Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Improving itinerary recommendations for tourists through metaheuristic algorithms: an optimization proposal

MARITZOL TENEMAZA[1, 2], SERGIO LUJÁN-MORA[3], ANGÉLICA DE ANTONIO[2], (Member, IEEE), and JAIME RAMÍREZ [2]

[1]Departamento de Informática y Ciencias de la Computación, Escuela Politécnica Nacional, Quito - Ecuador (e-mail: maritzol.tenemaza@epn.edu.ec)
[2]Escuela Técnica Superior de Ingenieros Informáticos, Universidad Politécnica de Madrid, Madrid - Spain(e-mail: rm.tenemaza@alumnos.upm.es, angelica;jramirez@fi.upm.es)
[3]Departamento de Lenguajes y Sistemas Informáticos, Universidad de Alicante, Alicante - Spain (e-mail: sergio.lujan@ua.es)

Corresponding author: Maritzol Tenemaza (e-mail: maritzol.tenemaza@ epn.edu.ec; rm.tenemaza@alumnos.upm.es).

ABSTRACT In recent years, recommender systems have been used as a solution to support tourists with recommendations oriented to maximize the entertainment value of visiting a tourist destination. However, this is not an easy task because many aspects need to be considered to make realistic recommendations: the context of a tourist destination visited, lack of updated information about points of interest, transport information, weather forecast, etc. The recommendations concerning a tourist destination must be linked to the interests and constraints of the tourist. In this research, we present a mobile recommender system based on Tourist Trip Design Problem (TTDP)/Time Depending (TD) – Orienteering Problem (OP) – Time Windows (TW), which analyzes in real time the user's constraints and the points of interest's constraints. For solving TTDP, we clustered preferences depending on the number of days that a tourist will visit a tourist destination using a k-means algorithm. Then, with a genetic algorithm (GA), we optimize the proposed itineraries to tourists for facilitating the organization of their visits. We also used a parametrized fitness function to include any element of the context to generate an optimized recommendation. Our recommender is different from others because it is scalable and adaptable to environmental changes and users' interests, and it offers real-time recommendations. To test our recommender, we developed an application that uses our algorithm. Finally, 131 tourists used this recommender system and an analysis of users' perceptions was developed. Metrics were also used to detect the percentage of precision, in order to determine the degree of accuracy of the recommender system. This study has implications for researchers interested in developing software to recommend the best itinerary for tourists with constraint controls with regard to the optimized itineraries.

INDEX TERMS Genetic algorithm, heuristic algorithm, k-means, Tourist Trip Design Problem, recommender system.

I. INTRODUCTION

TOURISM is a worldwide industry that involves the propagation of large amounts of information [1]. All over the world, tourist destinations offer many interesting attractions and places for travelers and tourists. Since each visitor has different interests when visiting a destination (e.g., adventure, shopping, cultural/historical, most important points of interest), it is impossible to tie their interests to a unique itinerary for the visit. Indeed, before visiting a place, tourists tend to prepare a trip plan; one that responds adequately to their own interests and time constraints. These requirements limit the range of local attractions they can visit. According to their available time, tourists select what they consider to be their Points of Interest (POIs).

To recommend an itinerary, it is very important to have accurate up-to-date information about the POIs in each tourist destination. For this, there are two options: (1) preparing a database including all the information about all the possible POIs of each destination [2], or (2) leveraging the potential of data already available on the Internet to recover the information about POIs and produce recommendations that consider the changing environment of these tourist destinations [2] [15] [21]. The problem with the first solution lies in the huge amount of work required to prepare the database and the static nature of of such data. For Padrón-Ávila et al. [2], the database can be enriched from users information, through surveys, advertising, sales, and details of specific spots. Indeed, tourist destinations and POIs are not static; they evolve over time and to remain up-to-date with regard to such changes needs continuous supervision of the system. The second option seems to be the most suitable solution; for Padrón-Ávila et al. [2], the database can be enriched through the use of web analysis [3] and Geolocation. Nonetheless, this solution presents a challenge since it is necessary to process a large volume of data (especially for large cities) to produce good recommendations in the case of tourists. In our proposal, we use the second solution in terms of the use of POIs on the Internet and web analysis.

The main benefit for researchers associated with this recommendation system is to be aware that the best itineraries can be derived from an analysis of the restrictions specified both by the users and by POIs and that they can be analyzed with regard to a specific place described in a recommendation system. We propose a parametrized fitness function where each constraint is analyzed. Indeed, choosing a suitable customizable method is important, as it determines the outcomes of the research. Likewise, we include a complete proposal with regard to the recommendation system; our proposal includes all the processes needed to provide optimized itineraries. This proposal can also help researchers and managers of tourist information centers to organize the itineraries which incorporate user' personalization.

To test the validity of our proposal, we apply a survey addressed to 131 tourists that were asked about our recommender system. Especially, we focused our interest on the quality of the proposed POIs while the tourist visited the city. The second part of the evaluation aimed to understand if the recommended POIs were also relevants to the tourist or not.

The remainder of the article is organized as follows: in section 2, we present the theoretical background. In section 3, we discuss related works, presented in chronological order. In section 4, we describe the method employed for developing our recommender system. In section 5, an example of execution is presented. Section 6 explains the evaluation of our recommender system with real users. In section 7 a brief discussion is shown. Finally, in section 8, the conclusions and future work are set out.

II. THEORETICAL BACKGROUND

Our research proposes an algorithm to identify the best itinerary based on constraints provided by the tourist. Therefore, the first step is to acknowledge the source of the best places in each location in the light of these constraints. An identification of the tourist-tracking techniques implemented by Padrón-Ávila et al. [2] has contributed to understanding the different sources from which information; the processing of this data enables a reduction in the existing inequalities in an economic region in terms of tourism activities and improving the management with regard to specific places, cities, etc. For Padrón-Ávila et al. [2], this database identifies the places visited, the preferred tourist activities, and the degree of satisfaction of the tourists. The authors identify a variety of sources of information: (a) from tourists themselves, surveys, web analysis, geolocation, (b) from tourist companies, through advertising, sales, and specific spots. We conclude that an analysis of tourist satisfaction produces clear and, detailed advantages and disadvantages of particular places. For example, we can understand such aspects as accessibility to such places, and the time needed to visit specific places (the time used is relative to the user's interests). Additionally, this data collection could serve to better understand user' attitudes, their interests and their behavior.

TTDP refers to the itineraries-planning problem for tourists interested in visiting multiple POIs with the objective of maximizing tourist profit [4]. There are two types of TTDP: single tour TTDP and multiple tour TTDP. The single tour defines an itinerary from start to finish choosing the best POIs that maximize the collected profit. The multiple tour TTDP differs in defining multiple tours based upon the number of days of the tourist visit [5].

TTDP is based on the Traveling Salesman Problem with Profit (TSPP) and the Vehicle Routing Problem with Profit (VRPP) [5]. Both TSPP and VRPP seek to maximize the collected profit and minimize the travel cost [5]. A variant of TSPP is the Orienteering Problem (OP) and a variant of VRPP is the Team Orienteering Problem (TOP).

The variant OP considers single criteria, maximizes the total collected profit and minimizes the travel cost. Likewise, OP has two versions: Orienteering Problem Time Windows (OPTW) and Time Depending Orienteering Problem (TDOP). OPTW considers visits to locations within a predefined time window (this allows modeling of the opening and closing hours of POIs). TDOP considers time dependency in the estimation of time required to move from one location to another [5]. This time and space can be calculated when moving on foot or by motor vehicle. The combination of the two options is Time Depending Orienteering Problem Time Windows (TDOPTW); that is to say, time in POIs and time between POIs.

TTDP considers more user requirements and POI constraints. Indeed, it could take in account other constraints such as weather conditions, accessibility features of POIs, budget restrictions, etc. A realistic itinerary should therefore provide time for breaks, either for resting such as in a nearby park, or for a coffee and a meal (e.g., a meal should be scheduled around noon), budget constraints and user preferences. Also, a realistic TTDP should take into account multiple time windows (TW) and a selection of POIs of a maximum number of certain types of POIs per day [5].

III. RELATED WORK

Souffriau et al. [6] present a mobile recommender tourist guide based on TTDP/OP. The prediction of interest is based on the tourists' interest of the user obtained from a corpus. In contrast to our proposal, Souffriau et al. propose an itinerary in mobile tour planning based on TTDP/OP, whereas we propose a mobile or web tour planning for many days based on TTDP/TDOPTW. Additionally, their objective is to maximize the total score of the locations visited, while keeping the total time (or distance) below the available time. In contrast, in our work, we maximize the number of POIs to be visited, finding the best itinerary. We also consider visiting times to each place, dining facilities during the day and optimizing; in such a way, that the required times do not conflict with the periods of time for eating. The best aspect of our algorithm is that it is parameterized and, therefore, it is scalable.

Tsai and Chung [7] propose an itinerary recommender for a theme park. We can deduce that they apply TTDP/OP. The authors address the personalized POI selection and sequencing based on the experience of previous tourists. Their recommendation provides personalized visiting itineraries that consider a set of visiting constraints. The system works with an itinerary database and a visitor's personal preferences to offer an itinerary recommendation. The advantage of such a system is that it works in a well-known space. Therefore, the POIs database is comprehensive, because it is possible to include all the information about each attraction. For theme parks, the attendance period of the public is known; however, when it comes to recommenders for POIs to visit in a tourist destination, as in our case, the hours of attendance can be variable between days. The disadvantage of this system is that is can neither scale outside the limits nor be easily adapted to new environments or to variable environments.

Chen et al. [8] propose a solution that considers several traveling factors such as the budget and available time. Thus, their recommendation system refines an exact set of tourist locations by applying a GA based on minimum cost. Nonetheless, these authors also work with predefined tourist locations in a well-known environment. They predict interests based on other tourists' interests using collaborative filtering. They use a traditional GA approach to minimize the duration time of tourists' visits. They apply TTDP/OP. On the contrary, we differentiate ourselves in that we apply TTDP/OPTW. In each time window, the itinerary is optimized considering real POIs, each with its own restrictions. Our optimization respects the lunch schedule. We show that our algorithm, when parameterizing options, widens the range of restrictions that the user can specify.

Schaller and Elsweiler [9] use the data of three well-known events to customize their recommendations. These are the long night of music in May 2013, the long night of Munich museums and the long night of science. They use static POIs. Then, a tourist is able to choose between two tour recommendations: one that is based on the current user preferences by interviews and another lightly modified that iteratively leads to learning a user's preference model. They allow for editing of the itinerary and for the user to change the visit's duration to POI; however, they do not analyze the restrictions of the points.

Chiang and Huang [10] propose a web personalized travel planning system that considers users' requirements and provides them with a travel schedule planning service. Their framework uses a database that is renewed through a feedback mechanism that records a user's travel schedule and choices as a basis for future recommendations. The personalized travel planning system obtains the user's requirements and presents a travel plan by means of an adjustable interface. The users can modify their travel schedule planning until it satisfies their requirements. The authors consider a set of locations, restaurants and hotels, and a specific set of travel days. They do not mention TTDP specifically; however, in their proposal they present sites close to the user's location and choose the popular sites from their own database. This proposal differs from ours in that we include optimization processes and algorithms to detect and present points of greatest enjoyment but in an automated way. Our fitness algorithm helps to optimize the proposal.

Gavalas et al. [11], with eCOMPASS RS, propose a context aware web and mobile application that allows filtering of the most important POIs. To move between POIs, users can walk or use public transport. The authors apply TTDP/TDOPTW with a cluster-based heuristic approach for deriving daily tourist tours that match tourists' preferences. In this study, as observed, the restaurants are all along the itinerary, thus there is the risk of generating itineraries with more restaurants than POIs. On the contrary, in such a way that the user's work on the application is reduced, we work with an analysis of a user's interests, detected by machine learning and by asking the user questions in an application. At the same time, Gavalas et al. compile locations in two cities; in contrast, we work with the real context. After clustering, they include their algorithms; after clustering, we propose to optimize using a GA with an improved fitness function.

Herzog [12] deals with TTDP by considering the preferences and constraints of tourist groups. They use information about members' preferences as well as those of other groups. Their approach is collective rather than individual, in a group recommender system. They test some TTDP algorithms to calculate the distance between points. In contrast, we propose a best fitness function to optimize the itinerary including any parameters.

Cenamor et al. [13] present the PLANTOUR system, which is based on human-generated information gathered from the MINUBE traveling social network and the user's preferences requested from the user. The system generates a tourist guide for multiple days. They use the clustering techniques to split the problem with a k-means algorithm. In this work, the authors use the information generated by other humans (scores stored in the social network).

