From Music to Museum: Applications of Multi-Objective Ant Colony Systems to Real World Problems

Pierre-Edouard Osche
CNRS - LORIA - Univ. of Lorraine
Vandoeuvre, France
pierre-edouard.osche@loria.fr

Sylvain Castagnos
CNRS - LORIA - Univ. of Lorraine
Vandoeuvre, France
sylvain.castagnos@loria.fr

Anne Boyer
CNRS - LORIA - Univ. of Lorraine
Vandoeuvre, France
anne.boyer@loria.fr

ABSTRACT

Recommender systems are a flourishing domain in computer science for almost 30 years now. This rising popularity follows closely the number of data collected all around the world. Each and every internet user produces a huge amount of content during his lifetime. Recommender systems proactively help users to navigate these pieces of information by gathering, and selecting the items to users' needs. In this paper, we discuss the possibility and interest of applying our Multi-Objective Ant Colony System called AntRS to recommend items in different application domains. In particular, we show how our model performs better than the state-of-the-art models with music dataset, and describe our work-in-progress with the museum of fine arts in Nancy (France). The motivation behind this change of application domain is the recommendation of progressive sequences rather than unordered lists of items.

KEYWORDS

Recommender Systems; Multi-Agent Systems; Multi-Objective Sequences; Music Online Services; Cultural Heritage

1 INTRODUCTION

Noah, an athletic young man, decided to listen to music while running, and found the perfect app for that. It allowed him to generate a playlist of recent songs that are adapted to his preferences. This application knew how to renew its recommendations from one session to another making him discover new music while, in the same time, proposing titles that he already knew and loved.

A few blocks away, Kate decided to go to a museum before her next university course. As she had only 2 hours free, she installed on her smartphone the mobile application provided by the museum to be recommended an ideal path taking into account both her eclectic tastes and her aversion to the crowd. She was willing to travel longer distances to avoid congestion points. But most importantly, she did not want to miss her favorite painting.

The same scientific problematic is behind these 2 stories: being able to provide adapted and coherent sequences to users. The target state can be a relevant painting to see, a museum exit to reach or an energetic song to recommend when finishing the race. But the way users get to this target state is also important. The sequences should be as personalized and progressive as possible while offering a good compromise between multiple objectives (precision, diversity, density of population and distance within the museum for Kate; precision, novelty, serendipity and music features such as recency, and tempo for Noah).

To address these issues in the music application domain, we proposed a multi-objective recommender system called AntRS [28]. The latter belongs to the multi-agent system family in which agents wander through a graph where nodes represent the items of the domains (music titles) and edges represent links between those items. AntRS is a multi-objective model because it integrates several classic objectives considered for a good recommendation (similarity, diversity, novelty, preferences), and searches for an optimum between all of them. In this paper, we aim at investigating the generic nature of our model. We argue that it is possible to add or remove objectives on the fly, and that AntRS can be applied on different domains. In this paper, we thus propose to extend AntRS with new results showing the flexibility of our model through a new dimension called progressiveness. We also discuss the possibility to translate our model to another application domain (cultural heritage) which is an on-going research.

This paper is organized as follows: Section 2 presents the related work on multi-objective recommenders and their application in music and museum domains. Section 3 describes our AntRS model. Section 4 describes the experiments carried out and the results obtained in music as well as our plan to apply our model in the museum space. Finally Section 5 concludes this paper and presents our perspectives.

2 BACKGROUND

2.1 Ant Colony Systems

The Ant Colony Optimization algorithms (ACO) are Nature-inspired programming algorithms that computationally reproduce the pheromone communication between ants to find shortest paths between a starting node and a target node. Our model, called AntRS, is derived from one of those systems: the Ant Colony System (ACS) algorithm [9]. The latter relies on a pseudo-random proportional rule to favor exploitation of the pheromone information, and a local pheromone update to diversify the search performed by ants during one iteration. In other words, each time an ant takes an edge, it deposits some pheromones along its way, regardless of the quality of the path. As explained by Dorigo, this version of the ACO algorithms is known to "*decrease the pheromone concentration on the traversed edges, thus encouraging ants to choose other edges and, hence, to produce different solutions. This makes it less likely that several ants produce identical solutions during one iteration*". We took inspiration from the ACS algorithm so as to identify relevant items in the very large exploration space of online services (e.g. millions of songs on music streaming services, and hundreds of paintings in a museum), recommend them to users, and evaluate the quality of recommended lists through several joint metrics (precision, diversity, novelty...). We refer the reader to the Dorigo and Birattari paper [9] for all the details about the model as we will use some of the notations/equations of their work in the section 3. In the recent years, many studies have been using multi-agent systems and more specifically the ACO algorithms. One of the first study using the ACO in recommender systems was from Semet et al. [34] in an e-learning system. The authors built a graph where the nodes represent educational resources and the edges have transition probabilities associated to them. The authors also introduce success and failure pheromones as an additional information. The system could however be improved as there is no user model and it also needs professors to evaluate the possible paths, which could be an important drawback in a large database. Optimization problems have also been tackled by ACO algorithms with success. We can cite [12] which proposed a framework to use multiple ant colony system for the vehicle routing problem with time windows, a classic multi-objective problem where multiple objectives are conflicting with each other. Since this paper is more focused on applying our model in two different domains (music and museum), the next two subsection will focus on recommender systems in these particular domains.

2.2 Recommender Systems in Music

When it comes to recommending music, a system can either propose a list of independent items at each time step, or it can suggest a sequence of items [29]. Let us note that Sequence-Aware Recommender Systems may refer to the order of the past events (i.e. looking for co-occurrence patterns [3] or for sequential patterns [16] in past sessions) or the order of the future recommendations (such as in playlists [17, 25]). As recent research has found little evidence that the exact order of songs actually matters to users [36], we first limited our state-of-the-art to the recommendations of lists.

