Expert Systems with Applications 38 (2011) 6540–6546

Contents lists available at ScienceDirect

Expert Systems with Applications

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

The City Trip Planner: An expert system for tourists

Pieter Vansteenwegen [a,c,*], Wouter Souffriau [a,b], Greet Vanden Berghe [b], Dirk Van Oudheusden [a]

[a] Katholieke Universiteit Leuven, Centre for Industrial Management, Celestijnenlaan 300A, 3001 Leuven (Heverlee), Belgium
[b] KaHo St.-Lieven, Information Technology, Gebr. Desmetstraat 1, 9000 Ghent, Belgium
[c] Ghent University, Department of Industrial Management, Technologiecampus 903, 9052 Zwijnaarde (Gent), Belgium

ARTICLE INFO

Keywords:
Personalised tourist guide
Orienteering Problem
Greedy Randomised Adaptive Search Procedure

ABSTRACT

Over the last few years, advanced digital applications have become available to tourists. Some of these offer the possibility of creating personalised routes. This paper introduces a tourist expert system, called the City Trip Planner, that allows planning routes for five cities in Belgium. It is implemented as a web application that takes into account the interests and trip constraints of the user and matches these to a database of locations in order to predict personal interests. A fast and effective planning algorithm provides an on-the-fly suggestion of a personal trip for a requested number of days, taking into account opening hours of attractions and time for a (lunch) break. The expert system is discussed in detail. Usage statistics and user feedback demonstrate that it is highly appreciated by tourists.

1. Introduction

If a tourist wants to visit every tourist attraction during a city trip to large cities such as Paris, Barcelona or London, a considerable amount of time is required. However, most tourists have a limited amount of time and budget available and thus have to make a selection of the most interesting places to visit. Therefore, most tourists gather information from different sources about the different POIs. Then, they make a selection of the POIs to visit and plan a route between them, taking into account the available time and the opening hours of the different POIs. When carried out properly, this is a rather complex and time consuming process.

The personalised electronic tourist guide is a device that addresses this problem by suggesting, at very short notice, a (near) optimal selection of POIs and a route between them, taking into account the weather, opening hours, crowded places and personal preferences. The underlying selection and routing problem is called the tourist trip design problem (Vansteenwegen & Van Oudheusden, 2007). The application introduced in this paper is based on this problem.

The Dynamic Tour Guide (ten Hagen, Kramer, Hermkes, Schumann, & Mueller, 2005) was the first application to calculate personal tourist trips on-the-fly. In order to construct a tour, the system uses a branch-and-bound algorithm to connect so-called Tour Building Blocks, while maximising Interest Matching Points that reflect the user's personal interest. Souffriau, Vansteenwegen, Vertommen, Vanden Berghe, and Van Oudheusden (2008) use a metaheuristic approach, namely Guided Local Search, to solve the simplest form of the tourist trip design problem, in which a plan has to be devised by only taking the maximum allowed distance into account. They clearly outperform the Dynamic Tour Guide system for problem instances with over 50 POIs. Souffriau, Maervoet, Vansteenwegen, Vanden Berghe, and Van Oudheusden (2009) extended their approach by taking opening hours of POIs into account, for one day, and examined the feasibility of the planning algorithm on a mobile device, with limited computational resources.

Castillo et al. (2008) and Lee, Chang, and Wang (2009) both present a multi-agent based system for planning tourist visits. Castillo et al. (2008) use Case Based Reasoning and the k-Nearest Neighbour algorithm (Dasarathy, 1991) to compare the user's interest to the available activities and to the interests of similar users. Their planner agent takes interesting activities as input and calculates a feasible plan using predicate logic. The context decision agent of Lee et al. (2009) first finds concepts of the ontology that match the tourist's requirements. Next, their travel route recommendation agent uses fuzzy logic to select and sort a top three of historic sites and a top five of local gourmet food stores. A travelling salesperson problem, that consists of these eight locations, is solved by an ant colony optimisation algorithm (Dorigo & Gambardella, 1997), which results in a personal travel route. In both approaches, selection and routing are clearly distinct.

This paper presents the City Trip Planner, which is a web-based tourist expert system that proposes city trips tailored to the user's context and personal interests. The system plans city visits of multiple days, with for each POI multiple time windows which

* Corresponding author at: Ghent University, Department of Industrial Management, Technologiecampus 903, 9052 Zwijnaarde (Gent), Belgium. Tel.: +32 16 322760; fax: +32 16 322986.