They consider the TTDP algorithm with planning domain definition language to minimize the total itinerary traveled and maximize the user utility of visiting places. In contrast, in our recommender system, we combine two heuristics algorithms both to clusterize and as our best effort at optimization.

Zheng et al. [14] present a proposal for solving TTDP/OPTW by combining a GA and a difference evolution algorithm (DEA). They analyze both users (budget time available and tourists' preferences) and POIs. This new algorithm deals with itinerary coding, initial itinerary construction, itinerary set evolution and itinerary evaluation. The proposed algorithm differs from existing methods in three major ways: (1) it applies a double-layer, variable-length chromosome approach for coding the itinerary; (2) it uses an improved greedy algorithm to construct the initial itinerary set; and (3), as mentioned, it obtains the optimal itinerary by combining a GA with a DEA. However, the authors work with predefined POIs, and these sequences are not complete tours. Also, they do not include constraints such as duration or length of the tour.

Ahmadian et al. [15] use social relationships among users to find preferences, and to provide relevant suggestions to users through social recommendation systems. They propose a novel method for predicting reliable virtual ratings based on user recommendations and clustering models. Additionally, they include a method to reduce the amount of noise in the data. They assure as that the performance of the social recommendation system can be improved through incorporating reliable virtual ratings. Like Ahmadian et al., we consider that the identification of users' interests is a very important field in the area of recommendation system; these authors suggest interesting options in order to expand this area of research.

Ahmadian et al. [16] explain the construction of a recommendation system based on a temporal clustering approach. The researchers constructed a user' network using a combination of similarity values and trust relationships between the users. They develop a graph-based method to find the initial centroid set of the clusters. The final clusters are obtained using an iterated mechanism on the initial sentroid set. The researchers based their research on the data from social networks, with the use of two leading collaborative filtering approaches. These are interesting options that are proposed to identify the variable centroids of each cluster of POIs. The centroids vary over time (because interests can change over time), based on the "likes" and "dislikes" expressed by users. These are interesting options that have been proposed for identifying the variable centroids of each cluster of POIs. Our recommendation system carries out the optimization processes using genetic algorithms. However, this proposal is useful in the sense of opting for new options for the k-means algorithm.

Tarantino et al. [17] present an interactive electronic guide application prototype able to recommend personalized multiple-day tourist itineraries to mobile web users. This recommender system is based on users' preferences, especially a model used to adapt the tourist itinerary to user preferences and constraints. The researchers base their proposal on a TDOPTW using different constraints. The more important module is the evolutionary optimizer presented by the researchers, in which Tarantino et al. propose a user model to register the users' information. Conversely, we propose a complete model with analysis of the user [18] [19]. We have not included this analysis because we have focused on the optimization algorithm. Their optimizer uses GA and they do not explain how they organize the POIs for days, but rather they let the user select where the tour begins and ends. On the contrary, we independently organize the different itineraries and allow the user to add POIs through the application.

Rahmani et al. [20] propose an interesting POI recommendation method based on a local geographical model, which considers both users' and locations' points of view. In their model, they considered, from the users' point of view, the geographical information was modeled considering the users' geographic interest. On the other hand, in terms of the locations' point of view, the geographical information was modeled as the number of individuals accessing a selected POI. We offer an alternative to this proposal in that we capture our data from Google, which in turn evaluates the POIs from user data. This is the reason why we have tried our recommendation system some places or cities. The work carried out by Rahmani et al. gives us new criteria to consider in our work. We deduce that, despite the dispersion of data, they have analyzed user' interests in this geographical case. They apply a TTDP-OP. Conversely, we use a TTDP-OP-TW, that is, we add time periods to our recommendation system.

Ahmadian et al. [21] propose a method to enhance the rating profiles of the users who have low user reliability, measured by adding several virtual ratings. This approach is used to generate more reliable data in order to generate recommendations from a recommendation system. This type of study broadens the area of research on the topic of recommendation systems in order to reduce a possible lack and dispersion of data, and thus avoid the problem of the cold start of applications. We have considered this appraoch since it is a very interesting OP method, where the analysis of user data from social networks becomes the core of this broad field of research into recommendation systems.

Table 1 summarizes and compares the various recommender systems for tourists that we have analyzed. As it is deducted from our analysis, to build a touristic recommender, all researchers apply a TTDP. The difference in some cases is based on the algorithms used for clustering or in the use of GA or other algorithms. All investigators agree in their consideration of the users' interests by asking the users. Most of them optimize the distance and use a personalized database. In our case, we work with a public Google database, so POIs cannot be outdated.

Although research investigating how people interact with a recommender system is in progress, and researchers focus

TABLE 1. Comparison of recommender systems for tourist

	Users' interests	Others users' interests	Multiple days	Time budget	Algorithm	TTDP	Distance optimization	Predefined database
Souffriau et al. [6]	*			*	GLS	TTDP /OP	*	*
Tsai & Chung [7]		*			k-means	N/A	*	*
Chen et al. [8]		*	*	*	GA	TTDP /OP	*	*
Schaller & Elsweiler [9]	*				N/A	N/A	*	
Chiang & Huang [10]	*		*	*	N/A	N/A		
Gavalas et al. [11]	*		*		k-means	TTDP /TDOPTW	*	*
Herzog [12]	*				N/A	TTDP /TDOPTW	*	
Cenamor et al. [13]	*	*	*		k-means	TTDP /OP		*
Zheng et al. [14]	*			*	GA	TDOPTW	*	*
Ahmadian et al. [15]	*	*			k-means	OP		*
Ahmadian et al. [16]	*	*			k-means	OP		*
Ahmadian et al. [21]	*	*			other method	OP		*
Rahmani et al. [20]	*	*			other method	TTDP-OP		*
Tarantino et al. [17]	*		*	*	GA	TDOPTW	*	*
This research	*	*	*	*	k-means GA	TTDP /TDOPTW	*	

on specific algorithm proposals. Until now, this research has not yet been performed to examine how to emphasize specific elements of software construction. Therefore, we provide a fully parametrized algorithm, which is included in an optimization heuristic by a GA.

This paper develops an itinerary recommendation system, discovering the best itinerary for optimizing visits to as much POIs as possible respecting the tourist's available time. For that, we consider multiple-day planning without violating tourists' preferences. We provide a better tourist experience by not only obtaining the best POIs and itineraries, but also by optimizing them by considering contextual parameters and constraints of POIs and updated information about them. We combine two heuristic algorithms: (a) a k-means algorithm for clustering all the POIs related to the users' interests depending on the number of available days; (b) a GA algorithm for optimizing the itinerary of POIs. Our GA includes personalized functions, which represents a new algorithm for solving TTDP/TDOPTW. All these improvements are detailed in the following sections.

IV. METHOD

A. DESIGN OF THE RESEARCH

For the development of this work, a research approach based on Design Science Research [22] [23] was applied over multiple case studies [24]. This is fundamentally a paradigm for problem-solving that allows innovation by redefining ideas, practices, technical capacities and products that enable all the tasks of analysis, design, implementation, administration and use of Information Systems to be performed more effectively and efficiently [25] [26]. The principle of research in Design Science Research is that both the knowledge and the understanding of a design problem, as well as the solution of the problem itself, are acquired through the construction of an artefact [22] [27] [28]. In our case, the resulting artefact is a recommender system of the best POIs that provides recommendations according to a user's interests. The recommender system considers the limitations in available days for the visit, considers the actual constraints of POIs, and includes the transfer time between POIs. In this sense, the recommender developed constitutes a valid and useful practical construction to understand the different aspects involved in the resolution of the TTDP [29].

B. GENERAL ALGORITHM

The itinerary recommendation system consists of four major modules as shown in Fig. 1: module I is the user's *previously identified interests*; module II is *other users' detected interests based on POIs popularity* [3]. This paper develops module III (k-means algorithm) and module IV (GA algorithm).

- **Previously identified user's interests**: module I in Fig. 1. The objective of our approach based on learning machine is to avoid the "cold start", that is, our recommender has data to make initial recommendations. The architecture was based on the Bluemix[1] of Watson Platform as a service (PAAS). The Natural Language Classifier (NLC) server was our training machine that allows natural language processing. This server uses Elasticsearch to classify and identify the words that represent the user's interests. In a Decision Optimization Server (DOS) we use the Trade Off Analytics Service to determine the initial POIs to start our recommender; with these POIs our recommender will avoid a "cold start". To obtain the user's interests, it is necessary to explicitly obtain the user's authorization.
  Due to the difficulty of obtaining data inherent to the user, the necessary data were obtained from surveys of 600 users, of which 80 % of the answers were used for training and 20 % for tests. In addition, a list of categories to be used was defined. For example,

[1] https://www.ibm.com/cloud-computing/bluemix/node/3451

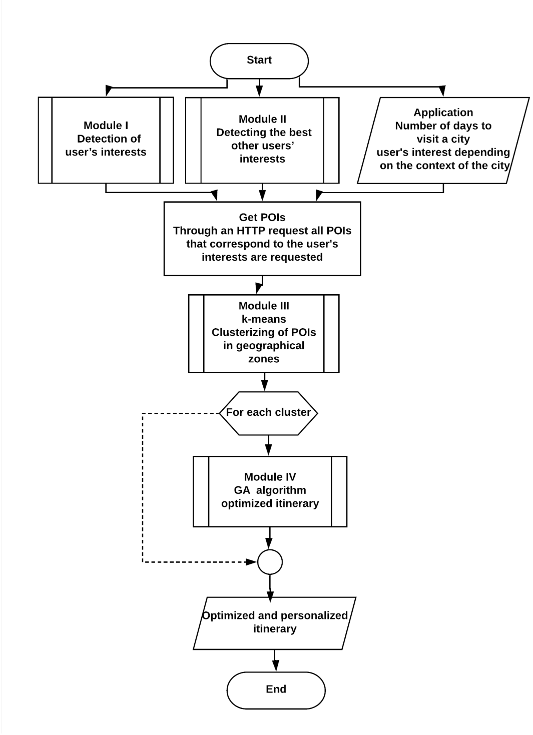


FIGURE 1. General Logic. Modules to build a recommender.

museums, restaurants, hotels, entertainment centers, etc. This list can be as large as the one specified by Google[2].

- **Other users' interests are detected based on POIs popularity** were analyzed using big data techniques [3], as shown in Module II in Fig. 1. In order to avoid a "cold start" of the recommender, when no user data is available, the first recommendation can be obtained from the sites identified by other users. The objective of this architecture was to identify the most-visited places through a sentiment analysis of the tweets posted by people who visited a specific region of a tourist destination. The data analyzed were related to preferences and opinions about tourist places. The Twitter API was implemented on a virtual machine. Elasticsearch, Kibana, and Cerebro were used as servers. Additionally, Python scripts were necessary to apply the harvesting architecture.
- The **k-means algorithm** was developed in module III in Fig. 1. Next, based on the users' interests, the first group of POIs was generated. In this work, we applied a heuristic modified k-means algorithm to clusterize all POIs suitable to the user's interests.
- The **GA algorithm** is shown in module IV in Fig. 1. Finally, we executed two types of optimization based on a GA: (a) the tourist itinerary, and (b) the best POIs based on realistic user's interests and POI constraints. The GA fitness and crossover functions are enhanced to determine the best points of interest in an itinerary of a tourist destination according to the user's interests, maximizing tourist profit. For this, we ensured that in the time available, tourists can visit the best sites for which we optimize the itinerary and the order to visit, we analyze the opening and closing times of the places to visit, we optimize the lunch time of such, so that there is no overlap of times, and our algorithm is left open so that it is possible to include other conditions.

C. K-MEANS ALGORITHM

In the proposed method, the number of clusters in the k-means algorithm is set to the number of days that the user can visit a place. We use this mechanism because (1) our recommender offers a different itinerary for each day of visit to a city or place; (2) we evaluated the risk of working without a previously established data set; we take the best POIs evaluated by Google (from where the best POIs are filtered considering the interests of the user and the context of the city to visit) and web analysis (to obtain the interests of other users) which meant that the number of POIs could be very high; (3) k-means does not consider inherent restrictions to each POI, but rather it considers geographical distances calculated based on a Euclidean distance metric.