Besides its ability to generate recommendation lists, our system sees the selection of relevant items as a multi-objective optimization problem. Most recommenders solely focus on the accuracy (precision and recall) [32]. Yet, more and more systems attempt to find a compromise between several dimensions such as precision, serendipity and novelty [19], or between precision and diversity [23, 47] for example. There are several ways to address this issue. One can either look for a set of Pareto solutions, considering that a solution is optimal if it is not possible to make any objective better off, without making at least one objective worse off [48]. In that case, the goal of recommender systems is to produce as many solutions as possible from the Pareto-front solution set. Or one can rank the items in a single list by aggregating or reordering the results of each objective taken separately [30]. This list can be computed at once [14], or come out of a 2-step process consisting in building several lists of candidate items for the active user and in merging them [11, 13, 31, 40, 41, 43]. Recommending several Pareto solutions offers the advantage to leave the choice to the active user. It can be interesting in some application domains such as e-commerce where an explicit validation process from the user is mandatory. However, in the context of online music services, it is not conceivable to request a user decision at each timestep. The songs must come one after another without disturbing the user in his/her main current task. For this reason, we focused on recommenders which produce only one solution (i.e. only one list of recommendations).

As a conclusion, our model AntRS is a multi-objective single-list recommender. Those recommenders usually suffer from several limitations: they are dependent from the application domain (any change in the set of objectives has a drastic impact on the implementation of the model) and they are very time-consuming. To bypass these difficulties, we rely on a Multi-Agent System (MAS), and more precisely on an Ant Colony System explained below. MAS have multiple advantages in our context such as the low computational complexity, the ability to tackle multi-objective problems and the resilience to changes. In this paper, we first deepen the results we got on a music dataset in [28]. Then, we show how this model can be translated to another application domain. We chose to apply AntRS to Cultural Heritage, and more specifically to museum guides, since the visitors' lists of recommendations should notably be ordered in sequences so as to minimize the distance to travel and to tell a coherent story. Thus, this change of context allows us to extend our model with the notion of progressiveness.

2.3 Recommender Systems in Museums

In the recent years, a trend emerged in recommender systems because experimentations showed that users needed more than precise recommendations. Metrics like diversity, novelty, serendipity slowly were more and more considered into recent systems alongside user's contextual information (Is the user alone? Is she/he at work or at home? …) with Context-Aware Recommender Systems (CARS) [1]. This recent trend follows the development of technologies capable of feeding models with contextual and geographical information like smartphones, GPS or indoor positioning system [8]. The recommender systems that take into account all of these pieces of information are gathered under the name Location-Aware Recommender Systems (LARS) [22].

The tourism domain is an obvious choice to field test algorithms taking into account the context and the location of user in a real physical environment. As early as the beginning of the 2000s, some experimentations were conducted in e-tourism to help visitors find their way to interesting areas. Cheverst *et al.* [6] were one of the first to propose a system taking into account contextual information (location, date, hour of the day, opening and closing hours for attractions,…). However, no visitor model were created and, thus, no recommendation were made. Chou *et al.* [7] propose a system were works of art are modelled as well as the visits themselves. They then used these data to model visitors preferences and recommend them works of art, areas of interest or even a route. To model visitor preferences, studies have shown that viewing time is correlated with interest for simple item like clothes. Though it still has to be tested with more complex items like exhibits (i.e. it might be easier and less time-consuming to tell if we like a t-shirt or not compared to a 5x3m painting).

In the recent years, a lot of studies expanded the work already done by using new technologies as well as advancement in recommender systems in general. Some of those studies are particularly interesting for our current work. For example, Van Hage *et al.* [37] created a model capable of guiding visitors through the Rijksmuseum in the Netherlands. The algorithm takes into account the visiting time specified by the visitors, the ratings given on already seen exhibits or the distance between the different items available for the recommendation. However this study lacks some important points: the authors only simulated the visit and visitors had to manually rate what they saw to feed the recommender system, potentially degrading the user experience. Woerndl *et al.* [42] tried to alleviate this problem by creating an algorithm which proactively proposes recommendation to the user based on internal metrics computed on contextual information. Nonetheless, the users still had to validate afterwards if the recommendation was made at a good moment or not.

Some studies focused on other subject specific to tourism like how the crowd might impact the visitor experience and how to manage it as best as possible [27]. The authors in [45] performed an extensive study in the Louvre museum on more than 24,000 visitors with Bluetooth sensors and a mobile application to track their location, giving some interesting insights on the behavior of the visitors. Based on the duration of their visit, the visitors were divided in two groups: the short-stay and the long-stay visitors. It was found that these two groups, despite not having the same visiting time, were mostly looking at the same works of art and taking the same paths. These findings illustrate the importance of having a way to manage the crowd.

Finally, other studies highlighted how a single person may behave while visiting a museum. Veron and Levasseur [39] in an ethnographic study in the Centre Georges Pompidou were able to categorize four visiting style: (1) the ant who looks at every artwork in the scenography order, (2) the butterfly who performs a zigzag visit, alternating between each side in a room, (3) the grasshopper who progresses in the museum by doing big leaps to only look at a few selected pieces interesting to him, and (4) the fish who stays at the center of the room to have a quick look from away to all the exhibits before going into another room. Recent studies have tried to retrieve or to use these visiting styles in recommender systems. Like so, the authors in [21] did an experiment where 140 visitors were put in a room with paintings and an audio-guide. Based on how the visitors behave, the authors conclude that the four visiting style were retrieved and that it was possible to predict the visiting style of a single visitor relatively early. They nuanced their results because, unless [39], they showed that visitors were able to change from a visiting style to another during the visit. This study did however not include a recommender system. Finally, Lykourentzou *et al.* [24] developed a simulation were entities wandered through a virtual museum based on the four visiting styles. They then showed that with intelligent recommendations, it was possible to increase the satisfaction of the entities at the end of the visit. The study was however not verified in a real life setting.