E-mail addresses: pieter.vansteenwegen@cib.kuleuven.be (P. Vansteenwegen), wouter.souffriau@kahosl.be (W. Souffriau), greet.vandenberghe@kahosl.be (G.V. Berghe), dirk.vanoudheusden@cib.kuleuven.be (D.V. Oudheusden).

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.11.085

can differ from day-to-day. Moreover, lunch breaks can also be scheduled and the city's tourist office can suggest a few POIs to be included in a trip. The City Trip Planner integrates the selection of POIs and the routing between them. To the best of the authors' knowledge, such an integrated system is unique.

The City Trip Planner expert system suggests personal trips by first capturing the tourist's trip constraints and interests through a small questionnaire. These interests are used to predict a personal interest score for each POI that a tourist can visit within the constraints. The GPS coordinates and opening hours of the POIs, the personal interest scores, the tourist's location and available time give rise to a specific instance of a "tourist trip design problem", which is a difficult combinatorial optimisation problem (Vansteenwegen & Van Oudheusden, 2007). Next, a heuristic algorithm solves this instance, resulting in a personal trip, that is tailored to the user's interests, current location, destination, available time and the opening hours. Finally, the trajectory can be printed or downloaded to a GPS navigation device. Fig. 1 overviews the different components of the system.

The remaining of the paper is structured as follows. Section 2 describes the different components of the tourist expert system, provides a mathematical model for the tourist trip problem and proposes a solution approach. Section 3 describes the implementation of the system. Section 4 provides user statistics and evaluates the system. Finally, Section 5 concludes the paper and points out directions for further research.

2. City Trip Planner

Section 2.1 describes the structure of the tourist POI database, Section 2.2 describes how to synthesise personal tourist information into the user profile. Next, Section 2.3 matches this user profile to the pre-determined POI database in order to predict personal interest scores for the different POIs, making the formulation of a tourist trip design problem instance possible. The algorithm presented in Section 2.4 solves this instance and proposes a trip to the user. When the tourist is happy with the proposal, the trip can be finalised, as is explained in Section 2.5.

2.1. POI database

The expert system contains a spatially enabled database with all the POIs of five historic cities in the region of Flanders, Belgium. The city of Antwerp has 170 POIs, Bruges has 143 POIs, Ghent 216, Leuven 83 and Mechlin 109. Every POI is characterised by its GPS coordinates, a calendar of opening and closing hours, an average visiting duration and textual descriptions in English and Dutch. A POI belongs to exactly one "type": abbeys, beguinages, castles, churches, musea, etc. Each city defined at most ten POI types that are very relevant for the city concerned. Moreover, the POIs have been classified manually using the following categories:

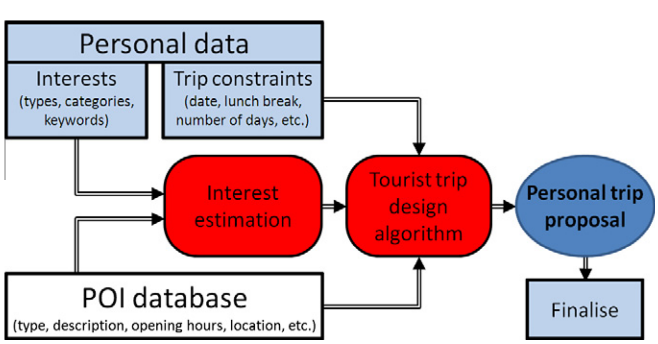


Fig. 1. System overview.

archaeology, architecture, classical art, "local markets and streets", modern art, nature, religious art and science. A system administrator of the local tourist office determined membership degrees to each category: not at all, a bit, mostly or absolutely. The system administrator can mark at most five POIs as "not to be missed". These POIs will always be included in the initial trip that the expert system suggests to the tourist, provided that the time requirements are met.

All the data is provided and kept up-to-date by the tourist offices of the different cities. A web-based authoring tool supports this process.

2.2. Personal data

The tourist first enters a number of trip contraints that restrict the possibilities and chooses one of the five cities to visit, the date of arrival and the number of days the trip will last. Next, for each day of the trip, the tourist chooses a starting and ending location, and a start and end time. The starting and ending locations can be chosen from a predefined set of locations, e.g. the railway station or a parking lot. It is also possible to enter any address, e.g. the address of a hotel. The user can request a (lunch) break of a certain duration to be scheduled in an interval during the day. E.g. a tourist prefers to have a 1 h lunch break between 1 and 3 pm. Based on this information, the database is queried and only the set of POIs that can be visited within these trip constraints, based on distance and opening hours, are retrieved.

