# A DECISION SUPPORT SYSTEM FOR TRIP TOURISM RECOMMENDATION

PATRIZIA BERALDI[1], ANNARITA DE MAIO[1], FILOMENA OLIVITO[2], GIUSEPPE POTRINO[1], IMMACOLATA STRAFACE[3], ANTONIO VIOLI[4]
[1]Department of Mechanical, Energy and Management Engineering, University of Calabria, Italy
[2]Eurora.s.a.s., Rende, Italy
[3]Valuetech s.r.l., Rende, Italy
[4]University of Sannio, Benevento, Italy

## ABSTRACT

The rapid growth in the use of recommendation systems in the tourism sector is mainly related to the possibility to access updated data deriving from social networks, thus providing more appropriate and personalized suggestions. The paper presents a tourist trip recommendation system that suggests personalized itineraries defined as sequence of Point of Interest (PoI) to visit. The system core integrates two software modules: a neural network and an optimization engine. For every pair user-PoI typology, the neural network provides, on the basis of the analysis of the social media data, a score between 0 and 1. These latter values are then used as input parameters for a routing optimization problem that suggests the itinerary by considering additional restriction, as, for example, time windows, budget and time limitations, specified by the end user. Being a computational demanding problem, the model solution is carried out by applying a heuristic approach that is proven to provide high-quality solution in a limited amount of time.

*Keywords: Social media, neural network, routing problem*

## 1 INTRODUCTION

Nowadays, tourism and technology are becoming more and more interconnected. According to Google, each traveller visits, on average, 22 websites before booking a vacation, and the percentage of those who use mobile technologies to register a flight or hotel is approaching 70%. The rapid digital evolution faced by different sector of economy has deeply influenced the behaviour of tourists and, as a consequence, the touristic offer is required to evolve as well. The term, borrowed by the industry sector, to denote this new trend is Tourism 4.0. The novelty of the paradigm relies on the processing of large amount of data collected by social media networks to create personalized travel experiences. The suggested travel plans are more efficient, safer, ecological and less problematic by optimizing travel times and minimizing costs for travellers.

The aim of the paper is to present a tour planning system developed within an Italian funded R&D project to boost the tourism offer in the Southern Italy. The system exploits user's preferences learned from the social network analysis to suggest personalized and optimized itineraries for visiting a set of Points of Interest (PoIs).

With millions of active users, social media platforms, such as Facebook, Twitter, Instagram and Flickr, have become potential big data sources of individual behaviour, thus creating a tremendous opportunity to gather digital traces. Analysing millions of user footprints, it is possible to extract travel behaviour at a scale unimaginable before (Hendrik and Perdana [1]). A valuable approach to analyse a so huge amount of data is based on the use of data-mining techniques. The basic idea is to derive for a given user a list of possible PoIs together with personalized scores without explicitly asking users about their specific tastes. The system allows overcoming the so-called 'cold start' effect typical of the content-based


ISSN: 2058-8305 (paper format), ISSN: 2058-8313 (online), http://www.witpress.com/journals
DOI: 10.2495/TDI-V5-N1-69-80

recommendation systems (see, for example, Lee et al. [2]) and can be casted into collaborative filtering systems with recommendation process based on ratings of other users who have similar preferences (Claypool et al. [3]).

The use of data-mining techniques is not new in the tourism sector. Among the others, we cite the recent paper by Hasnat and Hasan [4] (see, also, the references therein), where a tailored framework is proposed for understanding out tourists' travel behaviours from social media data. Clustering methods are then used to determine destination choice patterns of tourists. It is worthwhile mentioning that, besides machine learning approaches, online social network data can be analysed and integrated into tourist recommendation system, also by using non-machine learning techniques. Here, we mention the very recent contribution by Persia et al. [5] who propose the use of a social sensing approach to obtain personalized PoI scores to use in the definition of personalized routing plans.

The decision support system proposed in this paper integrates neural network for the user's characterization in terms of selection of PoI categories he/she can prefer with advanced optimization tools to generate efficient and reliable routing plans.

As for this second element is concerned, we note that the optimization engine provides the definition and solution of a routing problem with a special structure. The problem, known in the scientific literature as orienteering problem (OP) (see, for example, Golden et al. [6]), includes time windows regarding the time interval a given PoI can be visited and budget constraints that the decision maker is willing to account for. Since the identification and suggestion of the recommended plan are required to be returned in a very short time, the system is empowered with a specialized heuristic approach.