Like other clustering algorithms, k-means uses an iterative procedure where each iteration tries to minimize the sum of the Euclidean distances between the elements of each cluster in order to maximize the similarity between them. The sum of the Euclidean distances between the elements of each cluster is observed in equation (1) [30] [31]. The distances are calculated with respect to an axis or centroid *u* fixed within each cluster **c**. It is necessary to include the initialization of *k* centroids where *k* represents the number of days that the user can visit a tourist destination. Each cluster can be sized for any number of POIs. Next, the POIs nearest to each centroid are assigned. The centroid values are adjusted and then modified until the best clusters are generated with the best possible centroid. Finally, the best groups of POIs for a geographical sector are presented.

$$d = \sum_{j=1}^{k} \sum_{\forall x_i \epsilon c_j} \|x_i - u_j\|^2 \tag{1}$$

d=Euclidean distances
$x_1, x_2, ...x_n$ = x is the POI, there are *n* POIs in *k* clusters
$x_i(1 \leq i \geq n)$ = n POIs
k = *k* clusters, Number of clusters = number of days to travel
$c_1, c_2, ...c_k$ = *c* is the cluster set
$u(u_1, u_2, ...u_k)$ = *u* is the centroid in each cluster
$cj(1 \leq j \geq k)$ = *k* is the number of centroids. There are as many centroids as clusters
$uj(1 \leq j \geq k)$

Once the interests have been detected, a set of POIs are generated, $POIs = POI_1, POI_2, ..., POI_n$ where *n* is the number of possible POIs to visit in a tourist destination. These POIs represent the tourist's interests. The module III (see Fig. 1) includes a k-means algorithm for clustering

[2] https://cloud.google.com/maps-platform/places/?hl=es&sign=0

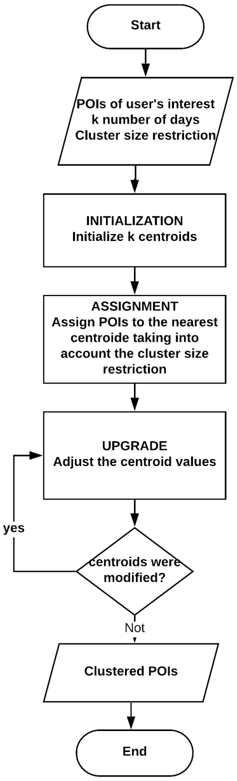


FIGURE 2. k-means algorithm.

and thereby ensuring the matching of the user's interest with POIs according to their geographical distribution. And the module IV develop the GA algorithm to optimize the itineraries proposal for the tourist visit. The logic of the k-means algorithm is shown in Fig. 2.

We codify the solution by means of permutations of numbers to describe the order of visits to the different POIs. Fig. 3 and Fig. 4 exhibit some results from the application of the k-means algorithm for clustering. Whereas Fig. 3 shows all the POIs for a specific location, Quito-Ecuador in this example, Fig. 4 shows the POIs after clustering. Fig. 4 shows points in three colors that represent three days of visits. The grouping of points in a color symbolizes the possible POIs to visit in a day and each color means a cluster with a set of POIs related to the user's interests.

D. GENETIC ALGORITHM

A fourth module, based on the GA, considered the metaheuristic search used in combinatorial optimization problems. This is based on the mechanics of natural selection and genetics. It combines the survival of the fittest among structures with a structured and random exchange of information [32] [33]. In this research, the benefits of GA were used to optimize travel time in tourist itineraries. Our GA module for the itinerary optimization considered the POIs' constraints and the user's constraints. The logic of the GA algorithm is shown in Fig. 5.

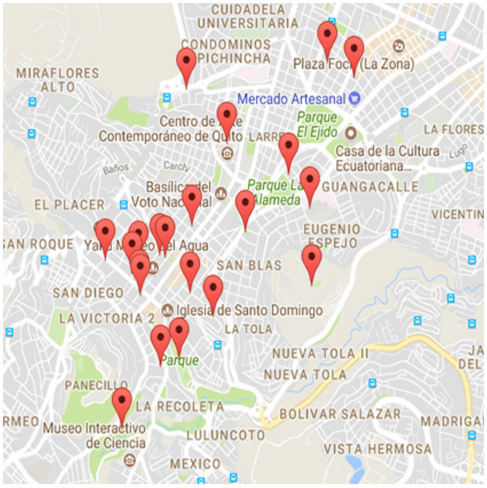


FIGURE 3. POIs for a geographical sector.

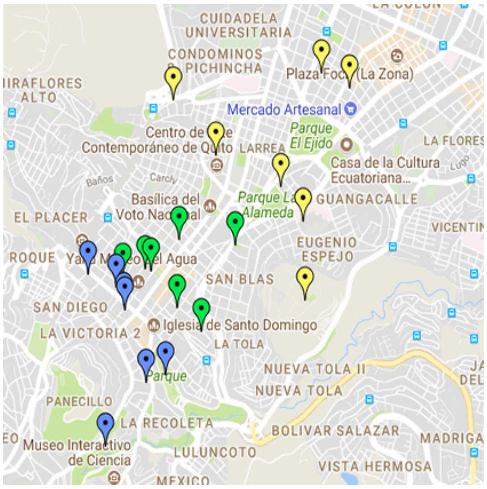


FIGURE 4. POIs clustering with k=3 days.

1) Parameters of the GA

For a GA to succeed in optimization, the correct parameters must be chosen. Some taxonomies differentiate between exogenous and endogenous parameters. Exogenous parameters are generic parameters that define the global properties of a GA such as the size of the population or the probability of crossover. Whereas, endogenous parameters define more specific properties that affect the coding of solutions [34]. For the development of our GA, both endogenous and exogenous parameters were used.

The exogenous parameters used in our approach are:

- The predefined size of the population.
- The total number of generations.
- The probability of crossover.

The endogenous parameters used in the fitness function are:

- The date of the day of the tour.

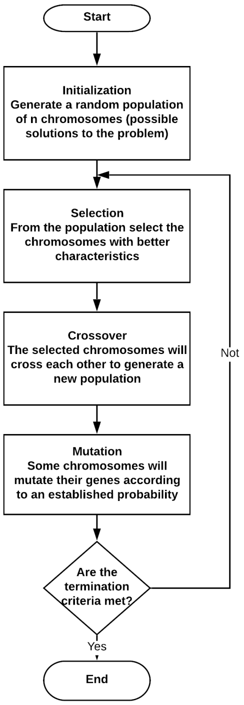
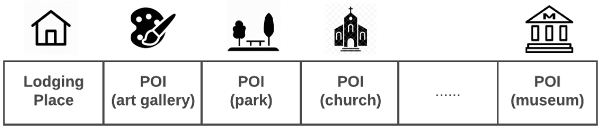

Start
Initialization
Generate a random population of n chromosomes (possible solutions to the problem)
Selection
From the population select the chromosomes with better characteristics
Crossover
The selected chromosomes will cross each other to generate a new population
Mutation
Some chromosomes will mutate their genes according to an established probability
Are the termination criteria met?
Not
Yes
End

FIGURE 5. Genetic algorithm.



🏠	🎨	🌳	⛪		🏛
Lodging Place	POI (art gallery)	POI (park)	POI (church)	......	POI (museum)

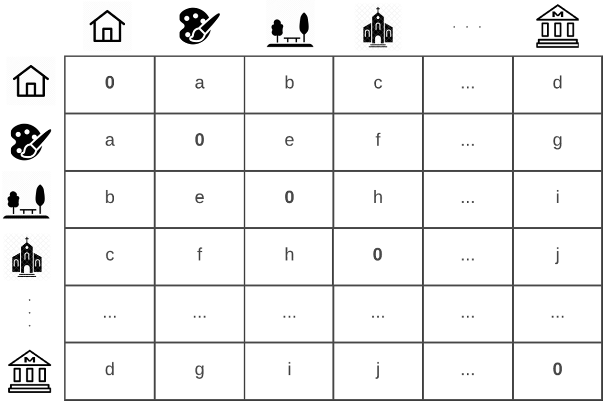
FIGURE 6. Example of clusters.

- A cluster (see Fig. 6) represented as a vector of objects that contains the place of lodging and the points of interest with their detailed information (opening hours).

- A matrix of times of displacement between the different points of the cluster, as shown in Fig. 7.

Each chromosome is coded by permutations of numbers as shown in Fig. 8. Each gene of the chromosome (element of the coded solution) is equivalent to a position within the cluster, so that, if the solution is iterated in order, a tour will be obtained. Starting from the input parameters, a random population will be generated according to the specified size. The genetic code of each individual of the population is composed of the positions of the POIs received in the input vector of the entry. This is observed in the example of Fig. 8, where position 0 will always correspond to the initial location of the tourist. The first point to visit is in position 3 of the POIs vector. The second point is in position N of the POIs vector. The POIs' information includes:

- formatted-address



	🏠	🎨	🌳	⛪	. . .	🏛
🏠	0	a	b	c	...	d
🎨	a	0	e	f	...	g
🌳	b	e	0	h	...	i
⛪	c	f	h	0	...	j
. . .	...	...	...	...	...	...
🏛	d	g	i	j	...	0

FIGURE 7. Matrix of displacement times between different points.

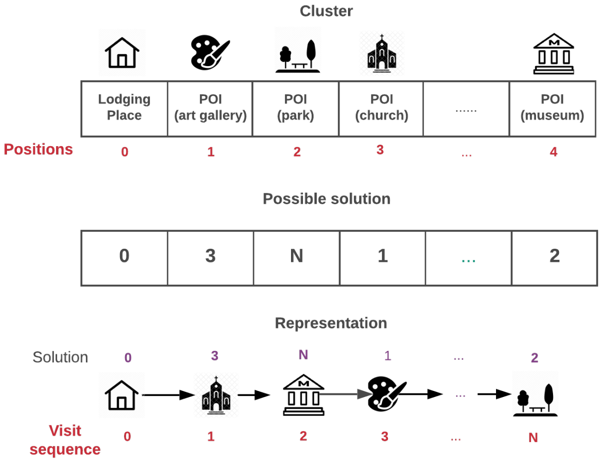


FIGURE 8. Representation of a possible solution.

- formatted-phone-number
- icon
- id
- international-phone-number
- name
- opening-hours
- periods
- place-id
- rating
- url
- utc-offset
- vicinity
- website
- location:lat, lng
- expected-time
- latitude
- length
- weekday-text

- start
- end

2) The fitness function

The fitness function is a particular type of objective function that allows evaluation of the quality of the possible solutions (chromosomes) of a GA. Besides indicating how good a solution is, it can also show how close a chromosome is to being optimal. An ideal fitness function is adapted in order to optimize the algorithm to perform the selection process [35]. Our best contribution is in the fitness function, as we evaluate to optimize itineraries during the execution of the GA.

The feasibility of the solutions is evaluated based on metrics such as the time of travel between the different places, their opening hours and the estimated time to visit each place. The fitness function is detailed in Algorithm 1, which calculates the total time it would take the tourist to visit the points according to the suggested order and then compares it with the available tour time. In addition, time penalties are added in case of arriving outside the opening hours of a POI, and whether or not the itinerary harms the time set for lunch is considered.

The GA explores a fairly large set of solutions and aims to find an optimal solution. In case of finding solutions that have the same fitness, the two are selected again to find new solutions, and so on until they end up with a number of generations. That is, in the end only one solution is returned, the best among the whole set. The best aspect of our proposal is that the recommendations are personalized. The parameterization allows the function to be scalable. The parameters that are used depend on the web services that a tourist destination makes available.

After calculating the fitness value of each individual, we proceeded to select those with the highest score because they are the itineraries that fit better with the available time of a tourist. A metric called "invaded time" was also defined, (calculated in Algorithm 2), which corresponds to the time that the scheduled visit overlaps the scheduled lunch time. For example, if lunch is parameterized between 1 pm and 2 pm, and if an event overlaps the lunch schedule at 1:20 pm, the time invaded will equal 20 minutes. The shorter that time is invaded, the better the quality of the solution.