Next, the system queries the user for his or her interests and creates a user profile. This profile is composed of three parts, namely types, categories and keywords. The tourist states his degree of interest in the different categories mentioned above: not at all, a bit, mostly or absolutely. Degrees of interest should also be expressed for the POI types. Furthermore, the user may refine his profile by means of arbitrary keywords, e.g. football, war, dark ages, etc. All this does not take much more than a minute. Finally, the tourist is offered the possibility to create an account in order to re-use his input for future visits to the web application.

2.3. Interest estimation

After a tourist entered the trip constraints and preferences, the system estimates the personal interest in each POI that can be visited, resulting in a score for each possible visit. This interest score is the sum of three components: the type score, the category score and the keyword search score. Every POI category membership, not at all, a bit, mostly or absolutely, is quantified as 0, 1, 2 and 3, respectively. This is also done for the tourist's type and category interests.

The *type score* of a POI is set equal to three times the interest degree of the tourist for that particular type. If the tourist is not interested in the type, the total score will be set 0, and the POI will be removed from the set of POIs. If a POI has a type score of 9, the POI is a must for the tourist and a bonus of 3 is added to the type score, in order to reinforce its possible selection. This bonus increases the attractiveness of top POIs.

For the sake of clarity, Table 1 presents an example of the interest score calculation of a POI, which is an abbey in this case.

In order to calculate the *category score* of a POI, the numerical degree of membership in each interest category is multiplied by the user's interest for each corresponding category. Similarly, if this product is 9, an additional bonus of 3 is awarded. The *category score* for a POI is the sum of the three highest scoring categories, resulting in a maximum of 36.

The *keyword search score* is calculated using the Vector Space Model of Baeza-Yates and Ribeiro-Neto (1999), an Information Retrieval technique widely applied by search engines. Every full text description in the POI database is preprocessed and indexed, resulting in a document vector. The user formulates interests as a number of arbitrary chosen keywords, which form a query vector. This query vector is compared to the document vector of the corresponding POI. This comparison forms the keyword search score, which is then normalised between 0 and 36. The keyword search scoring procedure is described in detail by Souffriau et al. (2008).

The three scores are added and form the *personal interest score* of the tourist in the POI.

Finally, each POI that has been marked by the city's tourist office as "not to be missed", receives a score equal to the sum of all the POI's interest scores plus one.

2.4. Tourist trip design algorithm

At this stage, a set of POIs with a score, a visiting duration, and a set of multiple opening and closing times per day is available. Combined with the trip constraints of the tourist, this set allows to define an instance of the tourist trip design problem. This problem extends the well-known Team Orienteering Problem with Time Windows (TOP/TW), modelled by Vansteenwegen, Souffriau, Vanden Berghe, and Van Oudheusden (2009). It introduces multiple TWs per POI and TWs that can be different on different days. Moreover, the possibility to schedule a lunch break is added. This break has no fixed location or exact timing. These extensions, significantly increases the complexity of the problem.

The TOP/TW is a difficult combinatorial optimisation problem. Exact solution approaches, e.g. Integer Programming, are not appropriate as they will require substantial computational resources and long execution times (Righini & Salani, 2009). A good suboptimal solution will suffice. Indeed, a small loss in solution quality is insignificant for the application, considering the inherent shortcomings of quantifying a tourist's personal interest in a location. For these reasons, the authors propose a metaheuristic as a solution method.

The proposed solution approach is based on a Greedy Randomised Adaptive Search Procedure (GRASP). This metaheuristic was first introduced by Feo and Resende (1989); it was successfully applied to the Team OP, without time windows, by Souffriau, Vansteenwegen, Vanden Berghe, and Van Oudheusden (2010). Algorithm Listing 1 overviews the GRASP approach for this scheduling problem.

