning content can be expressed by a dependency graph where nodes represents basic unit of knowledge and edges the relation between those units.

Furthermore, this step helped me to handle the database and understand some basic knowledge about genetic algorithms.

3.2.1 Knowledge Map

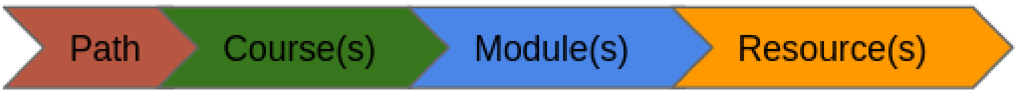


Figure 3.20: Learning Path

Mandarine have a large catalogue of courses that are regrouped by paths. Those paths are a succession of courses and modules which contain resources 3.20. As example, from early 2018 to late 2020, Mandarine proposed **41 Learning Paths, 142 Courses, 1294 Tutorial and 113 Use Cases**. To display this data, graphs are the most common technique. It can express the dependency of courses by connecting nodes between them.

Figure 3.21 show all learning paths for a specific MOOC, the most used one, Microsoft Office 365. For each node type (ie: Path, Course, Module, Resource) we expressed non active node by a lighter color.

We can notice that some courses/modules/resources nodes are common between paths. This mean that a learner can have multiple choices for a learning path.

3.2.2 User Journey

Another usage of graphs is to represent user historical browsing data. Mandarine maintain many trackers to track users activity. From those trackers we can retrace user visits.