3 ANTRS MODEL

AntRS has been built with several goals in mind: (1) be as generic as possible, (2) be able to include several competing objectives in a single list, (3) be resilient to changes in the environment (new items, new preferences,...). As explained previously, our model takes inspiration from the ACS algorithm because the latter gathers all the quality needed to satisfy those objectives. However, we want to point out the differences between the classic ACS and our model AntRS. In this section, we will outline those differences as well as new results in comparison to our previous study.

3.1 Graph creation

The first step before running our model is to create the graph the agents will roam on. To do so, we created nodes corresponding to items of the domain and applied a selection algorithm to create edges between these nodes, as working with a complete graph was beyond the realm of possibilities giving the usually huge number of items. At this point, we formulated two assumptions to help us construct the graph: (A1) past sequences created by previous users represent useful domain knowledge which should be exploited, and (A2) past sequences done by previous users are not always the best possible ones and could have been improved with clever recommendations. The general intuition behind these two assumptions can be explained like so:

- (A1): users consult items that satisfy their needs in the best possible way, thus there is interesting patterns and knowledge to exploit from these consultations;
- (A2): given that users do not generally browse all the existing items to select the perfect ones for their needs, it is safe to assume that another sequence could have satisfied the needs of users in a better way.

However in this work, we did not prove or disprove these assumptions, we simply used them to help building our graph.

To take into account these two hypotheses, we first computed the number of transitions (i.e. co-consultations) between each pair of items in our dataset. We added all the transitions above a specific threshold, and only some of those below this threshold as edges. Finally, if a given connectivity degree was not reached, we added new edges between items who were not connected in our dataset to allow our model to discover new potential interesting paths not known by users. The process of creating an edge is shown in Equation 1.

$$e_{ij} = \begin{cases} \text{if } t_{ij} \geq m \\ \text{or if } t_{ij} < m \text{ and } q < t_{ij} \text{ where } q \in [min\ t_{il}; \log(max\ t_{il})] \\ \text{or if } t_{ij} = 0 \text{ and if } deg(i) < d \text{ then pick a random} \\ \text{transition until } deg(i) = d \end{cases} \tag{1}$$

where e_{ij} is the presence of an edge between vertices i and j, t_{ij} is the number of transitions performed from item i to item j by the users in the dataset, m is the threshold where transitions are not directly added to the graph as edges, $q \in [min\ t_{il}; \log(max\ t_{il})]$ is a random variable uniformly distributed, $min\ t_{il}$ is the minimal number of transitions between the item i and all the others items $l \in V_i$ where at least one transition has been found, $deg(i)$ is the current degree of the node i in the graph and d is a parameter specifying the minimal degree each vertex must have in the final graph.

For our first study in [28], we only created one graph using all the data available with all the training users. This choice was justified by the fact that it takes less time to do so as the graph can be huge. Moreover, the edges in the graph reflect the global knowledge of all the user in the system. In this paper, we tested another way to proceed by creating a graph per test user. To do so, we computed the n most similar neighbours of the active user based on the known ratings with a classical cosine similarity measure. Those n users were then used to create the graph as explained above.

3.2 Objectives

It is now widely admitted that the sole precision is not sufficient to produce good recommendations to users. In [28], we proposed to define a set of 4 concurrent objectives that often have to be considered in the literature while recommending a list of items. In this paper, we aim at showing that these objectives are transposable in different application domains, guaranteeing the generic nature of our approach. For the purpose of this study, we also added a fifth objective – called progressiveness – whose relevance will be discussed depending on the application domain.

The ability to add objectives, modulate their importance or remove them on the fly is essential for having an adaptive model. To address this issue, we chose to include as many colonies as objectives in our model where each colony is specialized in maximizing its own objective. To do so, we modified the way the ACS model computes the distance d between two nodes of the graph. The rest of this Section describes the equations used to compute the distance for each colony.

Similarity - The goal of this colony is to find items as similar as possible of what the user previously viewed or is currently viewing. A lot of methods exist to compute the similarity of two vectors and, based on our dataset and on the metadata available, we decided to use a cosine similarity measure [35]. To compute the distance value on the edges of the graph, we simply computed the cosine similarity between the two items represented by the vertices. More formally, for an edge (i, j), its associated distance d_{ij} is computed with the cosine similarity between the vectors of the descriptive characteristics of the items i and j.

$$d_{ij} = \frac{1}{sim(C_i, C_j)} \tag{2}$$

where C_i are the characteristics of the item i. The item characteristics depend obviously on the dataset and on the meta-information available but, we can formalize that each item of the dataset is described by n characteristics as follow $C_i = \{c_1, c_2, \ldots, c_n\}$. We used the multiplicative inverse to transform the similarity metric $sim \in [0; 1]$ into a distance $d \in [1; +\infty]$. Therefore, a distance value near 1 on an edge (i, j) means that the two items i and j are similar.