In summary, the proposed system is based on the design and integration of the following elements:

- a neural network approach for the selection of PoI categories the user can be more interested to;
- a clustering approach used to cast the set of PoIs that can be potentially interesting for the user into subsets of PoIs which are close according to some criteria;
- a tailored algorithm for generating high-quality tourist routes in a limited amount of time.

The system has been developed and tested within the 'ASSD – Acceleratori semantici social driven per la generazione di itinerari turistici' project (ASSD), funded by Calabria region, in the South of Italy.

The rest of the paper is organized as follows: Section 2 describes the main functionalities and building blocks of the decision support system, while Section 3 is devoted to the introduction of the methodological kernel. Section 4 presents and analyses the results in terms of effectiveness and applicability on a real-life context. Conclusions and future research directions are highlighted in Section 5.

## 2 THE ASSD DECISION SUPPORT SYSTEM

The final output of the research project ASSD is the design and implementation of a decision support system for the definition of itineraries visiting sets of PoI that can match the tourist implicit and explicit preferences. The system, accessible via both web and mobile applications, has a high-level architecture composed of three main parts. The client side is devoted to the interaction with the end user, while the persistence side contains the modules providing the connection to the data warehouse for data persistence. The main methodological modules are included in the server side, which has been articulated in several components, as depicted in Fig. 1.

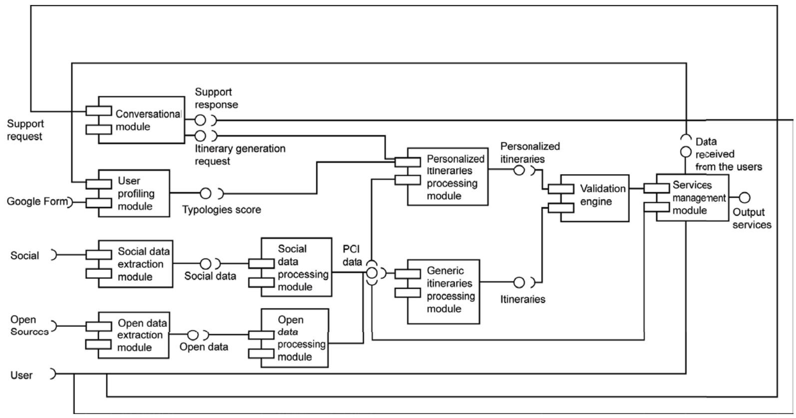


Figure 1: Server-side functional components.

Here, there are some modules for input definition, for both user and PoI characterization. First, a user-profiling module collects individual preferences and the specific parameters for a single request. Moreover, there are modules for data extraction from both open data sources and social media and for the processing of information in order to update PoI features, in terms of ranking, costs, time window and so on. These modules provide input for the 'decision' modules where different tourist itineraries are defined. In particular, in addition to classic 'personalized' itineraries, defined on the basis of the user request, 'generic' routes, that are based on average preferences of all the users in the last month, are generated as well. Finally, a validation engine processes the solutions, and a service management module transforms data into responses and coordinates the communication with the client for the output visualization.

The decision problem at the kernel of the system consists in the definition of tourist routes, i.e. sequences of PoIs to visit, that match better with the user preferences. In order to effectively model such a problem, some assumptions have been made. Each PoI is assumed to belong to a specific category, while each category refers to an area of interest. For example, an art gallery belongs to category 'Museum' of area of interest 'Art'.

Note that, actually, the categories inserted in the database include only cultural, natural and enjoying attraction, while there is not a specific category for the food places. A category related to special events is also present in the system (concert, temporary view and so on), while data related to the congestion, traffic and affluence are not included. These choices are due to the scarcity of open data sources that include the information. Nevertheless, this is a piloting version of the system, ready for the experimentation in the Calabria territory (a region in the South of Italy), but the flexibility of the system allows steady state to add categories and modules at the main database, so far as sufficient open data sources would be available to populate the PoI data set.

The decision support process that is implemented on the system can be articulated into the following steps:

- user preference profiling;
- PoI ranking update;
- route definition.