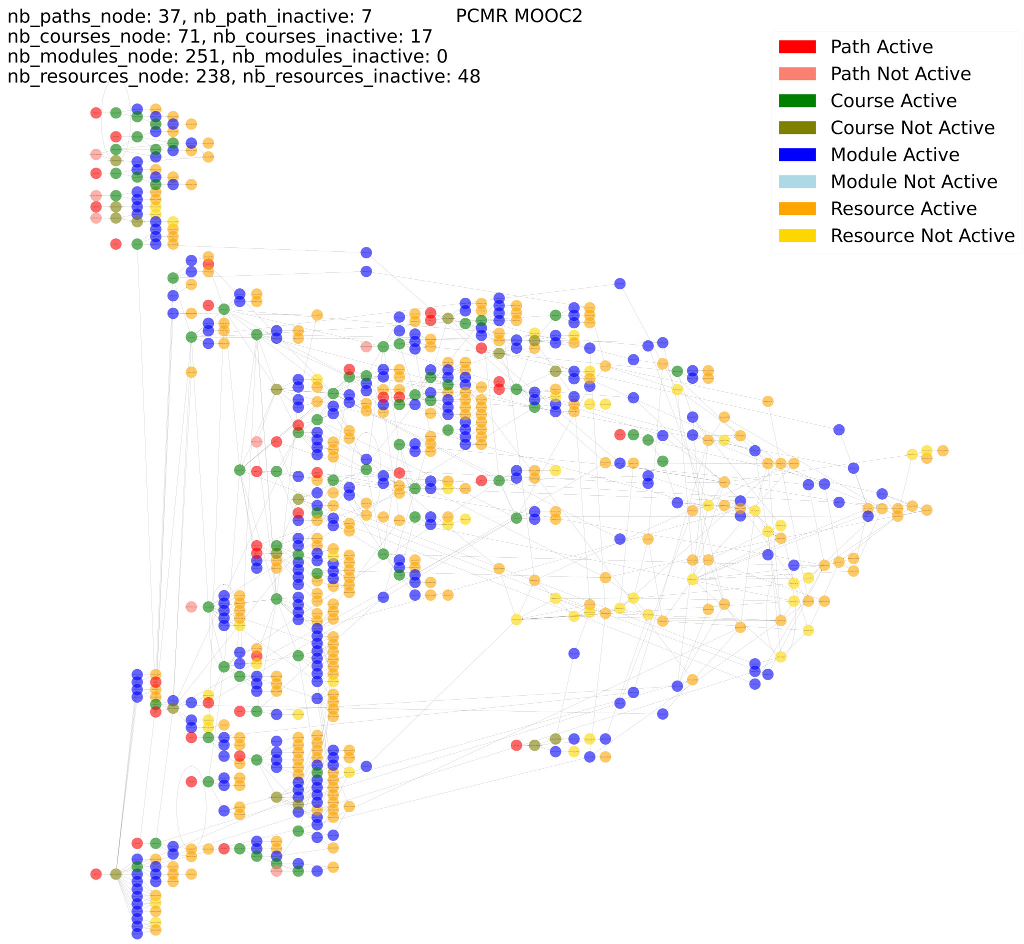


Figure 3.21: Learning Path

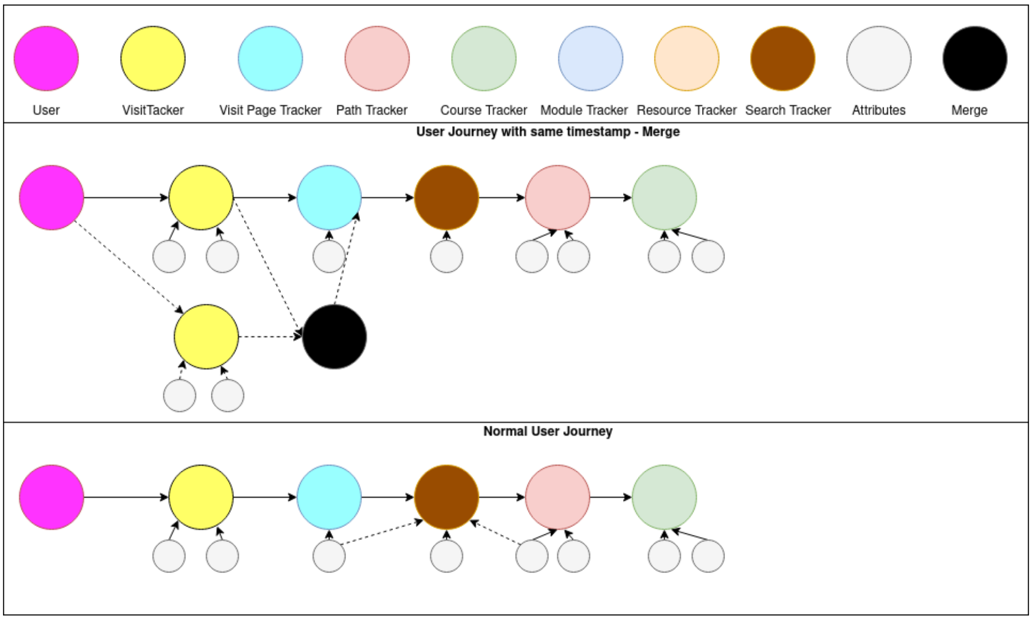


Figure 3.22: User Journey Representation

Figure 3.22 shows an example of a representation of a user journey. Each tracker entry have many attributes related to user browser information (Language, IP, User Agents, ...) that are expressed by grey nodes. We can see that we have 7 different trackers;

- Visit Tracker: Track sessions
- Visit Page Tracker: Track each visited page.
- Path Tracker: Track user visited a path.
- Course Tracker: Track user visited a course.
- Module Tracker: Track user visited a module.
- Resource Tracker: Track user visited resource.
- Search Tracker: Track user search keywords

To handle the fact that a user can have multiple entries with the same timestamp, which means that user clicked on many links in a short time, we merged them with a Merge Node with a black color. Figure 3.23 shows an example of a user journey for a specific user.