Algorithm 1. GRASP for the Tourist Trip Design Problem

```
while Stopping criterium is not met do
    Solution = empty;
    Greediness = random (0,1);
    VisitList = GeneratePossibleVisits (Solution);
    while VisitList not empty do
        foreach Visit in VisitList do
            Calculate heuristic value;
        end
        Determine threshold value;
        Restrict VisitList;
        RandVisit = random Visit from restricted VisitList;
        Solution = insert RandVisit;
        VisitList = GeneratePossibleVisits (Solution);
    end
end
Return best found;
```

A number of iterations is performed, until the stopping criterium is met. Each iteration starts by drawing a new greediness value from a uniform distribution. This greediness parameter, which lies between 0 and 1, prescribes a precise ratio between greediness and randomness. Next, a list of possible visits is generated from an initial solution which contains only the start and end of each tour, resulting in a list of possible visits, referred to as the Candidate List (CL) in the GRASP terminology. For each possible visit a heuristic value is calculated. A threshold is computed by multiplying the difference between the maximum and minimum heuristic values of the CL by means of the greediness parameter. The original CL is filtered and only members with a heuristic value that exceeds the threshold are collected into the restricted CL. A random visit from this restricted CL is selected and applied to the current solution. Finally, a new CL is generated, based on the new solution. An iteration is terminated when none of the generated visits is valid, implying an empty CL. The algorithm keeps track of the best solution found.

Table 1
Personal interest score calculation example.

		POI	Tourist	Subtotal	Total
Type	Abbey		3		12
Category	Archaeology	3	2	**6**	
	Architecture	2	1	**2**	
	Classical art	1	2	2	
	Local markets	0	0	0	
	Modern art	0	2	0	
	Nature	1	1	1	
	Religious art	3	3	**12**	
	Science	0	1	0	20
Keyword search	"History"				15
Total score					47

In order to take the lunch break into account, the insertion step of Vansteenwegen et al. (2009) is extended, enabling insertions of virtual POIs without locations. The original insertion step of Vansteenwegen et al. (2009) tries to add, one by one, new visits to a tour. Before an extra visit can be inserted in a tour, it should be verified that all visits scheduled after the insertion place still satisfy their time window. A quick evaluation of each possible insert move is required to reduce computational efforts. Checking the feasibility of all other visits would take much time. This can be avoided by recording *Wait* and *MaxShift* for each already included visit. *Wait* is defined as the waiting time in case the *Arrival* at a location takes place before the time window. The service itself can only start when the time window opens. If the arrival takes place during the time window, *Wait* equals zero. *MaxShift* is defined as the maximum time the service completion of a given visit can be delayed, without making any visit infeasible. *MaxShift* of location i is equal to the sum of *Wait* and *MaxShift* of the next location $i + 1$, unless *MaxShift* is limited by the end of its own time window. When a visit is added to the tour, the *Arrival* and *Wait* values of subsequent visits have to be calculated, as well as the *Maxshift* values of all the visits. Normal visits all have a *Duration* and a time window, a location, and travel costs to and from other locations. In order to extend this insertion step for dealing with the lunch break, the standard formulas of Vansteenwegen et al. (2009) are adapted. A lunch break is defined by only a *Duration* and a time window. It does not have a location and thus has no associated travel costs.

For each possible visit insertion, a heuristic value is calculated

$$h_j = \left| \frac{InterestScore_j^2}{Shift_j} \right| \tag{1}$$

where $Shift_j$ is the total time consumption for inserting an extra visit j. By means of this value, the GRASP algorithm selects a visit to insert.

The algorithm then presents a personal trip for multiple days to the tourist. Also, a list of possible interested visits that are not