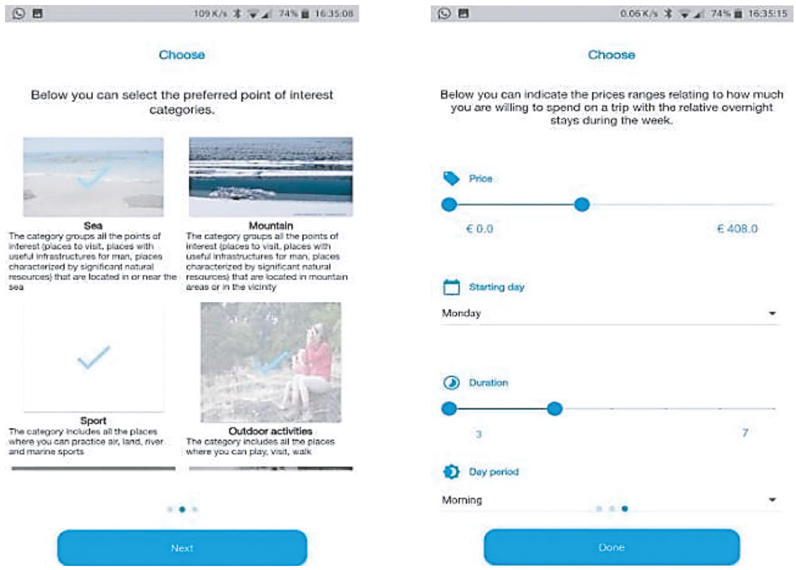


Figure 2: User interfaces for preference set-up.

As regards the first step is concerned, when the user accesses the system and sets/updates some configuration parameters and the areas of interest (e.g. Art, Cultural Heritage, Music, etc.), as shown in Fig. 2, his/her preferences are updated by using a machine learning procedure.

In particular, a score in the range [0, 1] for each pair {user-PoI category} is assigned, so to help the selection for the tourist itinerary of PoIs of categories belonging to user's areas of interest. The procedure is based on a *feed-forward* neural network, whose training is periodically performed based on a database of surveys, by means of a Resilient Backpropagation training algorithm. The neural network has an input layer, which consists of the following attributes:

- residence area;
- age;
- gender;
- educational qualification;
- current job;
- travel team;
- journey time;
- daily distance.

The neural network presents also a certain number of hidden layers, that are updated when the training phase is perfomed and the output layer, that implements a Sigmoidal activation function for computing the final score.

The PoI ranking step is performed periodically by means of the analysis of the presence of the PoI on the main social media (Instagram, Facebook, Twitter and Google Plus) and on other specific websites like TripAdvisor. A specific procedure collects and analyses the posts that refer to the PoI and considers for the previous week the following data:

- the number of posts;
- the average ‘sentiment’ (positive, negative, neutral) expressed on the PoI;
- the level of engagement, that is, the number of interactions of social media users about the posts.

By the analysis of a cumulative function of these data for all the PoI of the same category, the score of each PoI, and then the ranking within the category, is updated.

PoI scores are used in the definition of the routes proposed to the tourist. Such a phase is further articulated into three steps:

1. PoI selection, performed according to the length of the visit imposed by the user; indeed, a number *n* of PoIs with the higher score within the categories the user prefers are selected;
2. PoI clustering, where the *n* PoI are clustered into subsets that can be visited in one day according to the visit time and budget of each PoI;
3. Route definition, where, for each PoI cluster, the route that maximizes a certain score function and satisfies further constraints, like time windows for example, is defined, if any.

From a computational standpoint, the latter step is more critical since it has to be performed within a limited amount of time. For this reason, specific heuristic procedures have been designed and implemented for both the clustering and the routing steps.

In Section 3, the methodological approaches adopted for these procedures are described in detail.

## 3 TOURIST TOUR DEFINITION

The definition of the suggested tourist tours relies on the formulation and solution of a vehicle routing problem, known, in the scientific literature, as OP (Golden et al. [6]). The general model is used in many application domains where the selection of points to be visited represents a critical issue. In the tourist sector, the problem is also known as tourist trip design problem (TTDP).

The model is defined starting from the set of nodes that represent the PoIs, each one characterized by a score, a geographical position, an average time required for the visit and a time window, denoting the opening time interval the point can be visited.

Because of the side constraints related, for example, to limitations of time and budget, not all the PoIs could be visited, and the aim is to perform a PoI selection so to maximize the total score. The solution of the mathematical problem returns an itinerary that starts from the initial PoI, visits a subset of PoIs and finally arrives at the ending PoI. The suggested tour satisfies the time windows, length, time and budget constraints. Both TTDP and OP cannot be solved in polynomial time, and for this reason, all existing web and mobile applications are based on the use of efficient heuristic algorithms (Gendreau et al. [7]).