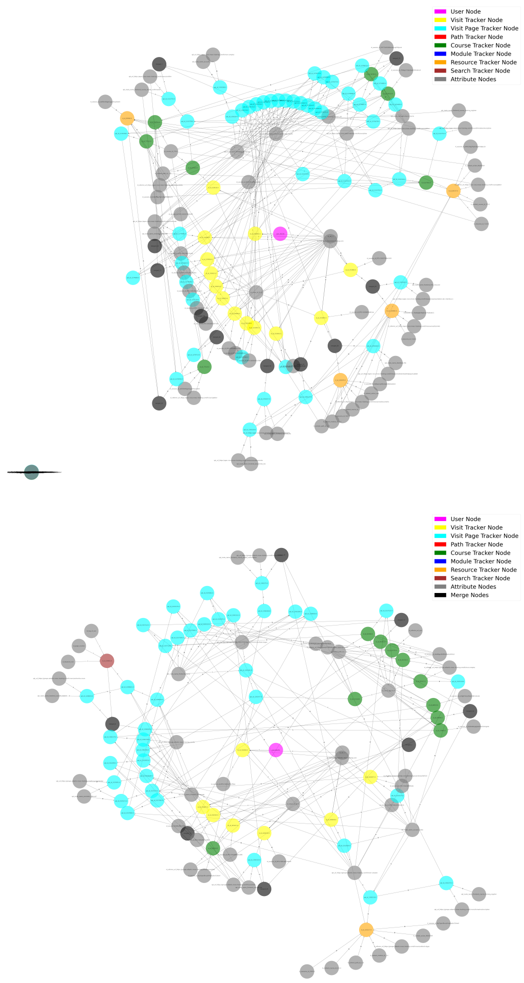


Figure 3.23: User Journey Example

3.2.3 jMetalPY POC

Considering the past representation, and literature example on genetic algorithms, I made a first POC to try genetic algorithms. A simple problem is to find the longest learning path while minimizing the total learning time. This problem can be expressed as a multi-objective problem with two objectives.

To modelize this problem I used the jMetalPy library for multi-objective optimization.

3.2.3.0.1 Encoding We represented a sequence of course as a chromosome (Figure 3.24). We allowed multiple lengths to express the multiplicity of learning paths (ie: for a same learning path we can have different course sequences).

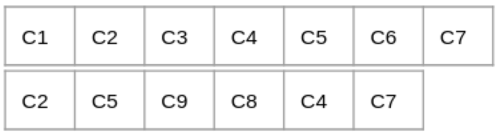


Figure 3.24: Chromosome Encoding

3.2.3.0.2 Fitness As stated before, it's a 2-objective problem.

- F1: Calculate the total learning time - Minimize
- F2: Path length - Maximize

3.2.3.0.3 Selection For the selection we choose the Roulette Wheel selection.

3.2.3.0.4 Crossover We defined a custom crossover which swap courses after a common sequence between two individuals (Figure 3.25).

3.2.3.0.5 Mutation We defined a custom mutation that seek for a new path between 2 random course in the chromosome representation. We extract that new path form our graph representation (Figure 3.26).

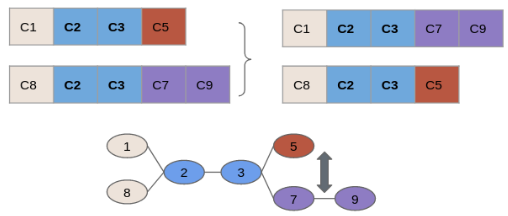


Figure 3.25: Crossover

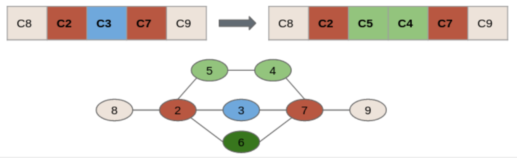


Figure 3.26: Mutation

3.2.3.0.6 Parameters and Results Table 3.1 shows the used parameters for the jMetalPY experiment. Those parameters are randomly chosen just for the sake of example.

Figure 3.27 shows the pareto front of explored solutions.

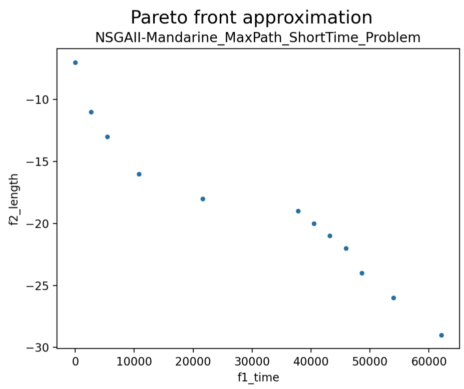


Figure 3.27: Pareto Front Result

Algorithm	NSGA2
Initial population size	100
Offspring population size	100
Crossover probability	0.7
Mutation probability	0.3
Termination	Max evaluation = 1000

Table 3.1: jMetalPy Parameters

3.3 Proposed Model

In this section we will presented our model which is Job/Tool based recommendation, which relies on resources difficulty ranking which can express the evolution of the user (ie: from Beginner to Expert).

The proposed model will be based on a dynamic user profile that capture in a first stage user goal which is a Job Goal or a Tool Goal, and in a second stage if the user complete a Learning Style questionnaire will adapt the content to the learner cognitive aptitude.