Diversity - This characteristic and the similarity are often described together as they are both related to the distance/correlation between the items liked by the user and his recommendations. But unlike similarity, diversity depicts how dissimilar two items are relatively to each other. Similarity and diversity are complementing each other in the sense that they are both needed to adapt the system to the needs of different users [18]. To compute this objective, we chose to apply one of the classic diversity metric which is obtained by computing the inverse of the similarity between two items, as shown in Equation 3. As for the similarity, we used the multiplicative inverse of the diversity to obtain a distance $d \in [1; +\infty]$.

$$d_{ij} = \frac{1}{1 - sim(C_i, C_j)} \tag{3}$$

Novelty - The goal of this colony is to search for items that are not yet known by the user. Novelty is an important characteristic of a recommender system to avoid a potential lack of interest of users due to too much foreseeability in the recommended items [38]. To determine if an item is novel or not relatively to a specific user, we used the work of Zhang [46] who defined the novelty as a notion composed of three characteristics: (1) Unknown: the item is unknown to the user, (2) Satisfactory: the item is liked by the user, (3) Dissimilarity: the item is dissimilar to the other items known by the user. The author proposed to evaluate the novelty of the item i for the user u as follow:

$$novelty(i, u) = p(i|unknown, u) \cdot dis(i, pref_u) \cdot p(i|like, u) \tag{4}$$

where $p(i|unknown, u)$ is the probability that the user u does not know the item i, $dis(i, pref_u)$ is the dissimilarity between i and the set of items in the users' profile and $p(i|like, u)$ is the probability that u will like i. However, the dissimilarity and the satisfaction of the user relatively to i are closely related to other objectives in our model, respectively maximized by the diversity colony and by both the preferences and the similarity colonies. Hence we decided to trim down the Equation 4 to the probability $p(i|unknown, u)$ only (see Equation 5).

$$p(i|unknown, u) = -log(1 - pop_i) \tag{5}$$

where pop_i is the popularity of item i.

Preferences - The preferences characteristic corresponds to what the user really likes. It intersects with the similarity notion but, again as with diversity and novelty, we think that preferences express another aspect of a good recommendation for a user. The similarity characteristic allows the recommender to propose items that are similar to the preferences of the user, but it is not guaranteed that she/he will like those items. It is for example perfectly common to both like and dislike some songs coming from the same album and artist, yet those songs will probably be treated as very similar relative to each other. The preferences characteristic favors items that are known to be liked by the user.

The goal for this colony is to find a sequence in the graph prioritizing items that are already known to be liked by the user. Thereby, items must have criteria conveying how the user like an item or not. This can be done either with explicit feedback (*e.g.* item rating,...), with implicit feedback (*e.g.* number of times the user viewed an item,...) or with a combination of both. The nature of the feedback will heavily depend on the domain, but we can formalize that each collected information concerning the behavior of a user on an item must be taken into account. Let C_u be the set of criteria representing all the actions that a user u may perform on the items of the system, thus $c_{u,i}$ is the sum of all interactions specific to a single criterion c that a user u performed on an item i (*e.g.* the number of times a user u viewed i). To aggregate all the different interactions possible in a single value, we use the presumed interest formula proposed by Castagnos *et al.* in [5] and described in the Equation 6.

$$presumed\ interest_{u,i} = v_{min} + \frac{\sum_{c \in C} (w(c) \cdot c(u, i))}{\sum_{c \in C} w(c)} \cdot \frac{(v_{max} - v_{min})}{c_{max}} \tag{6}$$

where $c(u, i)$ corresponds to normalized values given to the item i by the user u to each criterion c, $w(c)$ is the weight of the criterion c, v_{min} and v_{max} are the minimal and maximal expected values for the presumed interest and c_{max} is the maximal value that $c(u, i)$ can take regardless of the criteria. In our case, we considered the following criteria for each song: number of consultations, number of skips, number of bans (when the user do not want to listen to the song ever again), and number of likes.

Progressiveness - While the majority of the recommender systems propose a list of items ranked from most likeable to less likeable, we think that an ordered sequence of items could be more useful to the user in some specific domain. While a simple top-n list is generated from a single starting point, which is usually the user model at a time t, a well-crafted sequence will have a starting point, an ending point and a meaningful path to go from the former to the latter. In certain domain like e-education, recommending a progressive enough sequence (i.e. a sequence of courses progressing through skill level neither too slow nor too fast) is a key element to guarantee the user a good learning experience. One of the key element to a sequence is the progression between each item: a too slow progression and the user will stagnate, a too fast progression and she/he could be lost or feel that there is no progressiveness at all in the recommended list of items. In the context of cultural heritage, the progressiveness can be linked to the distance to go in a museum and/or to the semantic links between items so as to tell a coherent story through the recommended tour.

Progressiveness colony - The concept of progressiveness is tied up to our goal of finding a good sequence of items. Contrary to the majority of recommender systems, we not only want to provide a list of items but a sequence having a beginning, an end and where each item makes sense one to another. As stated before, in a museum, a good recommended sequence could consist of exhibits related in one way or another (i.e. from the same artist or from the same period) and located close to each other. Such a sequence could then slowly progress to another artist or a different section of the museum. Thus our goal is to recommend a sequence that is progressing neither too slow nor too fast: we call this the optimal progressiveness o. This metric is used to ensure that the agents of the progressiveness colony find a tour where each node offer a good progressiveness in relation to its predecessor and successor nodes. To do so, we computed o based on the first and the last items of a given sequence for a specific user.

$$o_n = \frac{c_{z_n} - c_{a_n}}{s} \tag{7}$$

where o_n is the optimal progressiveness value of the nth characteristic of a specific sequence, c_{z_n} is the nth characteristic of the last item z of the sequence, c_{z_n} is the same for the first item of the sequence and s is the number of items in the sequence. The Equation 7 will be applied for the m item characteristics, resulting in a m-sized vector of optimal progressiveness values $(o_1, o_2, \cdots, o_m)$ for the specific sequence. This vector will then be used in the calculation of the distance of an edge (i, j) for the progressiveness colony. This method implies to have numerical characteristics to apply the equation, like for example the tempo of a music track or the date of a painting.