Algorithm 1 Pseudocode of the fitness function to parameterize the recommender system, in order to adjust it to specific user requirements.

```
Require: All constraints defined by the user
Ensure: The optimal itinerary observing all constraints.
Function getFitness (chromosome, cluster, timeMatrix, tourSchedule, lunchSchedule)
currentTime = tourSchedule.start
availableTime = tourSchedule.end - tourSchedule.start
lengthOfRoute = chromosome.length
travelTime = timeMatrix[chromosome(0)][chromosome(1).traveltime]
investedTime = travelTime
i= 1
while i<lengthOfRoute - 1 do
    currentPOI = cluster[chromosome[i]]
    penalties = getPenalties(currentHour, currentPOI.schedules, travelTime)
    visitTime = currentPOI.timevisit
    durationOfVisit = runEvent(currentHour, visitTime, lunchchedule)
    travelTime = timeMatrix[chromosome[i][chromosome[i+1]].traveltime
    durationTravel = runEvent(currentHour, visitTime, lunchchedule)
    investedTime = investedTime + penalties + durationOfVisit + durationTravel
    i = i+1
end while
lastPOI = cluster[chromosome[lengthOfRoute - 1]]
penalties = getPenalties(currentHour, currentPOI.schedules, travelTime)
investedTime = investedTime + penalties + lastPOI.traveltime
return (availableTime - investedtime)
EndFunction
```

3) Selection

Selection is an important part of genetic algorithms since it significantly affects the degree of convergence [36]. Roulette-wheel selection is a frequently used method in genetic and evolutionary algorithms or in modeling of complex networks [36]. The basic strategy follows the rule: the better fitted an individual, the greater the probability of that individual's survival and opportunity for mating. The selection phase will be responsible for choosing chromosomes for reproduction based on their characteristics. Therefore, the fittest individuals will have greater opportunities to be selected. However, less fit individuals should not be ruled out completely because genetic variability would be lost [33] [36] [37]. An approach based on genetic algorithm could be applied to tourism to offer optimized itineraries. For example, (a) a person with disability can eliminate from the itinerary the POIs that do not provide accessible features and conversely can include POIs that facilitate access for people with disability; (b) if there are many restaurants selected in the morning, they can be removed and, on the other hand, restaurants can be offered according to user' interests, e.g,, only at the schedules required by the user. These are just two examples of the many that can arise in which there may be multiple conditions to optimize an itinerary.

In the proposed solution, the fitness value of each chromosome corresponds to the difference between the time invested and the time available for the tour. The smaller that difference, the better the solution. Adjustment of the fitness value is made before starting the selection phase, as shown in equation (2).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2990348, IEEE Access

Algorithm 2 Pseudocode of the function used to measure the time invested in the execution of an event considering the parameterized lunch schedule

```
Require: the current time, event time, lunch schedule.
Ensure: Measure the time spent executing an event. A metric called "invaded time" is defined.
Function runEvent (currentHour, eventTime, lunchSchedule)
StartOfLunch = lunchSchedule.start
lunchTime = lunchSchedule.end - lunchSchedule.start
totalTime = eventTime
NextHour = currentHour + eventTime
if StartOfLunch $\geq$ currentTime and StartOfLunch $\leq$ nextHour then
    currentHour = currentHour + lunchTime + eventTime
    if (|lunchStart-currentTime|) < (|nextHour-StartOfLunch|) then
        invadedTime = StartOfLunch - currentHour
    else
        invadedTime = nextHour - startOfLunch
    end if
    totalTime = totalTime + lunchTime + invadedTime
else
    currentHour = currenthour + eventTime
end if
return (totalTime)
EndFunction
```

$$AdjustedFitness = 1/currentFitness \tag{2}$$

Inverting its value ensures that less fit chromosomes will have a better chance of being selected.

We present a roulette-wheel selection algorithm. It selection process consists of choosing the chromosomes with the best characteristics within the population. For this, we need to evaluate each chromosome with a function called fitness that works as equation (3), where: $T_{(tourSchedule.end-tourSchedule.start)}$ is the available time that the tourist has per day; $\sum_{i=1}^{n} t_{(i)expected}$ is the iteration of POIs to visit according to the order of the individual genetic code; $penalties_{(i)}$ is a function that returns the time that the tourist would lose in the case where the site is not open in their mobilization schedules; $\sum_{i=0}^{n-1} t_{(i,(i+1)travel)}$ is the iteration where the travel time between the different points is obtained; $t_{(i)}$ expected is the average time that it takes to visit a certain POI; $t_{(i,i+1)travel}$ is the travel time from $POI_{(i)}$ to $POI_{(i+1)}$ and $t_{timeSpentOnLunch}$ is the time spent on lunch, and $-t_{invadedLunchTime}$ is the time invaded on the time defined to lunch. The final value is calculated as the absolute value.

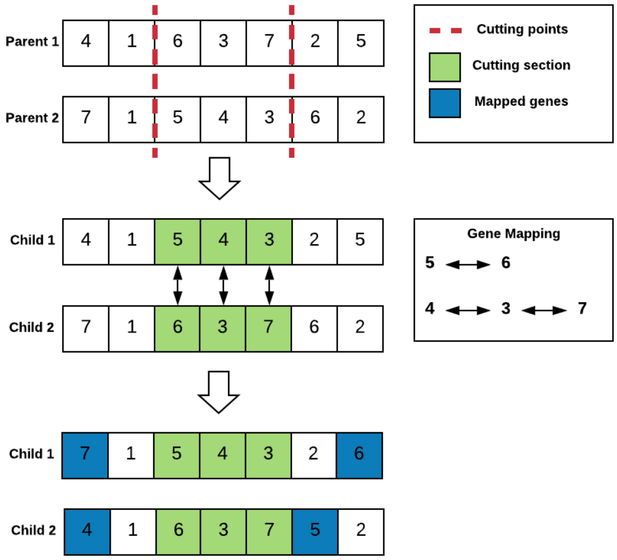


FIGURE 9. Partially mapped crossover.

$$\begin{aligned} fitness = \|&T_{(tourSchedule.end-tourSchedule.start)} \\ &-\sum_{i=1}^{n}(t_{(i)expected}+penalties_{(i)}) \\ &-\sum_{i=0}^{n-1} t_{(i,(i+1))travel}) \\ &-t_{timeSpentOnLunch} \\ &-t_{invadedLunchTime}\| \end{aligned} \tag{3}$$

4) Crossover

After selecting the individuals, we proceeded with the process of crossing them. The crossover or crossing is an operation that allows the recombination of two or more chromosomes to produce offspring that will be part of a new generation. The fundamental idea of crossover is that the new individuals will inherit the best characteristics of their parents and represent a better solution [33]. There are many techniques for applying crossover such as crossing a point, crossing two points and uniform crossing; however, none of these techniques is useful for permutation-encoded solutions since they would alter the sequence of numbers and repeat genes. For this reason, we used the *Partially-mapped crossover* technique, since it is an algorithm that works with a permutation combination.

For example, the *partially-mapped crossover* observed in Fig. 9 starts by selecting two random cut points and executes the same crossing procedure by two points. Subsequently, in the new solutions, a mapping of genes outside the cutting section is carried out in order to eliminate duplicates. The mapping corresponds to an inverse replacement of the genes that were selected in the cut area [38].

5) Mutation

The chromosomes are subject to a mutation process where their genetic code is modified probabilistically. Mutation plays an important role since it helps maintain genetic diversity in the population and allows exploration of more options within the solution space. There are some types of mutation that vary according to the coding of the solutions; for permutation coding (used in our research), one can select random genes within the chromosome and exchange their positions [32].

V. EXAMPLE OF EXECUTION

The aim of our proposal is to improve tourists' experience by optimizing POIs and itineraries. It is important to note that in our case, we are not only looking for an optimal itinerary, but also maximizing tourists' enjoyment or profit according to their interests. For this, we considered the following parameters:

- The user is asked the departure time for each day and the time they wish to have lunch.
- The time required to get from one place to another.
- The time required to visit a place.
- The user's preferences.
- The constraints of both the visitor and the POIs.
- To control access to POIs we use a penalty function in our algorithm. Penalties are calculated based on punctuality:
  - There is no penalty:
    - When the user arrives during the POI service hours.
  - There is a penalty:
    - When the user must wait for the site to open.
    - In the case of arriving after the scheduled time.

In our research, we aim to improve the tourist experience by optimizing POIs and itineraries. As detailed earlier, one advantage of our approach lies in the use of actual information about POIs from the Internet. To prove that our solution can be adapted to any environment, we tested our recommender over visits to three cities: Paris, Rome and New York. For each tourist destination, we used the recommender twice, for a total of six tests.

The results of our test are presented in Figs. 10. In these figures, the itinerary of a visit for each day is shown in different colors. Each POI is represented with a mark. The user's interests for each itinerary in Paris, Roma and New York are observed in Table 2.

TABLE 2. User's preferences presented in order

City	Days	Preferences
Paris	5	Movie-theatre, museum, cemetery, university, mosque
Paris	4	Park, museum, city hall, movie-theatre
Rome	7	Zoo, aquarium, natural feature, stadium, museum
Rome	5	University, city hall, mosque, amusement park
New York	4	Stadium, aquarium, cemetery, synagogue
New York	3	Museum, art gallery, natural feature

Fig. 10a presents six POIs to visit and five options of restaurants. Table 3 shows the details of the best itinerary for the first day of a five-day visit to Paris; we identified this with a red line. From the complete list of POIs that are in that trajectory, the most important ones are chosen considering the user's interests and without neglecting the context of the city. In this case, the day starts at 9:45 am and ends at 18:00 pm and the lunch time is from 13:45 pm to 14:45 pm. First it was necessary to identify the coordinates where the hotel is located. Then, from there the clusters were identified with the most important POIs. The route optimized by time metrics was presented, such as: opening hours of establishments, mobilization between them, lunch time and the estimated time of visit (which at the moment are random times) until an Application Programming Interface (API) is available that delivers this data. As shown in Table 3, the resulting POIs are the most important, well located geographically and with real time control. The remaining figures present similar results that consider both the user's interests and the context of the city in different countries.

VI. EVALUATION

We divide the assessment of our system into two parts. The first part is the evaluation via a survey to know the factors that influence the subsequent use of the recommender system, and the second part applies metrics to evaluate the qualite of recommended POIs.

To evaluate our recommender, we conducted an experiment with 131 tourists who used the recommender (16 % of tourists are older than 40 years old, and the 84 % are younger than 40 but older than 18). Fig. 11 shows a screenshot of the experiment interface, and in the Figs. 12 to 15 we present the calculated itinerary for each day.

A. EVALUATION THROUGH A SURVEY

Fig. 16 presents the model for recommender analysis. Our model is based on the user's perceptions [39]. We considered the user perception to identify factors that influence user preference and quality of experience. We need to know what factors influence the subsequent use of the recommender system [39].

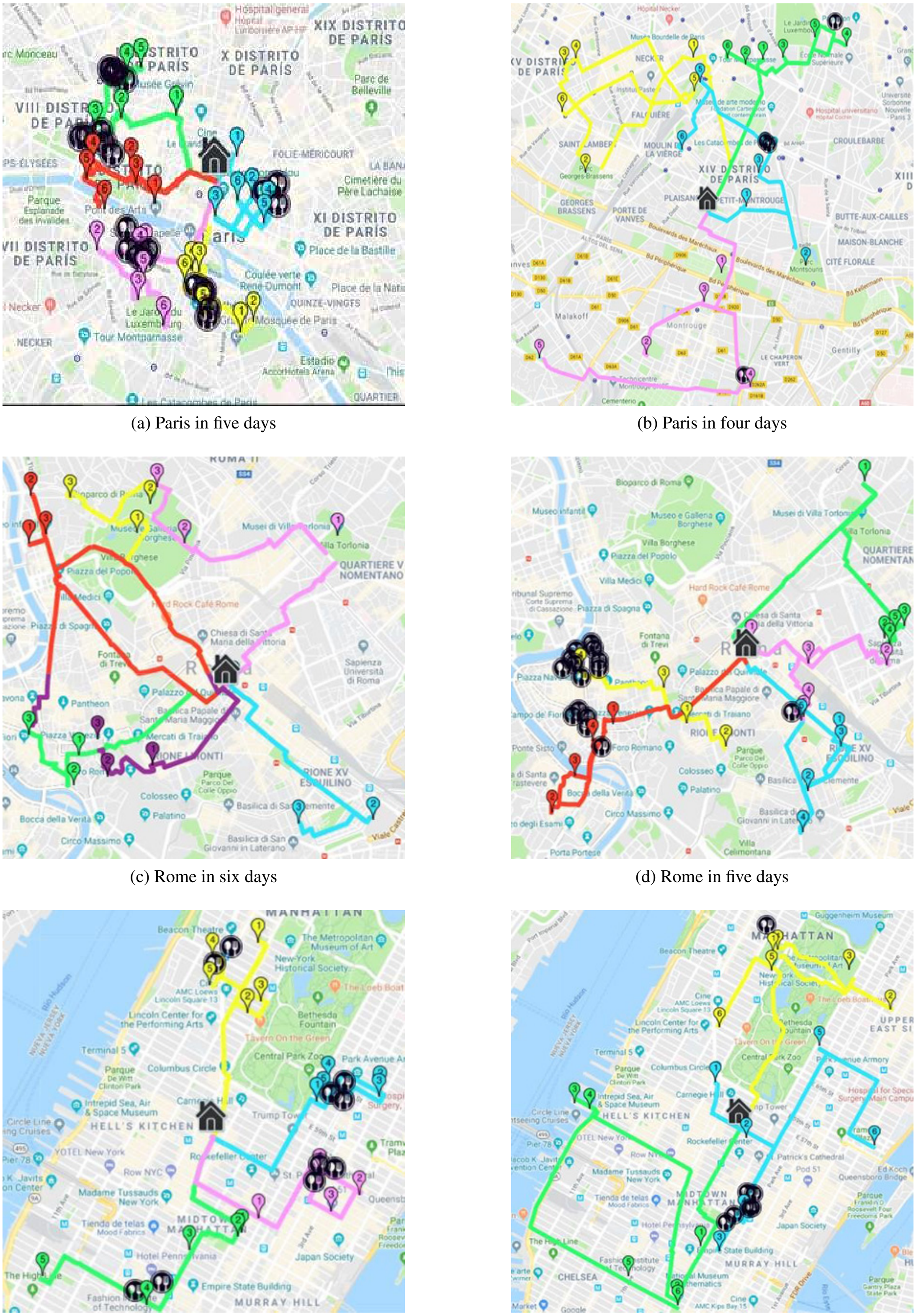
In this model, the variables are as follows:

- *Independent variables:* understand-me and satisfaction.
- *Dependent variables:* perceived novelty; accuracy and diversity, and attitude toward using.