We will describe the learning content (resource) by a difficulty approach. For each resource we will infer, from overall users, different difficulty scores that will be used by a genetic algorithm to find the best unique/sequence resource(s) according to different objectives (ie: scores).

3.3.1 User Profile

Like we saw in previous section, adaptive learning is based on a dynamic user profile that resumes users' personal data, preferences, goals (objectives), background knowledge and learning style. We made an exhaustive representation of recurrent used descriptors from literature in Figure 3.28.

Initially user is marked as Beginner. While studying and validating courses his level will increase from Beginner to Moderate, and from Moderate to Expert.

3.3.2 Content/Resource Difficulty Description

Mandarine propose many types of resources. We can find quizzes, videos, SCORMs ... For each resource we proposed a difficulty based on its type (Figure 3.29). For a first try we focused on Video and Quiz resources since they are the most present in the database. We based our difficulty extraction on hypothesis on overall users usage.

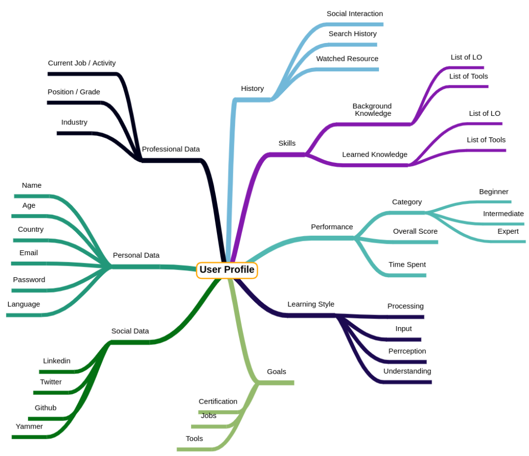


Figure 3.28: Exhaustive User Profile

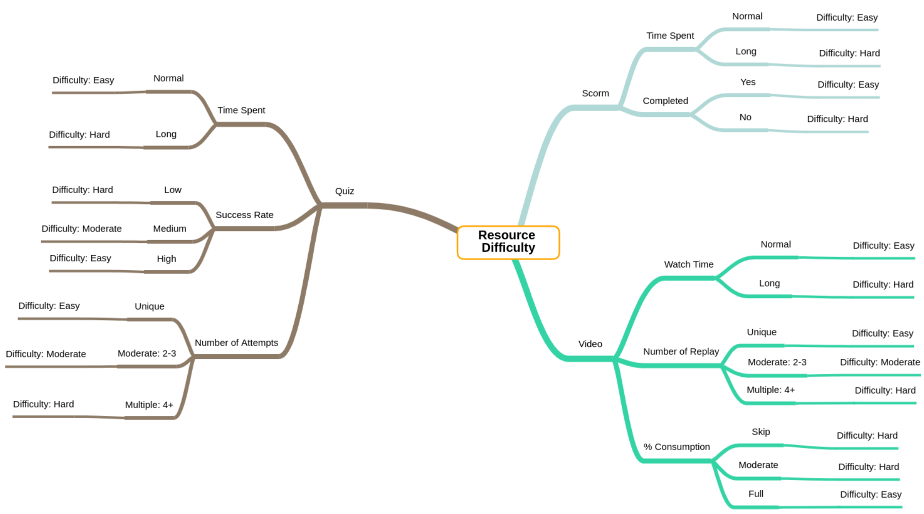


Figure 3.29: Resource Difficulty

To make sure of relying on clean data, we made an exhaustive data analysis on each difficulty criteria by removing outliers with the Interquartile Range (IQR) techniques. After this, we extracted statistical measurements like mean, median and standard deviation to set the appropriate thresholds.

Mandarine follows strict RGPD policy (*Règlement Général pour La Protection des Données*) which are stated by the French Government to protect personal users data. We applied our data analysis on anonymized data to preserve users integrity.

3.3.2.1 Video

We can express video difficulty by **Watch Time**, **Number of Replay**, **Consumption Percentage**. For each parameter we proposed hypothesis to classify the content.

3.3.2.1.1 Video Difficulty Hypothesis

- Video that have average watch time (over users) greater than resource length should be marked as Hard.
- Video that have average number of replay (over users) greater than average number of replay (over resources) should be marked as Hard.
- Video that have maximal Percentage Consumption and average time watch (over users) should be marked as Easy, and vice versa.
- If Percentage Consumption is Moderate and Watch Time is Long, video should be marked as Hard.
- If Percentage Consumption is High and Multiple Replay, video should be marked as Hard.
- If Percentage Consumption is High and Unique Replay, video should be marked as Easy.
- If Watch Time is Long, Multiple Replay and High Percentage Consumption, video should be marked as Hard.