The optimal progressiveness o value is then used to compute the distance d between two nodes of the graph.

$$d_{ij} = \frac{\sum_n^m w_{o_n} \cdot \frac{c_{jn} - c_{in}}{o_n}}{\sum_n^m w_{o_n}} \tag{8}$$

where m is the number of item's characteristics considered, o_n is the optimal progressiveness value of the nth characteristic for the sequence, w_{o_n} is the weight of the nth characteristic for the user and depending on his preferences and c_{j_n} is the value of the nth characteristic of the item j.

3.3 Merging tactics

In the previous section, we described five objectives that could provide suitable recommendations for users. Each of these five objectives is associated to a specific ant colony in our model. Thus, after this step, we are left with as many lists of recommendations as colonies, where each one should represent a part of the final recommended list. In order to build it, we needed a tactic to merge all the colonies' lists into one. To do so, we propose two techniques described below.

Merging colony - The first merging tactic relies once more on the ACS algorithm but with one additional colony that we called "merging colony". Starting from the set of items found by the other colonies (**step 1**), the merging colony considers all the objectives at once with a weighted sum to calculate the distances on the graph's edges (**step 2**), as shown in Equation 9.

$$d_{ij} = \sum_{col \in colonies} w(col) \cdot d_{ij}(col) \tag{9}$$

where $w(col)$ is the weight representing the expected importance of the colony's objective in the final recommendation. To estimate those weights, we calculated the average values of each objective (similarity, diversity, novelty and preferences) on the last n sessions processed of the user. This gave us the general importance of each objective while taking into account contextual information and recent tendencies in the user's behavior. We also built a new graph for this path of the algorithm. To construct it, we used all the items in the lists found by the other colonies as vertices, we added edges to each consecutive pair of items in the lists and finally we added random edges in the same way that is described in the last part of the Equation 1 to give the possibility of new paths to be found and chosen by the merging colony's agents.

Lists merging - For the second merging tactic, we calculated the weight $w(col)$ of each objective in the same way that for the merging colony (see above). We then built the list step by step by considering all the items found by the different colonies. We iterated through all the available items for each step of the list construction and we added to the final recommended list the item which yield the best amelioration towards the expected values. This process was stopped either when the remaining items degraded the list's metrics, when there was no items left or when the last item of the initial listened session was found.

4 EXPERIMENTS IN DIFFERENT APPLICATION DOMAINS

4.1 Music

Our first evaluation of AntRS was done in the music domain due to the ease of getting data [28]. We will not discuss the results obtained in this study which showed the legitimacy of our multi-objective approach represented by separate colonies, thus we refer the reader to our last article [28]. In this paper, we will present the results obtained with two major differences in comparison to our previous study: we changed the way users are selected to build the graph in the training set so as to validate the robustness of our model, and we built a new graph for each user rather than having the same graph for all users to improve performances. We used the same dataset as previously from Deezer with 3,561 unique users, 178,910 unique songs, and 1,871,919 listenings from which we randomly chose 500 users to work with. But first, let us describe how we proceed to evaluate our model.

We compared the performances of AntRS with four state-of-the-art algorithms capable of producing lists of items in the same conditions than our model. The first three are classical techniques spanning most of the work in the recommender systems domain: UserKNN [4], TrustMF [44], and SVD++ [20]. Those three algorithms were implemented using the Java library librec [15]. We also implemented a fourth hybrid multi-objective model named PEH described in [31] to be able to compare AntRS to a state-of-the-art multi-objective recommender system. We used the three algorithms described above in the hybridization process. In the first study, we performed all the tests on a cross-validation dataset with a training set of 400 users and an evaluation set of 100 users each time. The goal was to test our model in a situation where it did not know anything about the new user. We obtained good results, showing that our model was indeed capable of producing recommendations to new users. For this paper, we used an approach were 80% of the user sessions were in the training set and the last 20% were the evaluation set with the same cross-validation method. The goal was to prove that our model is also capable of producing good results in a more classical setting used by the majority of the domain studies. Either way, for each listening session composed of 5 items or more in the test base, a recommended session was produced. The users of the training base had listened to 10,621 sessions while there were 2,569 sessions in the test base.

We used different metrics capturing all the aspects of what we consider a good recommendation: Precision, Recall and F-measure [2], Similarity and Intra-list Similarity [47], Diversity and Relative diversity [26, 38]. We also used the preferences and the novelty metrics of Equations 5 and 6 as well as in [38].

As explained in Section 3, we also proposed and tested two merging tactics to combine the results of our four objectives. We also tested to run the merging colony alone, without the **step 1** in the first merging tactic of Section 3.3: in that scenario, the merging colony operates on the whole graph, rather than on the subgraph of items recommended by the four colonies.

We also tested a new version of our model where, instead of building a single graph for all the users (AntRS single-graph) we built one graph per user based on the *n* nearest neighbors, as explained in section 3.1 (AntRS multi-graph).

In Table 1, we present the summary of the results obtained for all the models tested and their variations. In our test, we used only the four first objectives colonies (similarity, diversity, novelty, preferences) without the progressiveness. We will explain why at the end of this section.

From the results, we can see that both AntRS variations obtain the best precision, recall and F-measure of all the models tested, which means that our model is the best to capture the preferences from the lists initially listened by the users in the training set. AntRS also outperformed the other models for the diversity, novelty and preferences metrics, while still managing to maintain a correct level of similarity. We can also see that AntRS multi-graph manages to obtain almost the same performance as AntRS single-graph. We explain this small disparity by our selection of the *n* neighbor which could be improved along with the number of users for this experience (i.e. if the single graph was built with a lot more users than 500 then a selection process for the users could be useful).