In this case, we describe the algorithm considering the following features. Let $G = (V, A)$ denote a complete directed graph where $V$ is the set of PoIs and $A$ is the set of arcs. We assume that $G$ is a Euclidean graph, so the triangular inequality holds. In the set $V$, index 1 corresponds to the starting point of tour and index $n$ corresponds to the ending point of the tour. At each arc $(i, j) \in A$, three non-negative parameters are associated: $t_{ij}$ denotes the traversing time, $l_{ij}$ is the traversing distance and $c_{ij}$ denotes the corresponding cost. Different parameters are associated with each node: $R_i$ is the score assigned to the PoI, $t_i$ is time necessary to visit the PoI, $c_i$ is price-ticket to visit the PoI, the time window is specified by $S_i$, i.e.

the starting time in which the PoI can be visited, and $E_i$ is the ending time in which the PoI can be visited. We also consider $L$ as maximum distance of the route, $B$ as the maximum budget for the route and T as the maximum time for the route. The score of each PoI is pre-computed through the neural network, as described before. In the following, we refer to the formulation described in Ciancio et al. [8], and we concentrate directly on the algorithm description. Note that the number of PoIs included in each tour is not fixed *a priori* by the user that can impose only the maximum duration of the tour and the budget as input parameters to the engine for the personalised creation of the tour. The engine is free to select the best number of PoIs that will insert into a tour considering the maximization of their scores. The PoIs are featured by a medium visit duration and a feasible time window (for example, a museum opens at 9:00 and closes at 18:00, and the medium time for the visit is 2 hours), so it is intuitive that these features affected the feasible scheduling of the PoIs and the order of visit generated by the system engine. For this reason, it is not possible to fix a priori the total number of PoIs.

The computational effort required to solve the model is affected by the number of PoIs and, as a consequence, it can be huge. In order to provide the system with a fast tool to support the users, we design a heuristic algorithm based on a clustering engine aimed at reducing the number of potential PoIs to be evaluated. So the algorithm is divided in two main phases:

- *Phase 1 – Clustering*: in the first phase of the heuristic, a clustering procedure is used to select the more convenient PoIs from the total set provided by the platform. They are chosen according to feasibility requirements related to the maximum length and budget decided by the user for his tour while maximizing the score. The clustering procedure is described in Section 3.1.
- *Phase 2 – Optimization*: in the second phase, the set composed by the PoIs selected in Phase 1 becomes the input set for the model described in Ciancio et al. [8], in order to find the good-quality solution and building the tour, in a short computational time.

In the routing field, the approach to *cluster before-route after* is largely used, and it could be considered very effective for the solution of routing problems with different operative constraints (e.g. time windows, budget, duration, capacity, etc.). Similar approaches have been adopted in different industrial routing problems, with application to real case study. For example, in Beraldi et al. [9], a similar approach is applied to solve the problem of a company specialised in van-sharing for heavy freight, so large instances of a pick-up and delivery problem with time windows were approached with success. In Bertazzi et al. [10], a matheuristic algorithm based on a clustering procedure is used for solving an inventory-routing problem in order to deliver freight to 942 different shops situated in the same urban area. Finally, different authors described the clustered vehicle routing problem, based on the concept that the customers are aggregated on the basis of capacity in clusters that have to be served in a precise order (Baratta et al. [11], Pop et al. [12]).

## 3.1 Clustering algorithm

The aim of the clustering procedure is the pre-selection of a subset of PoIs considered as the most promising ones in terms of maximization of the user satisfaction (measured in terms of score). This set will represent the node set of the graph used to define the optimization problem. The clustering procedure of the PoIs is carried out by a greedy approach designed by taking into account the bounds related to the budget B and length of the route in terms of time, denoted by T. In order to describe the clustering procedure, the following parameters are considered: *ResB* and *ResT*, representing the residual budget and residual time, respectively; *ListPoI*, i.e. the list of the PoIs considered into the clustering procedure. Moreover, we introduce the index $k$ to denote the $k$th cluster and to represent the corresponding PoI set. The starting point is chosen considering the geo-localisation of the user and selecting the nearest PoI contained in *ListPoI*. The procedure is described in the following pseudo-code:

**Initialization**
Set k=1. Select the starting PoI $i$ as the nearest PoI $\in$ *ListPoI* to the user's position. If two or more PoI are located at the same distance, select the index $i = argmax\{R_1,...R_n\}$. Insert $i$ into the first cluster $clus_k$.
Update *ListPoI* = *ListPoI* –{i}, ResB = B – $c_i$ and ResT = T – $t_i$
**While (ListPoI != ø) do**