3.3.2.2 Quiz

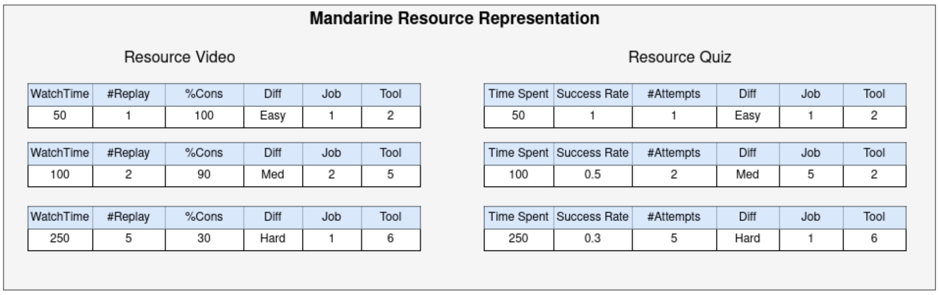
We can express quiz difficulty by **Success Rate**, **Time Spent**, **Number of Attempts**. For each parameter we proposed hypothesis to classify the content.

3.3.2.2.1 Quiz Difficulty Hypothesis

- Quiz that have high success rate should be marked as Easy and vice versa (ie: Hard).
- Quiz that have high Time Spent should be marked as Hard.
- Quiz that have normal (average) Time Spent should be marked as Easy.
- Quiz that have high number of attempts should be marked as Hard and vice versa (ie: Easy).
- Quiz that have high Attempts and High Success Rate should be marked as Hard.

3.3.3 Optimization

In order to use multi objective optimization to find the best content to recommend, we will use Genetic Algorithm. We represented a resource by its difficulty scores (Figure 3.30).


Mandarine Resource Representation

Resource Video

WatchTime	#Replay	%Cons	Diff	Job	Tool
50	1	100	Easy	1	2

WatchTime	#Replay	%Cons	Diff	Job	Tool
100	2	90	Med	2	5

WatchTime	#Replay	%Cons	Diff	Job	Tool
250	5	30	Hard	1	6

Resource Quiz

Time Spent	Success Rate	#Attempts	Diff	Job	Tool
50	1	1	Easy	1	2

Time Spent	Success Rate	#Attempts	Diff	Job	Tool
100	0.5	2	Med	5	2

Time Spent	Success Rate	#Attempts	Diff	Job	Tool
250	0.3	5	Hard	1	6

Figure 3.30: Resource Representation

We encode then a sequence of resources as a chromosome and use difficulty criteria as objectives. As a first try we expressed objectives like (Figure 3.31):

- Beginner Objectives
  - F1: Minimize Watch Time
  - F2: Minimize #Replay
  - F3: Maximize %Cons
- Intermediate Objectives

- F1: Minimize Watch Time
- F2: Maximize #Replay
- F3: Maximize %Cons

- Expert Objectives
  - F1: Maximize Watch Time
  - F2: Maximize #Replay
  - F3: Minimize %Cons