Finally, let us talk about the progressiveness colony. As we stated earlier, we did not include it in the experiment for two main reasons. First, we ran some tests beforehand and the results were not conclusive. The colony obtained a very poor accuracy without building apparent progressive sequences. To understand why, we looked at the original music sessions that the users listened to. With the help of the Mann-Kendall test, which is designed to detect monotonic upward on downward trend of a variable over time, we found that there was no monotonic trend in the sessions. These results match with the recent research done in music showing that the order of songs is of little importance to users in playlists and music radios [33, 36].

4.2 Museums

After the music experimentation and to prove that our model is generic, we now want to apply it to a new domain: cultural heritage. This is a sensible choice as recommendation in tourism is booming since the 2000s and we wanted to work in a partnership with a museum in our city. Furthermore, the cultural heritage brings a lot of new challenges that are specific to the physical space represented by a museum, like for example managing the crowd in the museum, taking into account the physical space of the museum, taking into account the state of the visitor (young, old, tired, in a hurry, …). In this section, we will describe the museum, the experience we are currently doing, the data we are collecting and what we plan to do with these data.

The museum - We are currently working with the museum of fine arts[1] in Nancy, France. This museum is well-known for its fifteen century to today collection of paintings, statues and modern art works. It has close to 130,000 visitors each year which are welcomed in a 9.000m^2 classified historic building. The museum permanently presents about 400 works of art, without counting those stored in the archives. We currently have a partnership with the museum where we have access to their internal resources (information on pieces of art like artist, date of creation, title and so on). The museum also granted us a full and free entry for us and the experimentation's subjects.

Experimentation - As we said earlier, an experimentation is ongoing in the museum of fine arts. We are currently recruiting people to visit the museum. To analyze their route, we considered three possible solutions: (1) a manual tracking with someone following the visitor and noting his positions, like in [39], (2) an indoor tracking device with the help of a smartphone application or RFID/bluetooth technologies, as explained in [10], and (3) an eye-tracking device. We decided not use the first solution as it necessitated a lot of work without giving a better accuracy in the data. We also did not choose the indoor tracking system because

[1]https://musee-des-beaux-arts.nancy.fr/accueil-145.html

Table 1: Experimentations with AntRS as a multi-objective model

	Precision	Recall	F-measure	Similarity	ILS	Diversity	RD	Novelty	Preferences	Coverage
AntRS single-graph (4 colonies + lists merging)	**0.379**	0.147	0.205	0.942	0.902	0.058	0.073	**0.748**	0.631	**71.68%**
AntRS single-graph (4 colonies + merging colony)	0.339	**0.176**	**0.215**	0.943	0.9	0.057	**0.085**	0.72	**0.718**	**71.68%**
AntRS multi-graph (4 colonies + lists merging)	0.369	0.149	0.203	0.941	0.901	**0.059**	0.075	0.737	0.633	**71.68%**
AntRS multi-graph (4 colonies + merging colony)	0.329	0.167	0.206	0.944	0.905	0.056	0.081	0.707	0.68	**71.68%**
AntRSv1 (merging colony alone)	0.195	0.121	0.142	**0.952**	**0.931**	0.048	0.068	0.307	0.39	**71.68%**
UserKNN	0.213	0.146	0.165	0.947	0.902	0.053	**0.085**	0.668	0.509	45.5%
TrustMF	0.214	0.148	0.166	0.947	0.903	0.053	**0.085**	0.667	0.512	45.48%
SVD++	0.212	0.147	0.165	0.947	0.903	0.053	0.084	0.667	0.51	45.47%
PEH	0.196	0.121	0.143	**0.952**	0.93	0.048	0.069	0.308	0.391	**71.68%**

we did some internal testing with smartphone tracking as well as bluetooth beacons and the results were not good enough for our needs. We could not get those devices to have a better precision resolution than a few meters, making them suitable for tracking people in a whole room but not sufficient to precisely know where a visitor is in the room and what she/he is looking at. Thus we chose to use the eye-tracker with the latest version of the Tobii Pro Glasses 2[2]. These glasses allow us to let the visitor relatively free: we do not have to follow him to write down his movements and we do not have to ask him to interact with a tracking device either. The glasses are still a minor inconvenience to wear but it was the best compromise available between tracking precision and visit disruption.

Data - In this experimentation, we have two types of data: museum data and visitors collected data. The museum data refers to all the information relative to the museum and to the works of art. We measured the the museum rooms and we also recorded the exact positions of all the works of art. The goal is to be able to recreate the museum virtually and, at least, have a good idea of where the visitor is and what she/he is looking at exactly at each time during his visit. The visitors' collected data refers to the information we are gathering with the experimentation itself. The Tobii Pro Glasses 2 record a video flow of what the visitor has in front of him while, at the same time, compute the fixation points (i.e. what the person is looking at exactly). As we have the video information of where the user is, what she/he is looking at and for how long, we plan to build a dataset where those information will be transcribed as numerical values.

Goals - The goals of this experimentation are multiple. First, we will build a dataset based on the user data collected during the experimentation with the eye-tracker. We aim at providing the community a clean dataset with the information about the museum space, the works of art and also the precise path taken by all the visitors with timestamp on each movements and stops. With this dataset, we will be able to do offline experiments with our model AntRS to confirm that it is working in another domain. We will be to compute similarity and diversity between pair of exhibits based on their characteristics (artist, period, style, popularity, subject portrayed through the work...), but also based on the co-viewing of exhibits (i.e. we can consider to artworks similar if they are viewed by the same users through their visit). We will also be able to integer the progressiveness of sequences, which was not possible with music. Finally, another area of interest would be to simulate the museum in virtual reality with crowd behavior based on real visitors to try to adapt recommendations to users given their crowd tolerance [24].