- Select the PoI $j \in$ *ListPoI* nearest to the last inserted point $i$
- If two or more PoIs are located at the same distance, select the one with the highest score
- If (ResB – $c_j$ >= 0 and ResL – $t_j – t_{ij}$ >= 0, then
  Insert $j$ in $clus_k$
  Update *ListPoI* = *ListPoI* – {j}, ResB = ResB – $c_j$ ResT = ResT – $t_j – t_{ij}$

Else
k = k+1
ResB = B
ResT = T

**End While**
The output of the algorithm is represented by a set of clusters, each containing a subset of potential PoIs to be included in the potential tours. By construction, the first cluster should provide better solutions since the choice of the points to include is carried out considering a set of larger cardinality. It is worthwhile pointing out that the clustering procedure accounts for feasibility in terms of time and budget, but not all the PoIs in the cluster could be visited because of the side constraints of the optimization model (the total distance and the restriction imposed by time windows). However, the solution of the OP on the clustered points clearly requires a shorter computational time and provides good results in terms of time and solution quality trade-off.

## 4 COMPUTATIONAL EXPERIENCE

This section is devoted to the presentation of the ASSD system, mainly focusing on the optimization engine. As illustrative example, we consider a test defined on a set of PoIs situated in a city of the South of Italy: Cosenza. The considered points of attraction belong to four categories: Museums, Parks, Entertainment and Places. The main PoIs' features are reported in Table 1.

We assume that the tourist has indicated the maximum duration of the daily itinerary (8 hours), the maximum budget (€50), the maximum length admitted (150 km) and the departure/arrival point, represented by the Hotel Excelsior located in the town. The connections representing the arcs of the graph used in the mathematical formulation are assumed to be known, and the corresponding costs and times are previously determined.

For solving Phase 1, the clustering procedure is executed in order to generate different sets of PoIs that are feasible from the duration and budget point of view. Figure 3 shows the generated clusters of the PoIs. Considering the time and budget limit, three clusters can be defined, identified into the map through three different colours:

Table 1: Features of the PoIs considered.

| Num. | PoI | Category | Time window | Visiting time | Cost | Ranking |
|---|---|---|---|---|---|---|
| 1 | Galleria Nazionale | Museum | 10:00–18:00 | 1.5 h | Free | 0.8 |
| 2 | Museo Multimed. Da Vinci | Museum | 10:00–20:00 | 1 h | €13 | 0.93 |
| 3 | Musei dei Bretti e degli Enotri | Museum | 9:00–18:30 | 2 h | €4 | 0.6 |
| 4 | Castello Svevo | Museum | 9:30–18:00 | 1.3 h | €4 | 0.75 |
| 5 | Ponte di Calatrava | Place | H24 | 15 min | Free | 0.7 |
| 6 | Teatro rendano | Place | H24 | 15 min | Free | 0.4 |
| 7 | Duomo | Place | 8:00–19:00 | 45 min | Free | 0.75 |
| 8 | Lungo fiume | Place | H24 | 1.5 h | Free | 0.35 |
| 9 | Parco Robinson | Park | H24 | 2 h | Free | 0.6 |
| 10 | Villa Vecchia | Park | H24 | 1 h | Free | 0.7 |
| 11 | Villa Nuova | Park | H24 | 30 min | Free | 0.4 |
| 12 | Corso Mazzini | Entertainment | H2 | 2 h | Free | 0.85 |
| 13 | Metropolis | Entertainment | 9:00–21:00 | 1.5 h | Free | 0.8 |
| 14 | Chiesa S. Domenico | Place | 8:00–18:00 | 0.5 h | €2 | 0.5 |
| 15 | Panino Genuino | Entertainment | 19:00–24:00 | 0.5 h | €10 | 0.7 |
| 16 | Parco Giorcelli | Place | H24 | 0.5 h | Free | 0.2 |
| 17 | Cremeria | Entertainment | 7:00–24:00 | 0.3 h | €2.50 | 0.8 |
| 18 | Museo Scienze Naturali | Museum | 9:00–13:00 | 1 h | Free | 0.5 |
| 19 | Museo del Presente | Museum | 9:00–17:00 | 1 h | Free | 0.3 |
| 20 | Piazza Bilotti | Place | H24 | 0.5 h | Free | 0.3 |

Cluster red = {10, 6, 7, 4, 5, 14, 20, 2, 1}, Cluster blue = {12, 3, 11, 8, 19} and Cluster green = {9, 13, 17, 16, 15, 18}. The clusters are all feasible for generating a tour. We consider only the first one because it represents the best trade-off between distance and score. We obtain the solution with a score of 4.63 that generates a route considering only PoIs {1, 2, 4, 10, 7, 5}. The algorithm suggests that the solution will consume a budget of €21.3. The tour is described in Fig. 4 that also reports the time required to move from one point to the next one.