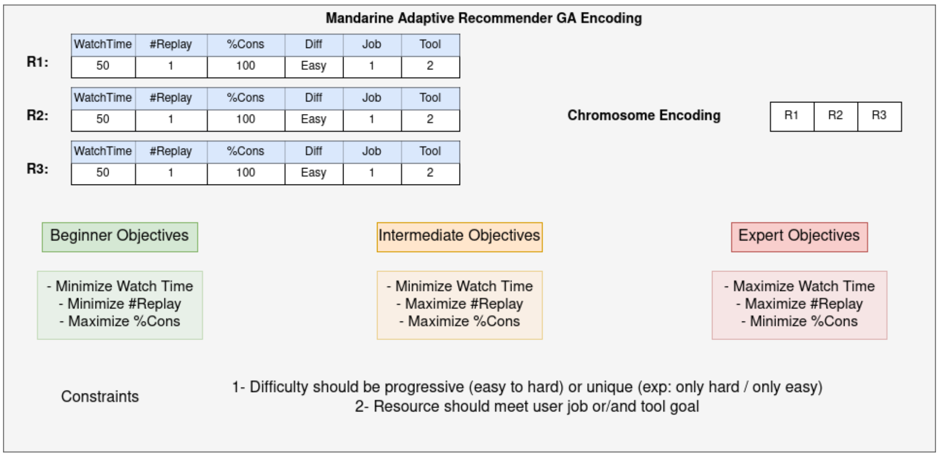


Figure 3.31: Resource Encoding

3.3.4 Initialization

We saw in past section that recommender systems suffer from cold start problem which can be resumed by the lack of information about new users. To mitigate this problem we proposed a 3-stage initialization model showed in Figure 3.32. With this approach we don't force users to complete their profile and adapt our recommendation depending on how much information we have about the user.

If the user don't specify a goal (Job or Software) recommendation will be based on simple statistical approach. Most watched content will be recommended for the user, as well as the newest content.

If the user specify a software or job goal, the recommendation will be focused on those criteria. User will be marked as **Beginner** and the right content will be displayed.

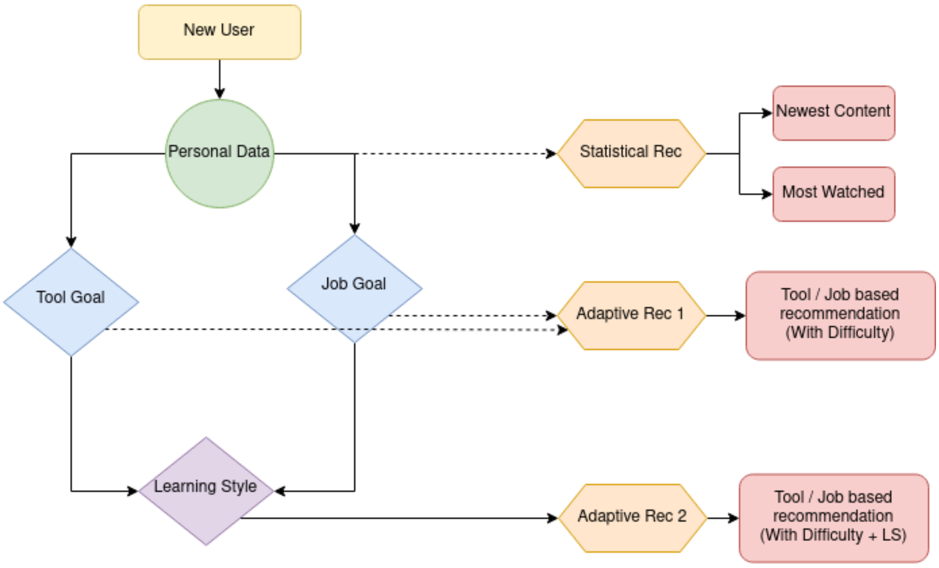


Figure 3.32: Proposed Initialization

3.3.5 Overall Architecture

Figure 3.33 show an overall architecture of our model. We can distinguish the 3-step initialization with different User Profile attributes used for recommendation.

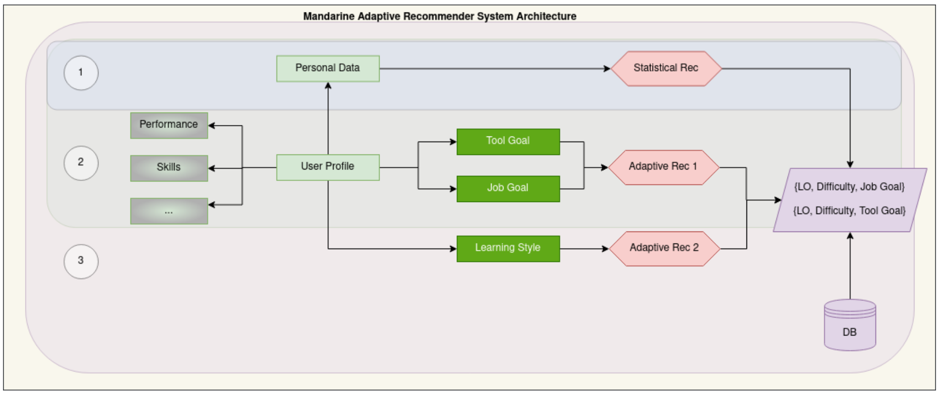


Figure 3.33: Proposed Initialization

Initially a user is marked as Beginner and evolves to Expert during the learning journey. This transition is made by memorizing the learned knowledge which are tagged by difficulty. For existing users without a Job/Tool Goal defined we propose to infer their interest (according to a Tool or a Job) from their activity (ie: visited content) and prioritize last learned knowledge.