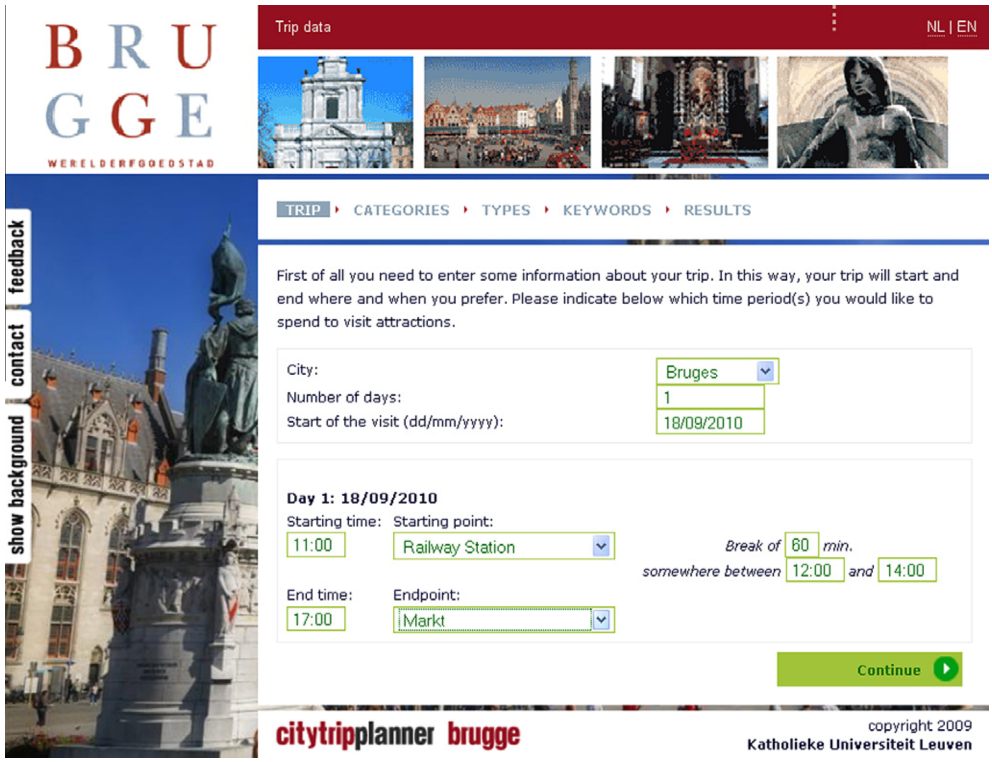


Fig. 2. Trip constraints.

selected due to the trip constraints, is presented. The tourist can mark visits from both the proposal and the interested visits list to be deleted or included and ask for a new proposal. Again, on-the-fly, the new tourist trip design problem is solved and a modified trip, that meets all constraints, is presented.

2.5. Finalise

When after a sequence of proposals and alteration requests, the tourist is satisfied with the latest proposal, the trip can be finalised. The user can print a detailed schedule of the trip, composed of all the visits with their arrival and leaving times, full text descriptions of each POI and detailed maps of the route between the POIs. This information can also be downloaded to a mobile GPS device, allowing the tourist to track his route accurately. Moreover, the final trip can be shared with family and friends using social networking sites.

3. Implementation

The expert system is implemented as a web application. All data is stored in a Postgres database with spatial PostGIS extensions. User interaction is achieved via dynamic web pages using PHP and Java scripting, in an Apache web server. The score prediction and tourist trip design algorithm was coded in Java 1.6 and exposed to the web server via a web service.

In order to demonstrate the different planning steps of the City Trip Planner expert system, a short planning case is discussed and illustrated. A one day trip to Bruges is planned, starting from the railway station and ending in the centre of Bruges at the "Markt". A 60 min lunch break is requested between noon and 2 pm (Fig. 2).

Concerning the categories and POI types, a large interest in "monumental buildings" and "city palaces" is entered together with no interest in "statues" or "city parks" (Fig. 3). The tourist, in this example, has a large interest in "architecture" and "local market and streets" and inputs "music" as a keyword.

The resulting trip proposal is presented in a table (Fig. 4) and on a map (Fig. 5). In this case, the trip contains no statues or city parks, but a number of squares, architectural buildings and city palaces. The POI with the highest score is the "Concert Hall". It is strongly related to the given keyword, to architecture and to monumental buildings. Although only a small interest in beguinages was given, a visit to the "Beguinage-inside" is included, since the local tourist office indicated this visit is a "not to be missed" for a tourist. Obviously, the tourist can still decide to omit this visit from the trip plan.

Fig. 5 presents part of the map of the trip, i.e. the end of the trip and the arrival at the "Markt". After finalising the trip, the user can download or print a complete map with detailed POI information.

4. User satisfaction

This section presents a number of statistics and a summary of a user enquiry.

4.1. Usage statistics

The data is collected by means of Google Analytics[1]; an on-line tool providing traffic statistics to a web site. Data is collected for a period of two months after the public launch of the system (July 7 till September 6, 2009).

17,510 unique visitors used the system 21,498 times, resulting in an average of 1,946 visitors per week or 282 visitors per day. Visits originated from 87 different countries, half of which from Belgium, 20% from the Netherlands, 7% from Spain and 6% from the UK. The United States, Germany, France, Italy, Japan and Canada complete the top ten. 80% of the users visited more than one

[1] www.google.com/analytics.