### 4.1 Focus on the effectiveness of the two-phase approach

This section reports the results of some preliminary computational analyses with the aim of investigating the efficiency of the proposed approach in terms of computational time and quality. The algorithm has been implemented using the software AIMMS 3.4, and all computational experiments have been carried out on a PC equipped with an Intel Core i7-6500U CPU running at 2.50 GHz, with 8 GB of RAM and Windows 10.

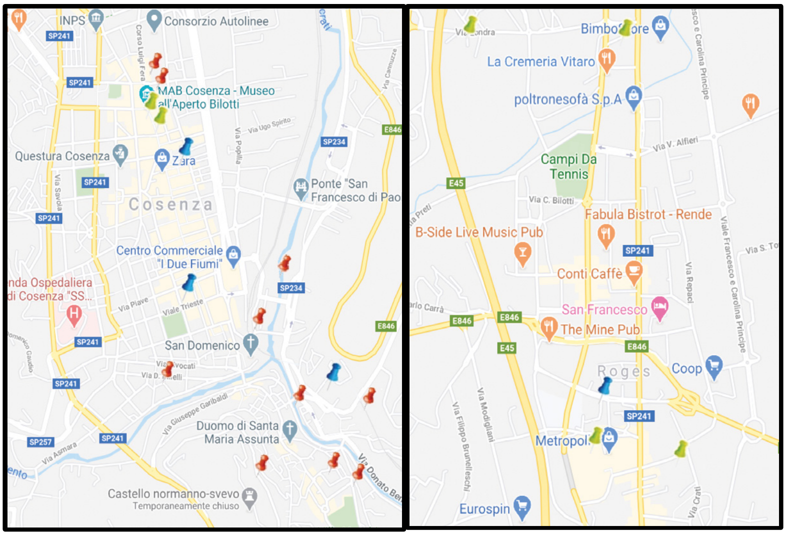


Figure 3: Three clusters generated.

A set of 15 instances have been defined by randomly generating the PoI position and rankings within a particular area and changing different input parameters: the number of available PoIs (range 10–45), the maximum budget (range €100–€200), the maximum length of the tour (range 100–200 km), PoI visit cost (range €5–€25), PoI visit time (1–2 hours), opening time window (range 7–10 a.m.), closing time window (range 18–20 p.m.) and maximum duration of the route equal to 10 hours. Note that we consider only one-day tour into the experiments. These instances have been solved through the pure optimisation model described in Ciancio et al. [8] within a time limit of 1,200 s, which represents an exact approach. Then, the same instances have been solved using the proposed algorithm, in order to compare the performances in terms of computational time and solution quality.

The results are described in Table 2 which is organized as follows: column *Instance* describes the instance name, column reports the number of PoIs considered into the instance, column *Budget* (€) introduces the maximum budget value and column *L* (km) describes the maximum length.

Moreover, column *Sol.Model* introduces the solution value of the exact model, column *Time Model* (s) reports the computational time in seconds for solving the exact model, columns *Sol.Algorithm* and *Time Algorithm* (s) describe the solution value and the computational time obtained through the algorithm and finally the column GAP% represents the gap between the two approaches computed as [(*Sol.Model- Sol.Algorithm)/ Sol.Model]**100.

The results collected in Table 2 shows that the exact model solves to optimality eight instances, while it finds only a feasible solution in seven instances within a time limit of 1,200 seconds, that is a very huge amount of time for an online application. On the other hand, the heuristic finds the optimal solution in eight instances, as the same, and in the other case, it is able to give a feasible solution of good quality, competitive with the model in a far shorter computational time. Indeed, the heuristic is solved in a medium time of around 9 s, and in particular, the clustering phase is approached in less than 3 seconds in all the cases (considering the worst value for the 15 instances), while the exact model usually reaches