Chapter 4

Future Work

This paper presented a large overview of state of the art techniques for recommending learning resources. It explores the notion of Adaptive Learning by personalizing the recommendation based on user objectives and preferences while taking into account their progress during the learning journey.

For a first modelization, we focused on incorporating the users' goals by defining a Job-based and a Tool-based recommendation by a difficulty ranking approach over resources. However, we didn't take into account the Learning Style dimension. Indeed, we faced some limitations in mapping resources to a learning style due to the lack of information about the content.

Furthermore the implementation is still in progress. We choose to The process of extracting and cleaning the data is time-consuming and needs more time to be completed.

Other perspectives are oriented on the content description. We used established dependency between courses to build our Knowledge Graph but state of the art techniques rely on Natural Language Processing techniques to extract topics from resources and apply similarity to establish relations between those resources.

Chapter 5

Conclusion

Recommender systems are widely used to predict users' interests, which increases content consumption. From basic to more complex systems, the main differences between them are the amount of information gathered about a user and the ability of the system to adapt to them.

In this work, we presented state of the art techniques used by recommender systems. We presented common techniques such as Collaborative Filtering and Content Based Filtering, which face the problem of *One Size Fit All Problem* (OSFAP). OSFAP is expressed by recommending content in the same way to users with different profiles. They mainly ignore the users' progress and changes during the learning process. Furthermore, they ignore the users' knowledge background and their ability to learn.

To mitigate this problem, we explored the notion of Adaptive Learning which emphasizes individual differences in learning preferences, goals, abilities, and knowledge background. We exposed major techniques used for it, like the Learning Style approach. Furthermore, we presented common multi-objective techniques used with those systems, like Genetic Algorithms or Ant Colony Optimization to optimize the search operation.

Finally, we proposed a Job/Tool based Adaptive Recommender System model which uses a multi-objective Genetic Algorithm to efficiently seek for the right resource in the large search space possibilities proposed by Mandarine.

References

[1] Siemens, G. (2013). Massive open online courses: Innovation in education. In McGreal, R., Kinuthia W., Marshall S. (Eds), Open educational resources: Innovation, research and practice (pp. 5–16)

[2] Aristomenis S. Lampropoulos and George A. Tsihrintzis, 2015. Machine Learning Paradigms, Springer International Publishing Switzerland 2015, DOI https://doi.org/10.1007/978-3-319-19135-5

[3] Yöş, K. (2020). Deep Learning For Implicit Feedback-based Recommender Systems.

[4] Lops, P., Gemmis, M. D., Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. Recommender systems handbook, 73-105.

[5] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. Recommender systems: an introduction. Cambridge University Press, 2010.

[6] John H Holland. Genetic algorithms and the optimal allocation of trials. SIAM Journal on Computing, 2(2):88–105, 1973

[7] Whitley, D. (1994). A genetic algorithm tutorial. Statistics and computing, 4(2), 65-85.

[8] Alberto Colorni, Marco Dorigo, Vittorio Maniezzo, et al. An investigation of some properties of an" ant algorithm". In Ppsn, volume 92, 1992.

[9] Mödritscher, F., Garcia Barrios, V.M., Gu tl, C. (2004). Enhancement of SCORM to support adaptive E-Learning within the Scope of the Research Project AdeLE. In Proceedings of E-Learn 2004, Washington, DC, USA (pp. 2499–2505). Retrieved on January 25, 2010 http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.60.6119rep=rep1type=pdf

[10] Deng, Y., Huang, D., Chung, C.-J. (2017). Thoth lab: A personalized learning framework for cs hands-on projects. In Proceedings of the 2017 ACM SIGCSE technical symposium on computer science education (pp. 706). ACM.

[11] Amir Hossein Nabizadeh, José Paulo Leal, Hamed N. Rafsanjani, Rajiv Ratn Shah, Learning path personalization and recommendation methods: A survey of the state-of-the-art, Expert Systems with Applications, Volume 159, 2020, 113596,ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2020.113596

[12] Nayak, S. (2020). Fundamentals of Optimization Techniques with Algorithms. Academic Press.