[2]https://www.tobiipro.com/product-listing/tobii-pro-glasses-2/

5 CONCLUSION

In this paper, we showed that AntRS is still obtaining better results than state-of-the-art methods despite switching to a more traditional k-fold method. We also showed that, while not obtaining the very best results, our new approach consisting of creating a graph per user is promising and will probably be the way to go with a larger dataset and some variables tuning. Finally, we discussed about our ongoing experimentation in the museum of fine arts in Nancy, France. This experimentation will allow us to test our model on another application domain to guarantee its generic nature and to test our fifth objective related to progressiveness.

ACKNOWLEDGMENT

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 693150.

REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. 2011. Context-aware recommender systems. In *Recommender systems handbook*. Springer, 217–253.

[2] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. 1999. *Modern information retrieval*. Vol. 463. ACM press New York.

[3] Geoffray Bonnin and Dietmar Jannach. 2014. Automated Generation of Music Playlists: Survey and Experiments. *ACM Comput. Surv.* 47, 2 (2014), 26:1–26:35.

[4] John S Breese, David Heckerman, and Carl Kadie. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 43–52.

[5] Sylvain Castagnos and Anne Boyer. 2006. A client/server user-based collaborative filtering algorithm: Model and implementation. In *17th European Conference on Artificial Intelligence (ECAI 2006)*. 617–621.

[6] Keith Cheverst, Nigel Davies, Keith Mitchell, Adrian Friday, and Christos Efstratiou. 2000. Developing a context-aware electronic tourist guide: some issues and experiences. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, 17–24.

[7] Shih-Chun Chou, Wen-Tai Hsieh, Fabien L Gandon, and Norman M Sadeh. 2005. Semantic web technologies for context-aware museum tour guide applications. In *Advanced Information Networking and Applications, 2005. AINA 2005. 19th International Conference on*, Vol. 2. IEEE, 709–714.

[8] Gabriel Deak, Kevin Curran, and Joan Condell. 2012. A survey of active and passive indoor localisation systems. *Computer Communications* 35, 16 (2012), 1939–1954.

[9] Marco Dorigo and Mauro Birattari. 2011. Ant colony optimization. In *Encyclopedia of machine learning*. Springer, 36–39.

[10] Zahid Farid, Rosdiadee Nordin, and Mahamod Ismail. 2013. Recent advances in wireless indoor localization techniques and system. *Journal of Computer Networks and Communications* 2013 (2013).

[11] Reinaldo Silva Fortes, Anisio Lacerda, Alan Freitas, Carlos Bruckner, Dayanne Coelho, and Marcos Gonçalves. 2018. User-Oriented Objective Prioritization for Meta-Featured Multi-Objective Recommender Systems. In *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization*. ACM, 311–316.

[12] Luca Maria Gambardella, Éric Taillard, and Giovanni Agazzi. 1999. *MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows*. Technical Report. Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale.

[13] Bingrui Geng, Lingling Li, Licheng Jiao, Maoguo Gong, Qing Cai, and Yue Wu. 2015. NNIA-RS: A multi-objective optimization based recommender system. *Physica A: Statistical Mechanics and its Applications* 424 (2015), 383–397.

[14] Adolfo Guimarães, Thales F. Costa, Anisio Lacerda, Gisele L. Pappa, and Nivio Ziviani. 2013. GUARD: A Genetic Unified Approach for Recommendation. *Journal of Information and Data Management* 4, " (2013), 295.

[15] Guibing Guo, Jie Zhang, Zhu Sun, and Neil Yorke-Smith. 2015. LibRec: A Java Library for Recommender Systems. In *UMAP Workshops*.

[16] Negar Hariri, Bamshad Mobasher, and Robin Burke. 2012. Context-aware Music Recommendation Based on Latenttopic Sequential Patterns. In *Proceedings of the Sixth ACM Conference on Recommender Systems (RecSys '12)*. 131–138.

[17] Dietmar Jannach, Lukas Lerche, and Iman Kamehkhosh. 2015. Beyond hitting the hits: Generating coherent music playlist continuations with the right tracks. In *Proceedings of the 9th ACM Conference on Recommender Systems*. ACM, 187–194.

[18] Nicolas Jones. 2010. *User Perceived Qualities and Acceptance of Recommender Systems: The Role of Diversity*. Ph.D. Dissertation. EPFL.

[19] Marius Kaminskas and Derek Bridge. 2016. Diversity, Serendipity, Novelty, and Coverage: A Survey and Empirical Analysis of Beyond-Accuracy Objectives in Recommender Systems. *ACM Trans. Interact. Intell. Syst.* 7, 1 (2016), 2:1–2:42.

[20] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 426–434.

[21] Tsvi Kuflik, Zvi Boger, and Massimo Zancanaro. 2012. Analysis and Prediction of Museum Visitors' Behavioral Pattern Types. In *Ubiquitous Display Environments*. Springer, 161–176.

[22] Justin J Levandoski, Mohamed Sarwat, Ahmed Eldawy, and Mohamed F Mokbel. 2012. Lars: A location-aware recommender system. In *2012 IEEE 28th International Conference on Data Engineering*. IEEE, 450–461.

[23] Amaury L'Huillier, Sylvain Castagnos, and Anne Boyer. 2014. Understanding Usages by Modeling Diversity over Time. In *22nd Conference on User Modeling, Adaptation, and Personalization*, Vol. 1181.