The characteristics to define the user's perception are distributed as follows [39]:

- *Understand-me:* the recommender understands their tastes and can effectively adapt to them.
- *Satisfaction:* the user's overall satisfaction with the recommender and their perception of its usefulness.
- *Accuracy:* the recommender's ability to find good tourist recommendations.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2990348, IEEE Access



(a) Paris in five days

(b) Paris in four days

(c) Rome in six days

(d) Rome in five days

(e) New York in four days

(f) New York in three days

FIGURE 10. Itinerary of visits.

TABLE 3. Fig. 10a, First day from the five-day itinerary for Paris. **User's interests: Movie-theatre, museum, city-hall, cemetery, university, mosque**. Wednesday: start: 09:45 am, end: 18:00 pm. Lunch: 13:45 pm 14:55 pm

Number of days: 5 Hotel: lat: 48.863456 lng: 2.350454	**Time on route**	**POI**	**Schedule**	**Entry time and exit time (time the user stays in one place)**
Day 1: Red itinerary	Departure: 9:45 am Arrival: 10:00 am.	Louvre Museum	Monday: 09:00 am-18:00 pm, Tuesday: closed, Wednesday: 09:00 am-21:45 pm, Thursday: 09:00 am-18:00 pm, Friday: 09:00 am-21:45 pm, Saturday: 09:00 am-18:00 pm, Sunday: 09:00 am-18:00 pm	Start: 10:00 am End: 10:59 am
	Departure 10:59 am Arrival: 11:09 am	Saint Roch	Monday: 08:30 am 19:00 pm, Tuesday: 08:30 am 19:00 pm, Wednesday: 08:30 am 19:00 pm, Thursday: 08:30 am 19:00 pm Friday: 08:30 am 19:00 pm Saturday: 08:30 am 19:00 pm Sunday: 08:30 am 19:00 pm	Start: 11:09 am End: 12:10 pm
	Departure 12:10 pm Arrival: 12:14 pm.	Museum of decorative arts	Monday : closed, Tuesday: 11:00 am 18:00 pm, Wednesday: 11:00 am 18:00 pm, Thursday: 11:00 am 21:00 pm, Friday: 11:00 am 18:00 pm, Saturday: 11:00 am 18:00 pm, Sunday: 11:00 am 18:00 pm	Start: 12:14 pm End: 13:44 pm
Lunch 13:44 pm to 14:45 pm	**Recommendation includes five restaurants**			
	Departure 14:45 pm Arrival: 14:57 pm.	Jeu de Pomme	Monday: closed Tuesday: 12:00 am 21:00 pm, Wednesday: 12:00 am 20:00 pm, Thursday: 12:00 am 20:00 pm, Friday: 12:00 am 20:00 pm, Saturday: 11:00 am 20:00 pm, Sunday: 11:00 am 20:00 pm,	Start: 14:57 pm End: 15:57 pm
	Departure 15:57 pm Arrival: 16:03 pm	Jardin Tuileries	Monday: 07:00 am 21:00 pm Tuesday: 07:00 am 21:00 pm Wednesday: 07:00 am 21:00 pm Thursday: 07:00 am 21:00 pm Friday: 07:00 am 21:00 pm Saturday: 07:00 am 21:00 pm Sunday: 07:00 am 21:00 pm	Start: 16:03 pm End: 17:05 pm
	Departure 17:05 pm Arrival: 17:13 pm.	Orsay Museum	Monday: closed, Tuesday: 09:30 am 18:00 pm, Wednesday: 09:30 am 18:00 pm, Thursday: 09:30 am 21:45 pm, Friday: 09:30 am 18:00 pm, Saturday: 09:30 am 18:00 pm, Sunday: 09:30 am 18:00 pm	Start: 15:13 pm End: 18:00 pm

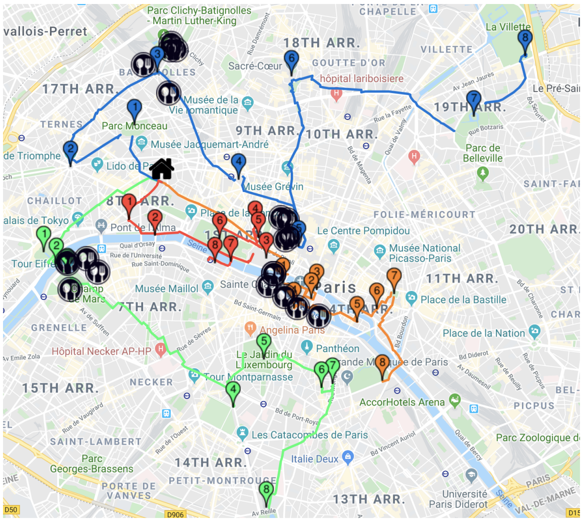


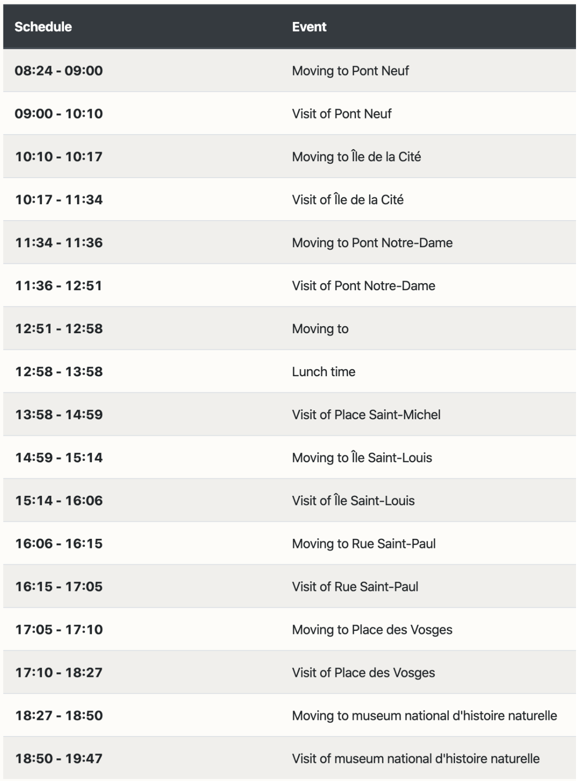
FIGURE 11. Visit Paris in four days.

- *Diversity:* the diversity of the recommended routes on different days.
- *Novelty:* the propensity of the recommenders to suggest items with which the user is unfamiliar.
- *Attitude toward using:* the user would recommend and/or use the recommender system again.

Table 4 presents 16 questions about various aspects designed to measure the users perception on tourism recommendation across five factors. The first column present the User's perceptions, the second column provides the respective item or indicator and the third column shows the question associated to each indicator.

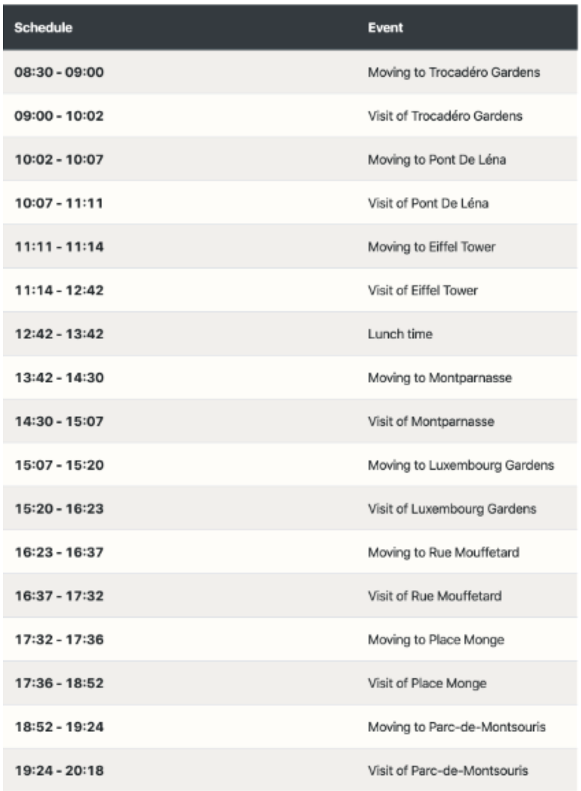
1) Results

To analyze the survey data we have used partial least squares structural equation modeling (PLS -SEM) in Smart PLS software. We have opted for a reflective model observed in Fig. 17. In our model there are five perceptions: understand-me, satisfaction, accuracy and diversity, novelty, and attitude towards using. In this case, we have chosen to measure the latent perceptions with reflective indicators. Table 5 presents the consolidated results after applying the model to the data resulting from the survey. As observed, most of the indicators are in the specified range. The more important to analyse is the question P5-Accur = 0.696: "Does the recommendation system allow me to access restaurants in the periods of time I chose?" This result could be explained because a relevant part of the population consulted were young people (between 18 and 30 years of age) who do not



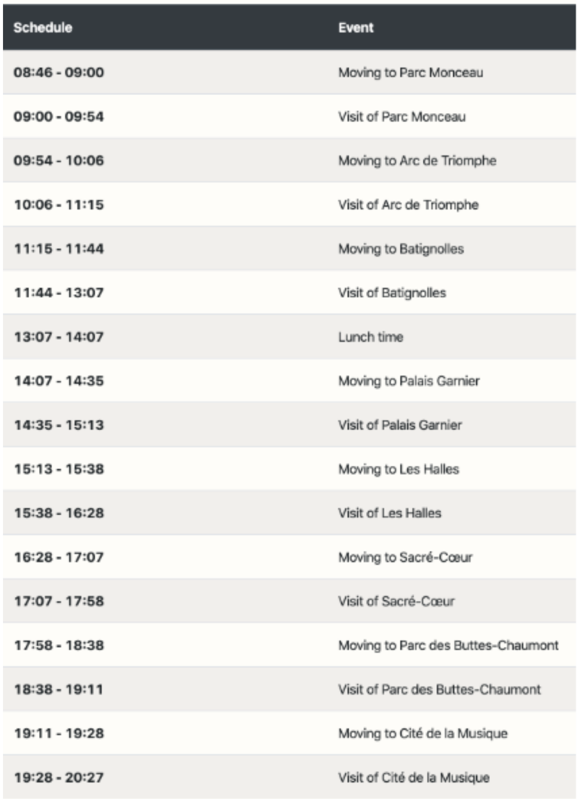
Schedule	Event
08:24 - 09:00	Moving to Pont Neuf
09:00 - 10:10	Visit of Pont Neuf
10:10 - 10:17	Moving to Île de la Cité
10:17 - 11:34	Visit of Île de la Cité
11:34 - 11:36	Moving to Pont Notre-Dame
11:36 - 12:51	Visit of Pont Notre-Dame
12:51 - 12:58	Moving to
12:58 - 13:58	Lunch time
13:58 - 14:59	Visit of Place Saint-Michel
14:59 - 15:14	Moving to Île Saint-Louis
15:14 - 16:06	Visit of Île Saint-Louis
16:06 - 16:15	Moving to Rue Saint-Paul
16:15 - 17:05	Visit of Rue Saint-Paul
17:05 - 17:10	Moving to Place des Vosges
17:10 - 18:27	Visit of Place des Vosges
18:27 - 18:50	Moving to museum national d'histoire naturelle
18:50 - 19:47	Visit of museum national d'histoire naturelle

FIGURE 12. Visit Paris in four days. Itinerary-day1.