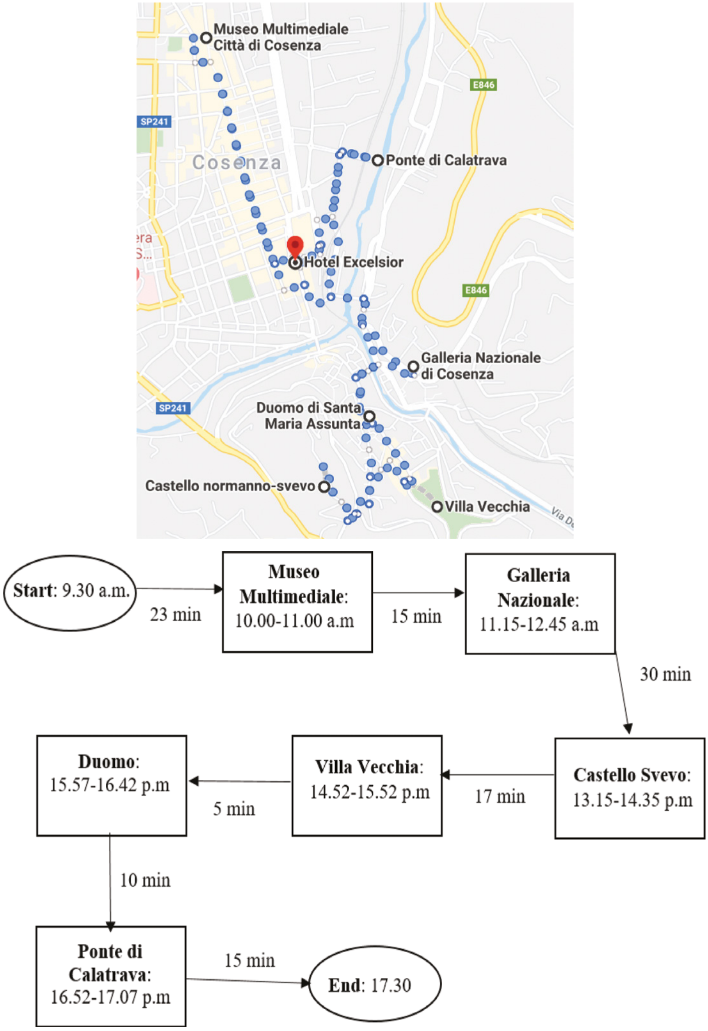


Figure 4: Example of a tour designed with real data.

the time limit for instances with a cardinality of PoIs greater than 25. The saving in terms of computational time using the algorithm is impressive. Furthermore, the quality of the algorithm solution is competitive with the exact model, with a medium GAP of −0.74%. The results demonstrate that the clustering-based approach is very effective in the tourist tour design problem, and it represents the best solution to be implemented inside the online system of the platform described. It is a potential decision support tool for the user that wants to improve his/her tour and having personalised suggestions by the platform in a short computational time.

Table 2: Comparison between algorithm and exact approach.

| Instance | Parameters | | | Results | | | | |
|---|---|---|---|---|---|---|---|---|
| | $N_{POI}$ | Budget (€) | L (km) | Sol. model | Time model (s) | Sol. Alg. | Time Alg. (s) | GAP% |
| Instance1 | 10 | 104 | 147 | 0.97* | 2 | 0.97 | 2.4 | 0.00 |
| Instance2 | 10 | 130 | 100 | 2.5* | 3 | 2.5 | 3.2 | 0.00 |
| Instance3 | 15 | 130 | 100 | 2.33* | 3 | 2.33 | 3.6 | 0.00 |
| Instance4 | 15 | 104 | 147 | 1.87* | 1 | 1.87 | 2.4 | 0.00 |
| Instance5 | 20 | 104 | 147 | 1.88* | 49 | 1.88 | 37 | 0.00 |
| Instance6 | 20 | 130 | 100 | 1.45* | 135 | 1.45 | 54 | 0.00 |
| Instance7 | 25 | 150 | 150 | 1.64* | 41 | 1.64 | 18 | 0.00 |
| Instance8 | 25 | 150 | 160 | 1.84 | 1,200 | 1.85 | 3.2 | 0.00 |
| Instance9 | 30 | 150 | 160 | 1.85 | 1,200 | 1.78 | 1.2 | 4.31 |
| Instance10 | 30 | 130 | 160 | 1.95* | 1,111 | 1.60 | 4.3 | 17.99 |
| Instance11 | 35 | 130 | 160 | 1.83 | 1,200 | 1.83 | 2.01 | 0.00 |
| Instance12 | 35 | 150 | 180 | 2.56 | 1,200 | 2.56 | 3.5 | 0.00 |
| Instance 13 | 40 | 150 | 180 | 1.99 | 1,200 | 2.64 | 1.03 | –32.41 |
| Instance14 | 40 | 130 | 180 | 1.95 | 1,200 | 1.95 | 1.01 | 0.00 |
| Instance15 | 45 | 150 | 180 | 1.93 | 1,200 | 1.95 | 3.02 | –1.04 |
| **Average** | | | | | **649.66** | | **9.32** | **–0.74** |