[24] Ioanna Lykourentzou, Xavier Claude, Yannick Naudet, Eric Tobias, Angeliki Antoniou, George Lepouras, and Costas Vassilakis. 2013. Improving museum visitors' Quality of Experience through intelligent recommendations: A visiting style-based approach.. In *Intelligent Environments (Workshops)*. 507–518.

[25] François Maillet, Douglas Eck, Guillaume Desjardins, and Paul Lamere. 2009. Steerable playlist generation by learning song similarity from radio station playlists. In *In Proceedings of the 10th International Conference on Music Information Retrieval*.

[26] Lorraine McGinty and Barry Smyth. 2003. On the Role of Diversity in Conversational Recommender Systems. In *Proceedings of the 5th International Conference on Case-based Reasoning: Research and Development (ICCBR'03)*. 276–290.

[27] Yannick Naudet and Ioanna Lykourentzou. 2014. Personalisation in crowd systems. In *Semantic and Social Media Adaptation and Personalization (SMAP), 2014 9th International Workshop on*. IEEE, 32–37.

[28] Pierre-Edouard Osche, Sylvain Castagnos, and Anne Boyer. 2019. AntRS: Recommending Lists through a Multi-Objective Ant Colony System. In *European Conference on Information Retrieval*. Springer.

[29] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-Aware Recommender Systems. *CoRR* abs/1802.08452 (2018).

[30] Marco Tulio Ribeiro, Anisio Lacerda, Adriano Veloso, and Nivio Ziviani. 2012. Pareto-efficient Hybridization for Multi-objective Recommender Systems. In *Proceedings of the Sixth ACM Conference on Recommender Systems (RecSys '12)*. 19–26.

[31] Marco Tulio Ribeiro, Nivio Ziviani, Edleno Silva De Moura, Itamar Hata, Anisio Lacerda, and Adriano Veloso. 2014. Multiobjective Pareto-Efficient Approaches for Recommender Systems. *ACM Trans. Intell. Syst. Technol.* 5, 4 (2014), 53:1–53:20.

[32] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2015. *Recommender Systems Handbook*. Springer, US.

[33] Markus Schedl, Hamed Zamani, Ching-Wei Chen, Yashar Deldjoo, and Mehdi Elahi. 2018. Current challenges and visions in music recommender systems research. *International Journal of Multimedia Information Retrieval* 7, 2 (2018), 95–116.

[34] Yann Semet, Evelyne Lutton, and Pierre Collet. 2003. Ant colony optimisation for e-learning: Observing the emergence of pedagogic suggestions. In *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE*. IEEE, 46–52.

[35] Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence* 2009 (2009), 4.

[36] Nava Tintarev, Christoph Lofi, and Cynthia C.S. Liem. 2017. Sequences of Diverse Song Recommendations: An Exploratory Study in a Commercial System. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization (UMAP '17)*. 391–392.

[37] Willem van Hage, Natalia Stash, Yiwen Wang, and Lora Aroyo. 2010. Finding your way through the rijksmuseum with an adaptive mobile museum guide. *The Semantic Web: Research and Applications* (2010), 46–59.

[38] Saúl Vargas and Pablo Castells. 2011. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems*. ACM, 109–116.

[39] Eliseo Véron and Martine Levasseur. 1989. *Ethnographie de l'exposition: l'espace, le corps et le sens*. Centre Georges Pompidou, Bibliothèque publique d'information.

[40] Shanfeng Wang, Maoguo Gong, Haoliang Li, and Junwei Yang. 2016. Multi-objective optimization for long tail recommendation. *Knowledge-Based Systems* 104 (2016), 145 – 155.

[41] S. Wang, M. Gong, L. Ma, Q. Cai, and L. Jiao. 2014. Decomposition based multiobjective evolutionary algorithm for collaborative filtering recommender systems. In *2014 IEEE Congress on Evolutionary Computation (CEC)*. 672–679.

[42] Wolfgang Woerndl, Johannes Huebner, Roland Bader, and Daniel Gallego-Vico. 2011. A model for proactivity in mobile, context-aware recommender systems. In *Proceedings of the fifth ACM conference on Recommender systems*. ACM, 273–276.

[43] Xiao Xia, Xiaodong Wang, Jian Li, and Xingming Zhou. 2014. Multi-objective Mobile App Recommendation: A System-level Collaboration Approach. *Comput. Electr. Eng.* 40, 1 (2014), 203–215.

[44] Bo Yang, Yu Lei, Jiming Liu, and Wenjie Li. 2017. Social collaborative filtering by trust. *IEEE transactions on pattern analysis and machine intelligence* 39, 8 (2017), 1633–1647.

[45] Yuji Yoshimura, Stanislav Sobolevsky, Carlo Ratti, Fabien Girardin, Juan Pablo Carrascal, Josep Blat, and Roberta Sinatra. 2014. An analysis of visitors' behavior in The Louvre Museum: a study using Bluetooth data. *Environment and Planning B: Planning and Design* 41, 6 (2014), 1113–1131.

[46] Liang Zhang. 2013. The Definition of Novelty in Recommendation System. *Journal of Engineering Science & Technology Review* 6, 3 (2013).

[47] Cai-Nicolas Ziegler, Sean M McNee, Joseph A Konstan, and Georg Lausen. 2005. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web*. ACM, 22–32.

[48] Y. Zuo, M. Gong, J. Zeng, L. Ma, and L. Jiao. 2015. Personalized Recommendation Based on Evolutionary Multi-Objective Optimization [Research Frontier]. *IEEE Computational Intelligence Magazine* 10, 1 (2015), 52–62.