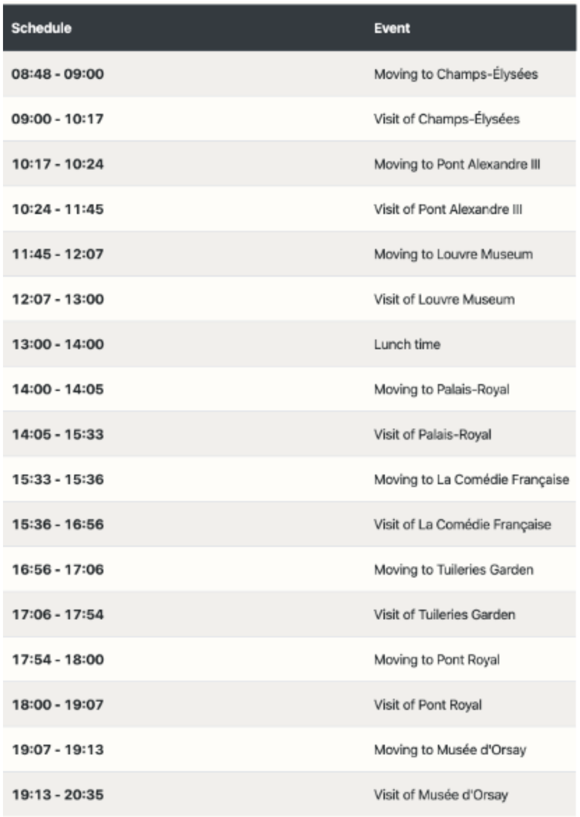
Schedule	Event
08:30 - 09:00	Moving to Trocadéro Gardens
09:00 - 10:02	Visit of Trocadéro Gardens
10:02 - 10:07	Moving to Pont De Léna
10:07 - 11:11	Visit of Pont De Léna
11:11 - 11:14	Moving to Eiffel Tower
11:14 - 12:42	Visit of Eiffel Tower
12:42 - 13:42	Lunch time
13:42 - 14:30	Moving to Montparnasse
14:30 - 15:07	Visit of Montparnasse
15:07 - 15:20	Moving to Luxembourg Gardens
15:20 - 16:23	Visit of Luxembourg Gardens
16:23 - 16:37	Moving to Rue Mouffetard
16:37 - 17:32	Visit of Rue Mouffetard
17:32 - 17:36	Moving to Place Monge
17:36 - 18:52	Visit of Place Monge
18:52 - 19:24	Moving to Parc-de-Montsouris
19:24 - 20:18	Visit of Parc-de-Montsouris

FIGURE 13. Visit Paris in four days. Itinerary-day2.



Schedule	Event
08:46 - 09:00	Moving to Parc Monceau
09:00 - 09:54	Visit of Parc Monceau
09:54 - 10:06	Moving to Arc de Triomphe
10:06 - 11:15	Visit of Arc de Triomphe
11:15 - 11:44	Moving to Batignolles
11:44 - 13:07	Visit of Batignolles
13:07 - 14:07	Lunch time
14:07 - 14:35	Moving to Palais Garnier
14:35 - 15:13	Visit of Palais Garnier
15:13 - 15:38	Moving to Les Halles
15:38 - 16:28	Visit of Les Halles
16:28 - 17:07	Moving to Sacré-Cœur
17:07 - 17:58	Visit of Sacré-Cœur
17:58 - 18:38	Moving to Parc des Buttes-Chaumont
18:38 - 19:11	Visit of Parc des Buttes-Chaumont
19:11 - 19:28	Moving to Cité de la Musique
19:28 - 20:27	Visit of Cité de la Musique

FIGURE 14. Visit Paris in four days. Itinerary-day3.



Schedule	Event
08:48 - 09:00	Moving to Champs-Élysées
09:00 - 10:17	Visit of Champs-Élysées
10:17 - 10:24	Moving to Pont Alexandre III
10:24 - 11:45	Visit of Pont Alexandre III
11:45 - 12:07	Moving to Louvre Museum
12:07 - 13:00	Visit of Louvre Museum
13:00 - 14:00	Lunch time
14:00 - 14:05	Moving to Palais-Royal
14:05 - 15:33	Visit of Palais-Royal
15:33 - 15:36	Moving to La Comédie Française
15:36 - 16:56	Visit of La Comédie Française
16:56 - 17:06	Moving to Tuileries Garden
17:06 - 17:54	Visit of Tuileries Garden
17:54 - 18:00	Moving to Pont Royal
18:00 - 19:07	Visit of Pont Royal
19:07 - 19:13	Moving to Musée d'Orsay
19:13 - 20:35	Visit of Musée d'Orsay

FIGURE 15. Visit Paris in four days. Itinerary-day4.

TABLE 4. Recommender system survey

User perceptions	Item or indicator	Questions
Understand-me	P1-Unders	Is it clear that the recommendation system presents recommendations based on my personal interest, when run for the first time?
	P2-Unders	Is it clear that the recommendation system updates-recommendations, based on my personal interest?
Satisfaction	P3-Satisf	In general, did the routes and points of interest provided by the recommendation system help me enjoy the visit?
Accuracy	P4-Accur	Were the number of points of interest proposed adequate for the time available on each day of the visit?
	P5-Accur	Does the recommendation system allow me to access restaurants at the periods of time I chose?
	P6-Accur	Did the periods of time allocated for traveling between points of interest coincide with the time I used while travelinng between the proposed points of interest?
	P7-Accur	Did the periods assigned to visit each point of interest seem appropriate?
	P8-Accur	Did the use of the recommendation system improve the quality of my visit?
	P9-Accur-Div	In general, did I find the recommendation system useful for my visit?
	P10-Accur-Div	In general, did the proposed routes for each day meet my expectations?
Novelty	P11-Novel	Does understanding the route analysis presented on the recommendation system map require mental effort?
	P12-Novel	Was the presentation of points of interest and routes on the map clear and understandable?
	P13-Novel	It is clear that the points of interest presented on the routes were the most important in the visiting area?
	P14-Novel	Does the presentation of the recommendations system routes facilitate the location of important points of interest in my visit?
Attitude toward using	P15-Attit	Would you suggest using our recommendation system to others?
	P16-Attit	For my next trips, would I use the recommendation system again?

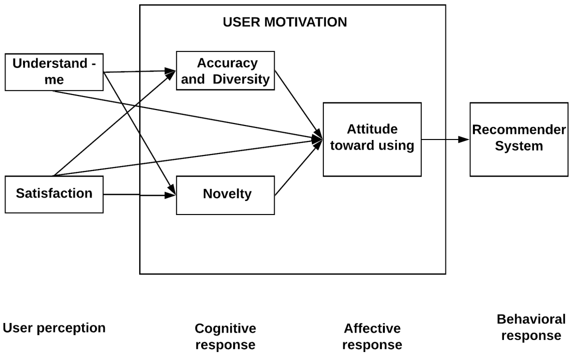


FIGURE 16. Model proposal for recommender analysis.

give relevant importance to the suggestions of restaurants. For the rest of the population, over 40 years of age, this seems important. For this reason, we will study how to improve the presentation of restaurants in our future work. Concerning P11-Novel = 0.580: "Does the understanding the route analysis presented on the recommendation system map require mental effort?" The result is indicative for our purposes because our objective is not linked to the analyze of the interface. However, this result will be very helpful to improve our future work.

In general, the indicators analyzed of the each perception are strongly significant. In conclusion, accuracy and diversity have the greatest influence on the attitude towards use of our recommender system. This is positive because it has been ratified that the recommended POIs satisfies to the tourist. We are sure that our recommender system has a good level regarding offering accuracy and diversity.

B. APPLYING METRICS FOR DETECTING THE PRECISION OF OUR RECOMMENDER SYSTEM

To complete the evaluation, we determined the degree of relevance of the proposed POIs. We have considered the three most frequent and basic measures for information retrieval effectiveness: *precision*, *recall* and *F-measure* [3] [40]. We applied evaluation of unranked retrieval sets.

Precision (P) in equation (4) is the fraction of retrieved POIs that are relevant.

$$Precision = (Relevant\ items\ retrieved)/(Retrieved\ items)$$
$$P = (Relevant/Retrieved) \tag{4}$$

Recall (R) in equation (5) is the fraction of relevant POIs that are retrieved.

$$Recall = (Relevant\ items\ retrieved)/(Relevant\ items)$$
$$R = (Retrieved/Relevant) \tag{5}$$

[3] https://nlp.stanford.edu/IR-book/pdf/08eval.pdf

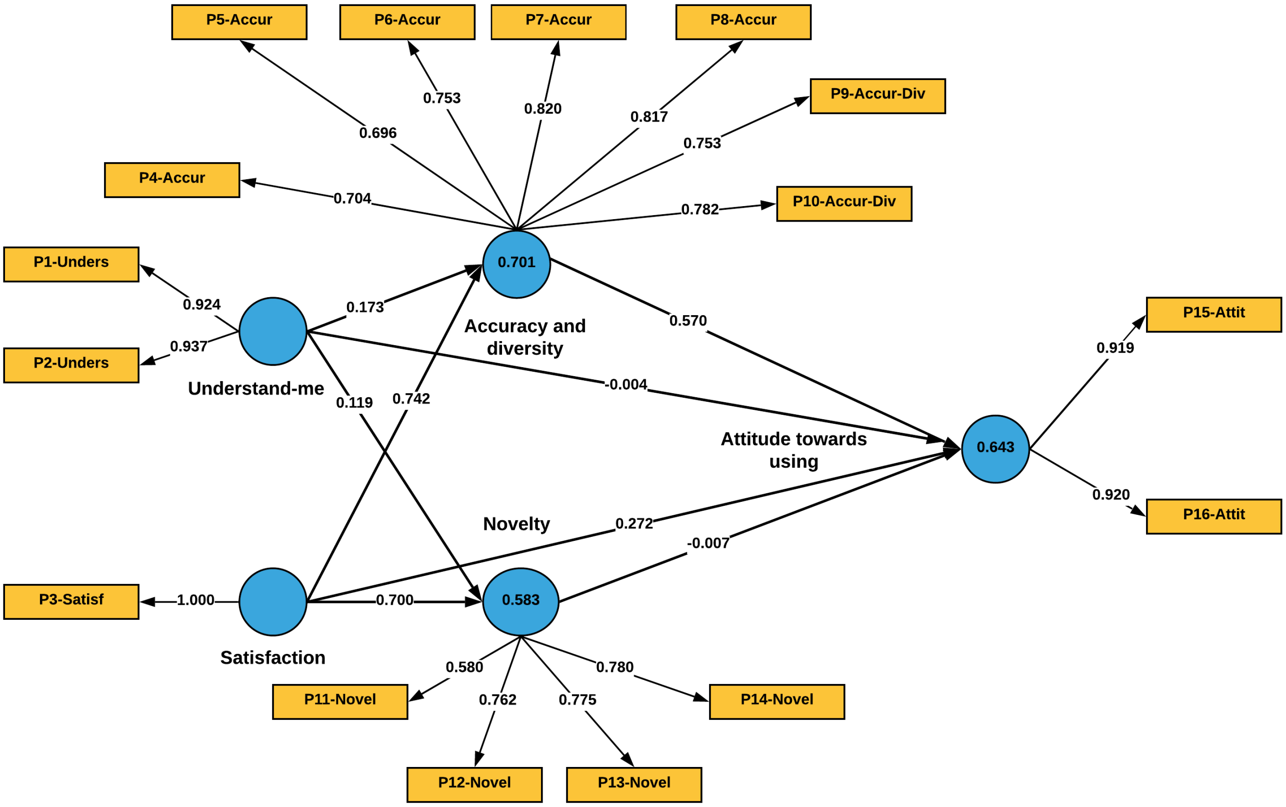


FIGURE 17. Relation exogenous and endogenous perceptions, data base for determine the acctitude toward using in our recommender system proposal.