## 5 CONCLUSIONS

The paper presents a decision support system designed within an Italian funded project for tourist trip recommendation. The system accessible via web and mobile applications provides itineraries visiting sets of PoI that match the tourist's implicit and explicit preferences. The system core integrates two software modules: a neural network and an optimization engine. For every pair {user-PoI typology}, the neural network provides a score that is an important component for the generation of personalised tours. This generation is made up through an optimization problem that suggests the tourist itinerary by considering additional restriction, as, for example, time windows, budget, length and time limitations. Being a computational demanding problem, the model solution is carried out by applying a heuristic solution that is proven to provides high-quality solution in a limited amount of time.

The tools built for the ASSD project could be considered very effective for approaching a so complex engine that is able to model preferences and use in a good way the information and the huge amount of PoIs existing in reality, for really supporting the tourist in the construction of the best travel he/she can desire. After the piloting phase, the system can be improved from a business point of view, easily including other modules, PoI categories, functionalities and information to enrich the service provided to the user, also taking into account the specific context of each region.

ACKNOWLEDGEMENTS

This work has been partially supported by POR Calabria FESR-FSE 2014-2020, with the grant for research project 'ASSD – Acceleratori semantici social driven per la generazione di itinerari turistici', CUP J28C17000370006).

REFERENCES

[1] Hendrik, H. & Perdana, D.H.F., Trip guidance: a linked data based mobile tourists guide. *Advanced Science Letters*, **20**, pp. 75–79, 2014.

[2] Lee, C.-H., Kim, Y.-H. & Rhee, P.-K., Web personalization expert with combining collaborative filtering and association rule mining technique. *Expert Systems and Applications*, **21**(3), pp. 131–137, 2001.

[3] Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., & Sartin, M. Combining Content-based and Collaborative Filters in an Online Newspaper. *ACM SIGIR workshop on recommender systems*. Berkeley, CA: ACM Press, 1999.

[4] Hasnat, M. & Hasan, S., Identifying tourists and analysing spatial patterns of their destinations from location-based social media data. *Transportation Research Part C: Emerging Technologies*, **96**, pp. 38-54, 2018.

[5] Persia,F,, Pilato, G., Ge, M., Bolzoni, P., D'Auria, D. & Helmer, S., Improving orienteering-based tourist trip planning with social sensing. *Future Generation Computer Systems*, https://doi.org/10.1016/j.future.2019.10.028, 2019.

[6] Golden, B.L., Levy, L., & Vohra, R., The orienteering problem. *Naval Research Logistics*, **34**(3), pp. 307–318, 1987.

[7] Gendreau, M., Laporte, G. & Semet, F., A tabu search heuristic for the undirected selective travelling salesman problem. *European Journal of Operational Research*, **106**(2), pp. 539–545, 1998.

[8] Ciancio, C., De Maio, A., Laganà, D., Santoro, F. & Violi, A., A Genetic Algorithm Framework for the Orienteering Problem with Time Windows. *New Trends in Emerging Complex Real Life Problems*, AIRO Springer Series, pp. 179–188, 2018.

[9] Beraldi, P., De Maio, A., Laganà, D. & Violi, A., A pick-up and delivery problem for logistics e-marketplace services. *Optimization Letters*, https://doi.org/10.1007/s11590-019-01472-3, 2019.

[10] Bertazzi, L., Coelho, L.C., De Maio, A. & Laganà, D., A matheuristic algorithm for the multi-depot inventory routing problem. *Transportation Research Part E: Logistics and Transportation Review*, **122**, pp. 524-544, 2019.

[11] Battarra, M., Erdoğan, G. & Vigo, D., Exact Algorithms for the Clustered Vehicle Routing Problem, *Operation Research*, **62**(1), pp. 58-71, 2014.

[12] Pop, P.C., Fuksz, L., Marc, A.H. & Sabo, C., A novel two-level optimization approach for clustered vehicle routing problem. *Computers & Industrial Engineering*, **115**, pp. 304-318, 2018.