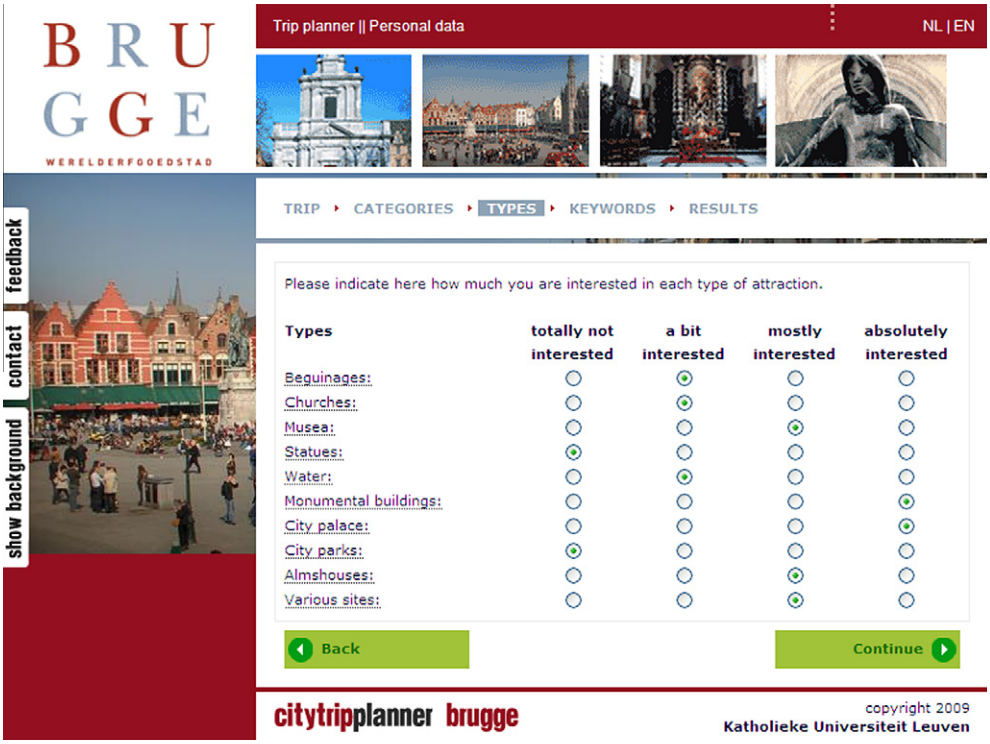
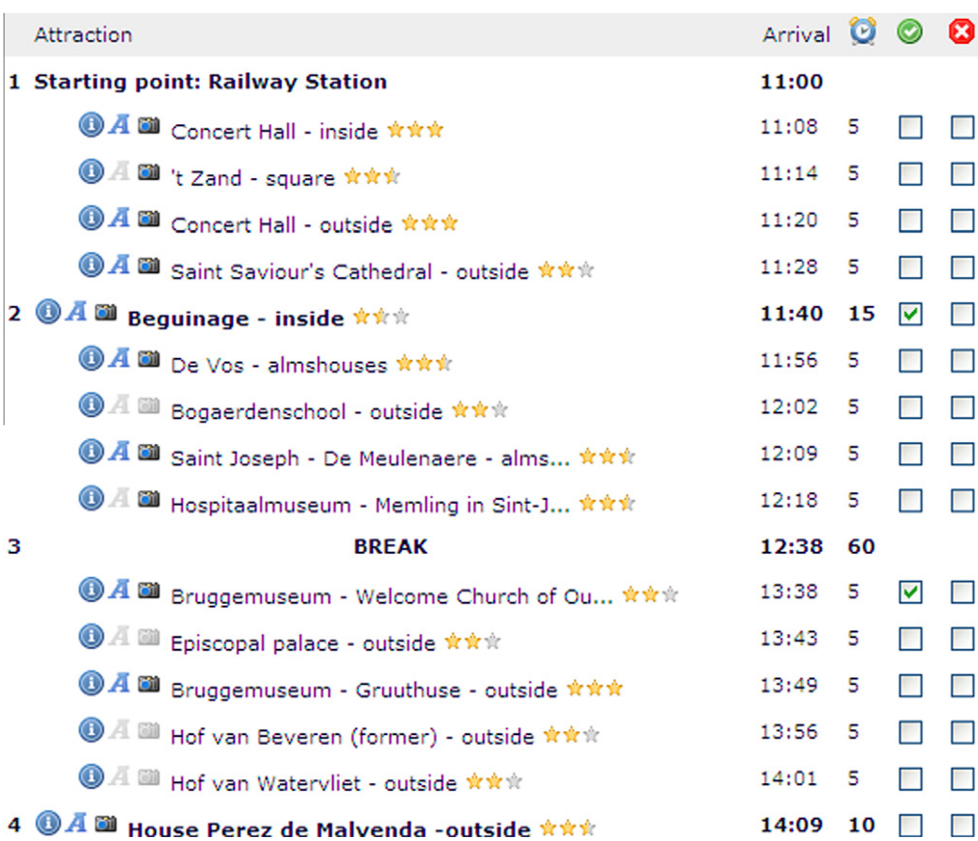