[13] Pareto, V. (1896). Cours d'économie politique: professé à l'Université de Lausanne (Vol. 1). F. Rouge.

[14] Klammer, M., Dybowski, J. N., Hoffmann, D., Schaab, C. (2015). Pareto optimization identifies diverse set of phosphorylation signatures predicting response to treatment with dasatinib. PLoS one, 10(6), e0128542.

[15] Blum, C., Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM computing surveys (CSUR), 35(3), 268-308.

[16] Curry, L. (1981). Learning preferences in continuing medical education. Canadian Med Assoc J., 124, 535–6.

[17] Wong, L. H., Looi, C. K. (2009). Adaptable learning pathway generation with ant colony optimization. Journal of Educational Technology Society, 12(3), 309-326.

[18] Nafea, S. (2019). A Novel Adaptation Model for E-Learning Recommender Systems Based on Student's Learning Style.

[19] Zhao, Q., Zhang, Y., Chen, J. (2016, December). An improved ant colony optimization algorithm for recommendation of micro-learning path. In 2016 IEEE International Conference on Computer and Information Technology (CIT) (pp. 190-196). IEEE.

[20] Hmedna, B., El Mezouary, A., Baz, O., Mammass, D. (2017). Identifying and tracking learning styles in MOOCs: A neural networks approach. International Journal of Innovation and Applied Studies, 19(2), 267.

[21] Villaverde, J. E., Godoy, D., Amandi, A. (2006). Learning styles' recognition in e-learning environments with feed-forward neural networks. Journal of Computer Assisted Learning, 22(3), 197-206.

[22] Son, N. T., Jaafar, J., Aziz, I. A., Anh, B. N. (2021). Meta-heuristic algorithms for learning path recommender at MOOC. IEEE Access, 9, 59093-59107.

[23] Zhu, H., Tian, F., Wu, K., Shah, N., Chen, Y., Ni, Y., ... Zheng, Q. (2018). A multi-constraint learning path recommendation algorithm based on knowledge map. Knowledge-Based Systems, 143, 102-114.

[24] Bian, C. L., Wang, D. L., Liu, S. Y., Lu, W. G., Dong, J. Y. (2019). Adaptive learning path recommendation based on graph theory and an improved immune algorithm. KSII Transactions on Internet and Information Systems (TIIS), 13(5), 2277-2298.

[25] Vesin, B., Budimac, Z. (2013). Applying recommender systems and adaptive hypermedia for e-learning personalizatio. Computing and informatics, 32(3), 629-659.

[26] Kritikou, Y., Demestichas, P., Adamopoulou, E., Demestichas, K., Theologou, M., Paradia, M. (2008). User Profile Modeling in the context of web-based learning management systems. Journal of Network and Computer Applications, 31(4), 603-627.

[27] Wan, S., Niu, Z. (2016). A learner oriented learning recommendation approach based on mixed concept mapping and immune algorithm. Knowledge-Based Systems, 103, 28-40.

[28] Sharma, R., Banati, H., Bedi, P. (2012). Adaptive content sequencing for e-learning courses using ant colony optimization. In Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011) December 20-22, 2011 (pp. 579-590). Springer, New Delhi.

[29] Piao, G., Breslin, J. G. (2016, July). Analyzing MOOC entries of professionals on LinkedIn for user modeling and personalized MOOC recommendations. In Proceedings of the 2016 conference on user modeling adaptation and personalization (pp. 291-292).

[30] Urdaneta-Ponte, M. C., Oleagordia-Ruíz, I., Méndez-Zorrilla, A. (2022). Using LinkedIn Endorsements to Reinforce an Ontology and Machine

Learning-Based Recommender System to Improve Professional Skills. Electronics, 11(8), 1190.

[31] Dai, K., Vilas, A.F., Díaz Redondo, R.P. (2017). A New MOOCs' Recommendation Framework based on LinkedIn Data. In: , et al. Innovations in Smart Learning. Lecture Notes in Educational Technology. Springer, Singapore. https://doi.org/10.1007/978-981-10-2419-1_3

[32] Alruwaili, F., Alahmadi, D. (2021). Enhanced Clustering-based MOOC Recommendations using LinkedIn Profiles (MR-LI). International Journal of Advanced Computer Science and Applications, 12(8).