TABLE 5. Indicator reliability and composite reliability

Latent variable	Item or indicator	R^2	factor loads	Convergent validity		Internal consistency	
			> 0.70	**Indicator reliability** > 0.50	**AVE** > 0.50	**composite reliability** $0.70 - 0.90$	**Cronbach's alpha** $0.70 - 0.90$
Understand-me	P1-Unders		0.924	0.854	0.867	0.919	0.846
	P2-Unders		0.937	0.878			
Satisfaction	P3-Satisf		Single Item Construct				
Accuracy and diversity	P4-Accur	0.701	0.704	0.496	0.581	0.906	0.879
	P5-Accur		0.696	0.484			
	P6-Accur		0.753	0.567			
	P7-Accur		0.820	0.672			
	P8-Accur		0.817	0.668			
	P9-Accur-Div		0.753	0.567			
	P10-Accur-Div		0.782	0.612			
Novelty	P11-Novel	0.583	0.580	0.336	0.532	0.818	0.705
	P12-Novel		0.762	0.581			
	P13-Novel		0.775	0.601			
	P14-Novel		0.780	0.608			
Attitud towards using	P15-Attit	0.643	0.919	0.845	0.845	0.916	0.817
	P16-Attit		0.920	0.846			

And F-measure in equation (6) or balanced F-score is the median of precision and recall.

$$F-measure = 2{\cdot}(Precision{\times}Recall)/(Precision{+}Recall) \tag{6}$$

To evaluate precision and recall, we have considered recommending a visit to Paris for four days, which is observed in Fig. 11. It is necessary to clarify that many relevant POIs can be excluded because we analyzed based on user interests, without forgetting the context of the city.

Table 6 shows, in the first column, the number of recommended POIs; the second column represents the number of recommended POIs that are relevant; and the third column displays the number of important and relevant POIS. Precision is an important factor to evaluate in our recommender and shows that the recommended POIs are relevant. Recall is the total relevant POIs that were recommended. And F-measure is the media measure.

Table 6 is structured in two parts;

1) In the first part, we analyze five data sets.
    a) First, a tourist provided 10 important POIs in such a large city; however, the system recommends far more POIs than those expected by the tourist. The precision being 28 %, recall 90 % and F-measure 42 %. The analysis of this data is shown on Table 7.
    b) Next, in order to clearly whether the proposal also covers the context of city. We analyzed the 33 POIs of Paris ranked by Google[6] were taken. In this case, by having a source ranked by Google from the information of the users. It is observed that the precision of 71 % was good, recall 69 % and F-measure was 70 %. To understand these results, Table 8 shown the same 32 POIs which we have proposed to the tourist; but we want to compare them with qualified POIs and suggested now by Google. As observed the precision and recall were greater. It should be recognized that the recommender system also identifies POIs qualified by the Google, allowing, the tourist to identify other POIs that help to recognize the context of the city As well as information for visitors as well as POIs of their interest.
    c) The 54 POIs relevant observed from the point of view of a tourist guide[7]. The precision improved to 84 %, recall 50 % and F-measure 62 %.
2) In the second part of the Table 6, we present the information provided for other tourist.
    a) A second tourist provided ten important POIs in such seven POIs were relevant. For the four day visit, the system recommender presented 31 recommended POIs. In this case, the precision was 22 % but the recall was 70 % and the F-measure was 33 %. Although the accuracy was low, the degree of recovery for the tourist was high. These results are favorable for the recommender system.
    b) The third tourist provided 10 important POIs in such the recommender system presented 15 recommended POIs. The precision was 22 %, but recall was 70 % and the F-measure was 33 %. Similarly, although the accuracy is low, the number of important POIs recovered is high.

From the results, we can conclude that the precision metric of the recommender system is reduced when the tourist describes few points to visit (this happens with the three tourists consulted). However, we are sure that our recommender system enriches the visit and helps tourists enjoy their visit. For this reason, we have chosen to compare the POIs recommended by the first tourist recommender system with POIs suggested by specialized websites. As a result of this analysis, we can conclude that when the precision metric is greater than 70 % it is good, both with the data set of the tourist guide and with the data set of the Google ranking. However, we are sure that our recommender system meets the expected expectations for tourist.

VII. DISCUSSION

According to the results reported in Section VI, our proposal for solving the TTDP offers a good improvement in results. A case study was conducted in Paris for a single user over four days. Regarding the interests of other users, we have differentiated ourselves from other researchers in that we analyze interests from tweets posted by Twitter users using big data techniques [3]. In our theoretical analysis, specified in Section III, we have observed that although there are many solutions for TTDP, none clarify exactly where to include each of the improvements that each researcher proposes. With this project, we resolve where and how to do each required activity to develop a complete touristic recommender.

The metaheuristic algorithms with their improved functions used in this study allowed us to achieve significantly greater profit and produce better recommendations. The fitness function algorithm was able to design more realistic and more personalized itineraries for the tourist, because our algorithm parameterizes all the conditions and constraints of the user and POIs and the web services that the tourist places offer. Since we take into account the actual data of POIs obtained from the Internet, our recommender is more realistic, scalable and adaptable to environmental changes. These characteristics supposes the main novelty of our approach because, as mentioned in Section III, no studies have explored TTDP using realistic POIs. Nonetheless, as the data employed comes from the Internet, our solution introduces a risk because there are some cities for which the information on the Internet, for some POIs, is neither reliable nor available. On the other hand, most of the research on

[6] https://www.google.com/destination?hl=es\&gl=us\&output=search\&dest_mid=/m/05qtj#dest_mid=/m/05qtj\&tcfs=EhIKCC9tLzA1cXRqEgZQYXLDrXM

[7] https://www.easyvoyage.com/france/paris

TABLE 6. Results Metrics: Precision, Recalls and F-measure

Tourist in Paris	**Recommended POIs**	**Recommended ∩ Relevance**	**Relevant POIs**	**Precision**	**Recalls**	**F-measure**
First tourist, Fig. 11, (four days)	32	9	10	28 %	90 %	42 %
Google travel guide [4]	32	23	33	71 %	69 %	70 %
Tour guide [5]	32	27	54	84 %	50 %	62 %
Analysis for other users						
	Recommended	**Recommended ∩ Relevance**	**Relevance**	**Precision**	**Recalls**	**F-measure**
Second tourist (four days)	31	7	10	22 %	70 %	33 %
Third tourist (two days)	15	9	10	6 %	90 %	72 %

TABLE 7. Paris (four days): Retrieved and Relevance, Including Tourist's Responses

Days	**Recommended POIs**	**Recommended ∩ Relevance**	**Relevant POIs defined by tourist**
Day 1	Champs-Élysées Avenue	Champs-Élysées Avenue	**Eiffel Tower**
	Alexandre III Bridge		**Louvre Museum**
	Louvre Museum	Louvre Museum	**Arch of Triomphe**
	Royal Palace	Royal Palace	**Champs-Élysées Avenue**
	La Comédie Française		**Tuileries Garden**
	Tuileries Garden	Tuileries Garden	**MontMartre neighborhood**
	Royal Bridge		**Sacré-Cœur Cathedral**
	Orsay Museum	Orsay Museum	**Pompidou Museum**
Day 2	Monceau Park		**Royal Palace**
	Arch of Triumph	Arch of Triomphe	**Sena River**
	Batignolles		
	Garnier Palace		
	Les Halles		
	Sacré-Cœur Cathedrale	Sacré-Cœur Cathedrale	
	Buttes-Chaumont Park	Buttes-Chaumont Park	
	Musique City		
Day 3	Trocadero Gardens		
	d'd'Iéna Bridge		
	Eiffel Tower	Eiffel Tower	
	Montparnasse		
	Luxembourg Gardens		
	Mouffetard Street		
	Monge Square		
	Montsouris Park		
Day 4	Neuf Bridge		
	Île de la Cité		
	Notre-Dame Bridge		
	Saint Michel Square		
	Île San Louis		
	Saint-Paul Street		
	des Vosges Square		
	National d'histoire naturelle Museum		

TABLE 8. Paris four days: Retrieved and Relevant, including Google Ranking

Days	Recommended POIs	Recommended ∩ Relevant	Relevant POIs defined by Google
Day 1	Champs-Élysées Avenue	Champs-Élysées Avenue	**Palais Garnier, Ranking Google (RG): RG=4.7**
	Alexandre III bridge	Alexandre III bridge	**Notre-Dame de Paris Cathedrale, RG=4.7**
	Louvre Museum	Louvre Museum	**Louvre Museum, RG= 4.7**
	Palais Royal	Palais Royal	**Orsay Museum, RG= 4.7**
	La Comédie Française	La Comédie Française	**Sacré-Cœur Basilica, RG=4.7**
	Tuileríes garden	Tuileríes Garden	**Garnier opera, RG=4.7**
	Royal bridge		**Alexandre III bridge RG=4.7**
	Orsay Museum	Orsay Museum	**La Comédie Française, RG=4.7**
Day 2	Monceau park	Monceau park	**Luxembourg garden, RG=4.7, Notre-Dam bridge, RG=4.7**
	Arch of Triomphe	Arch de Triomphe	**Eiffel Tower, RG=4.6**
	Batignolles		**Arch of Triomphe, RG=4.6**
	Palais Garnier	Palais Garnier	**de Vosges square, RG= 4.6**
	Les Halles		**Sainte-Chapelle, RG=4.6**
	Sacré-Cœur Basilica	Sacré-Cœur Basilica	**Tuileries garden, RG=4.6**
	Buttes-Chaumont park	Buttes-Chaumont park	**d'Orangerie museum, RG= 4.6**
	Cité de la Musique	Cité de la Musique	**Buttes-Chaumont park, RG=4.6**
Day 3	Trocadero gardens	Trocadero gardens	**Cité de la Musique, RG=4.6**
	de L'éna bridge	de L'éna bridge	**Trocadéro gardens, RG=4.6**
	Eiffel Tower	Eiffel Tower	**De Léna bridge, RG=4.6**
	Montparnasse		**Neuf bridge, RG=4.6**
	Luxembourg gardens	Luxembourg gardens	**de la Cité island, RG=4.6**
	Mouffetard street		**Saint-Louis island, RG=4.6**
	Monge square		**de la Concorde square, RG=4.5**
	Montsouris park		**de la Bastille opera RG= 4.5**
Day 4	Neuf bridge	Neuf bridge	**Saint Pierre de Montmartre church, RG=4.6**
	de la Cité island	de la Cité island	**Mont Martre museum, RG=4.5**
	Notre Dame bridge	Notre Dame bridge	**de la Madeleine church, RG= 4.5**
	Saint Michel square		**Monceau park, RG=4.5**
	San Louis island	San Louis island	**National d'histoire naturelle museum, RG=4.5**
	Saint-Paul street		**du Tertre square, RG=4.4**
	des Vosges square	des Vosges square	**Museum and cultural center George Pompidou, RG= 4.4**
	National d'histoire naturelle museum	National d'histoire naturelle museum	**La Conciergerie RG=4.3**

the topic has worked with a limited number of standardized POIs. Since we can manage an unlimited number of real POIs, our recommendations are only limited by the time available to tourists.

Concerning the adaptation of our recommendation to the user's interests, our recommender presents only the POIs identified as the most suitable for each user. This enables each user to obtain a clean itinerary that is easy-to-use and enjoyable. What is more, no other contributions have introduced time restrictions for their recommendations. Our solution proved the importance of introducing them to avoid the "restaurant problem", that is, avoid recommending restaurants at any time. Thus, our proposed solution differs from the contributions mentioned in the related work section in that: (1) in the GA algorithm, we use an improved fitness function to value POIs, the original fitness only defines a path, but our fitness chooses the better POIs. Additionally, we have exposed in which part of the optimization algorithm all constraints specified by both the user and the context of the visited a tourist destination can be included; (2) we deal with real POIs information with their real constraints, which are obtained from the Internet. This allows the system to generate recommendations for any tourist destination in the world; (3) the user's interests are considered to filter the POIs; and (4), our itineraries offer recommendations adapted to human behavior such as restaurants only at lunch times.

In GAs, it is necessary to use a specific technique to find the optimal solution. It is also neccesary to define a fitness function adapted to the problem. The fitness function f(x) evaluates each chromosome (x) in the population [41]. In our case, the fitness function solves the time used to visit POIs and traveling between POIs to find the best solution. In our research, we have looked for other algorithms similar to ours, but we have not found a similar proposal to establish differences in the algorithm. Thus, our proposed solution differs from the contributions mentioned in the related work section in that: (1) In the GA algorithm, we use an improved fitness function to value POIs, the original fitness only defines a path, but our fitness chooses the better POIs; additionally, we have exposed in which part of the optimization algorithm all constraints specified by both the user and the context of the visited a tourist destination can be included; (2) we deal with real POIs information with their real constraints, which is obtained from the Internet; this allows the system to generate recommendations for any tourist destination in the world; (3) the user's interests are considered for filtering the POIs; and (4) our itineraries offer recommendations adapted to human behaviours (i.e. restaurants only at lunch times).

VIII. CONCLUSIONS AND FUTURE WORK

The use of mobile technologies in the tourist industry has grown significantly in the last few years. In the future, innovative approaches led by research and using new technologies will allow the development of new ways to manage and market competitive destinations for the benefit of travelers. Therefore, tourists will find recommenders to be an interesting option to solve the problem of planning their itinerary for visiting a tourist destination over a few days.

To achieve better contributions in the tourist environment, data sources that allow the visualization of a continuously changing environment are necessary. Therefore, in addition to the contribution that Google makes through geolocation of sites on maps, studies such as those presented by Padrón-Ávila et al. [2] should be considered.

In this work, we have presented an approach that seeks to improve tourists' experience by optimizing POIs and itineraries. Our solution does not only look for a good itinerary, but also one that maximizes tourists' experiences according to the interests of individual travelers. For this, we took into account several parameters: the opening times of POIs, the suitability of the visit to certain POIs according to the time of day, the time required to get from one POI to another and the time required to visit a POI. Moreover, all the information about POIs was obtained from the Internet rather than being manually collected and prepared in advance. These factors allow our algorithm to produce recommendations that are closer to reality that, can scale without problems and that can adapt to changing environments. However, our recommendations depend on the quality and the availability of the information of the POIs in a given tourist destination, which might be a limitation for several destinations. Our evaluation shows that the POIs proposed by our recommender system helps tourists enjoy the context of the city, without neglecting their personal interests.

An important pending issue for the future will be to enable the application to adapt to unforeseen events that occur during the visit. For example, a tourist may be delayed between two places because a street is under construction or because a POI has been closed without previous notice. Although the analysis of the tourist interface was not our goal. From the results of the evaluation, in future works, we will consider other options to improve how we present the recommendations. We will focus some of our efforts for improving the presentation of restaurants. And we will analyze the interface to reduce the mental effort made by tourists when analyzing the user interface.

As confirmed by Padrón-Ávila et al. [2], tourism analysis helps countries identify sources of income, and better distribute the wealth generated by tourism. It is an area that requires a great deal of work in the future, such as the following:(1) Users: Studying the user and understanding their interests is the first level of study, while identifying the user and understanding their restrictions is the second step. Finally, the third step is analyzing the user in such a way as to identify the benefits and difficulties associated with different places; (2) Tourism companies: Analyzing the information that their users' comments allows the analysis of specific places, and the ability to identify places to visit; (3) Large companies like Google, Microsoft and others offer results that relate to points 1 and 2. The job of a researcher is to take that information and process it appropriately. We have already worked with information management using big data techniques and natural language analysis. We have used tourist surveys, data analysis on the web and POI information generated by Google. Therefore, our future work will consist of generating better data using all the proposals commented on in this research.

REFERENCES

[1] M. Kenteris, D. Gavalas, and D. Economou, "An innovative mobile electronic tourist guide application," Personal and ubiquitous computing, vol. 13, no. 2, pp. 103–118, 2009.

[2] H. Padrón-Ávila and R. Hernández-Martín, "How can researchers track tourists? a bibliometric content analysis of tourist tracking techniques," 2020.

[3] M. Tenemaza, L.-A. Edison, M. Peñafiel, Z. Juan, A. de Antonio, and J. Ramirez, "Identifying touristic interest using big data techniques," in Proceedings of the International Conference on Applied Human Factors and Ergonomics. Springer, 2019, pp. 169–178.

[4] P. Vansteenwegen, W. Souffriau, G. V. Berghe, and D. Van Oudheusden, "The city trip planner: an expert system for tourists," Expert Systems with Applications, vol. 38, no. 6, pp. 6540–6546, 2011.

[5] D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. Pantziou, "A survey on algorithmic approaches for solving tourist trip design problems," Journal of Heuristics, vol. 20, no. 3, pp. 291–328, 2014.

[6] W. Souffriau, P. Vansteenwegen, J. Vertommen, G. V. Berghe, and D. V. Oudheusden, "A personalized tourist trip design algorithm for mobile tourist guides," Applied Artificial Intelligence, vol. 22, no. 10, pp. 964–985, 2008.

[7] C.-Y. Tsai and S.-H. Chung, "A personalized route recommendation service for theme parks using rfid information and tourist behavior," Decision Support Systems, vol. 52, no. 2, pp. 514–527, 2012.

[8] J.-H. Chen, K.-M. Chao, and N. Shah, "Hybrid recommendation system for tourism," in Proceedings of the IEEE 10th International Conference on e-Business Engineering. IEEE, 2013, pp. 156–161.

[9] R. Schaller and D. Elsweiler, "Itinerary recommenders: how do users customize their routes and what can we learn from them?" in Proceedings of the 5th information interaction in context symposium. ACM, 2014, pp. 185–194.

[10] H.-S. Chiang and T.-C. Huang, "User-adapted travel planning system for personalized schedule recommendation," Information Fusion, vol. 21, pp. 3–17, 2015.

[11] D. Gavalas, V. Kasapakis, C. Konstantopoulos, G. Pantziou, N. Vathis, and C. Zaroliagis, "The ecompass multimodal tourist tour planner," Expert Systems with Applications, vol. 42, no. 21, pp. 7303–7316, 2015.

[12] D. Herzog, "Recommending a sequence of points of interest to a group of users in a mobile context," in Proceedings of the Eleventh ACM Conference on Recommender Systems. ACM, 2017, pp. 402–406.

[13] I. Cenamor, T. de la Rosa, S. Núñez, and D. Borrajo, "Planning for tourism routes using social networks," Expert Systems with Applications, vol. 69, pp. 1–9, 2017.

[14] W. Zheng, Z. Liao, and J. Qin, "Using a four-step heuristic algorithm to design personalized day tour route within a tourist attraction," Tourism Management, vol. 62, pp. 335–349, 2017.

[15] S. Ahmadian, M. Meghdadi, and M. Afsharchi, "Incorporating reliable virtual ratings into social recommendation systems," Applied Intelligence, vol. 48, no. 11, pp. 4448–4469, 2018.

[16] S. Ahmadian, N. Joorabloo, M. Jalili, M. Meghdadi, M. Afsharchi, and Y. Ren, "A temporal clustering approach for social recommender systems," in 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). IEEE, 2018, pp. 1139–1144.

[17] E. Tarantino, I. De Falco, and U. Scafuri, "A mobile personalized tourist guide and its user evaluation," Information Technology & Tourism, pp. 1–43, 2019.

[18] M. Tenemaza, A. de Antonio, and J. Ramírez, "The user model, vocabulary and logical architecture for adaptive augmented reality," in Proceedings of the Latin American Conference on Human Computer Interaction. ACM, 2015, pp. 1–8.

[19] ——, "The models and their vocabulary for the adaptive augmented reality a2r," in Proceedings of the 10th Computing Colombian Conference (10CCC). IEEE, 2015, pp. 180–188.

[20] H. A. Rahmani, M. Aliannejadi, S. Ahmadian, M. Baratchi, M. Afsharchi, and F. Crestani, "Lglmf: local geographical based logistic matrix factorization model for poi recommendation," arXiv preprint arXiv:1909.06667, 2019.

[21] S. Ahmadian, M. Afsharchi, and M. Meghdadi, "An effective social recommendation method based on user reputation model and rating profile enhancement," Journal of Information Science, vol. 45, no. 5, pp. 607–642, 2019.

[22] S. T. March and G. F. Smith, "Design and natural science research on information technology," Decision Support Systems, vol. 15, no. 4, pp. 251–266, 1995.

[23] H. A. Simon, The Sciences of the Artificial. MIT Press, 1996.

[24] R. K. Yin, Applications of Case Study Research. Sage, 2011.

[25] P. J. Denning, "A new social contract for research," Communications of the ACM, vol. 40, no. 2, pp. 132–134, 1997.

[26] D. Tsichritzis, "The dynamics of innovation," in Beyond calculation. Springer, 1997, pp. 259–265.

[27] S. Gregor and A. R. Hevner, "Positioning and presenting design science research for maximum impact," MIS quarterly, vol. 37, no. 2, pp. 337–355, 2013.

[28] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," MIS quarterly, vol. 28, no. 1, pp. 75–105, 2004.

[29] A. R. Hevner, "A three cycle view of design science research," Scandinavian Journal of Information Systems, vol. 19, no. 2, p. 4, 2007.

[30] N. Ganganath, C.-T. Cheng, and K. T. Chi, "Data clustering with cluster size constraints using a modified k-means algorithm," in Proceedings of the International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery. IEEE, 2014, pp. 158–161.

[31] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," Journal of the Royal Statistical Society. Series C (Applied Statistics), vol. 28, no. 1, pp. 100–108, 1979.

[32] R. Cheng, Genetic algorithms and engineering optimization. Wiley-Interscience, 2000.

[33] P. Fleming and A. Chipperfield, Genetic Algorithms in Engineering Systems. The Institution of Engineering and Technology, 1997, vol. 55.

[34] O. Kramer, Genetic Algorithm Essentials. Springer, 2017, vol. 679.

[35] S. Sumathi, T. Hamsapriya, and P. Surekha, Evolutionary intelligence: an introduction to theory and applications with Matlab. Springer Science & Business Media, 2008.

[36] A. Lipowski and D. Lipowska, "Roulette-wheel selection via stochastic acceptance," Physica A: Statistical Mechanics and its Applications, vol. 391, no. 6, pp. 2193–2196, 2012.

[37] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," 1988.

[38] J. Grefenstette, R. Gopal, B. Rosmaita, and D. Van Gucht, "Genetic algorithms for the traveling salesman problem," in Proceedings of the first International Conference on Genetic Algorithms and their Applications, vol. 160, no. 168. Lawrence Erlbaum Associates, 1985, pp. 160–168.

[39] M. D. Ekstrand, F. M. Harper, M. C. Willemsen, and J. A. Konstan, "User perception of differences in recommender algorithms," in Proceedings of the 8th ACM Conference on Recommender systems. ACM, 2014, pp. 161–168.

[40] E. M. Voorhees, D. K. Harman et al., TREC: Experiment and Evaluation in Information Retrieval. MIT press Cambridge, 2005.

[41] S. Sivanandam and S. Deepa, "Genetic algorithms," in Introduction to Genetic Algorithms. Springer, 2008, pp. 15–37.

MARITZOL.TENEMAZA was born in Ecuador in 1961. She is currently pursuing at Ph.D. degree in Software, systems and computing in the Universidad Politécnica de Madrid - Madrid. She is an engineer in Computer Systems and has a Computer Science degree from the Universidad de las Américas en Quito - Ecuador, and a Master's degree in Software and Systems from the Universidad Politécnica de Madrid - Madrid.

She is currently working as Professor at the Escuela Politécnica Nacional of Quito- Ecuador, where she researches the human-computing interaction, focused on the study of user interests. Her research interests include Recommender systems and Intelligent systems.

SERGIO LUJÁN MORA was born in Alicante, Spain, in 1974. He received at Ph.D. degree in computer engineering from the Department of Software and Computing Systems, University of Alicante, Alicante, in 2005, and at Computer Science and Engineering degree from the University of Alicante, in 1998, where he is currently a Senior Lecturer with the Department of Software and Computing Systems. He has authored several books and published many articles for various conferences, including ER, UML, and DOLAP, and high-impact journals, including DKE, JCIS, JDBM, JECR, JIS, JWE, IJEE, and UAIS. His main research interests include data warehousing, Web development, Web accessibility, and usability. In recent years, he has focused on e-learning, massive open online courses (MOOCs), open educational resources (OERs), and the accessibility of video games.

ANGÉLICA DE ANTONIO has been a faculty member of the Escuela Técnica Superior de Ingeniería Informática (ETSIINF) at Universidad Politécnica de Madrid (UPM) since 1990. She received a Ph.D. in Computer Science in 1994. Angelica has been Director of the MadHCI Lab of the UPM since 1995, where she has lead several R&D projects in the areas of Intelligent Tutoring Systems, E-Learning, Virtual Environments and Intelligent Agents.

JAIME RAMÍREZ is an associate professor in the Computer Science School of the UPM. He holds an MSc in Computer Science (UPM, 1996) and a PhD in Computer Science (UPM, 2002). His main research interests include Ontology Engineering, and Adaptive Systems with a special focus on Intelligent Tutoring Systems combined with 3D Virtual Environments.