Fig. 3. Interest types.



	Attraction	Arrival			
1	**Starting point: Railway Station**	**11:00**			
	Concert Hall - inside	11:08	5		
	't Zand - square	11:14	5		
	Concert Hall - outside	11:20	5		
	Saint Saviour's Cathedral - outside	11:28	5		
2	**Beguinage - inside**	**11:40**	**15**	✓	
	De Vos - almshouses	11:56	5		
	Bogaerdenschool - outside	12:02	5		
	Saint Joseph - De Meulenaere - alms...	12:09	5		
	Hospitaalmuseum - Memling in Sint-J...	12:18	5		
3	**BREAK**	**12:38**	**60**		
	Bruggemuseum - Welcome Church of Ou...	13:38	5	✓	
	Episcopal palace - outside	13:43	5		
	Bruggemuseum - Gruuthuse - outside	13:49	5		
	Hof van Beveren (former) - outside	13:56	5		
	Hof van Watervliet - outside	14:01	5		
4	**House Perez de Malvenda -outside**	**14:09**	**10**		

Fig. 4. Personal trip proposal-table.

page and can thus be considered as “real” users. Only 3% leaves the site while completing one of the interest measuring forms. Two third of the users starting the trip planning wizard, also complete it. 20% of the tourists receiving a trip suggestion, decided to use

Fig. 5. Personal trip proposal-map.

the print or download functionalities. It is clear that many users just try out the web site or use it to get a first idea about the touristic opportunities of the city. The local tourist offices are enthusiastic about the number of visitors of the expert system.

An average visit took four minutes and 19 s and was composed of 8.07 page views. This is rather high in comparison to other web sites, which experience a lot of accidental visits. Inputting user data takes on average one minute and a half. The user spends two minutes on average checking his personal trip proposal and reading the detailed POI information.

20,395 trips were proposed in total. In descending order, 7,313 were planned in Antwerp, 4,098 in Bruges, 3,335 in Leuven, 2,904 in Ghent and 2,745 in Mechlin. 15,748 were single-day trips, 2,067 were composed out of two days, 1,213 of three days, 289 of four days, 107 of five days, 21 of six days and 143 of a whole week. Averaged over all trips, one day of a trip consists of 26 activities on average, including start, lunch break and end. Many of these activities are only short visits (five minutes or less) to look at a statue or at the facade of a building.

4.2. User feedback

The website allows the users to give feedback using a questionnaire. First, the user can disagree completely with, disagree with, agree with, agree completely with or have no opinion about following statements:

S1 Completing the forms to construct my profile took too much time;
S2 The whole process to receive a personal trip took too much time;
S3 I had to change a lot before receiving a satisfying trip;
S4 The proposed attractions meet my interests;
S5 It was always immediately clear what was expected from me;
S6 The City Trip Planner is very clear and easy to use.

Next, the user is asked for remarks, suggestions for extra functionalities, or other suggestions. During the first two months, 43 visitors have filled in this form, of which 20 reported small technical malfunctions right after the launch. The results of the other 23 are summarised in Table 2.

Table 2
Summary of user feedback.

Statement	Strongly disagree	Disagree	Agree	Strongly agree	No opinion
S1	13	6	1	0	3
S2	13	8	1	0	1
S3	7	8	2	2	4
S4	1	0	15	5	2
S5	3	2	8	9	1
S6	1	3	10	8	1

For a positive evaluation of the website, users should (strongly) disagree with statements 1 to 3 and (strongly) agree with statements 4 to 6. It can be verified in Table 2 that this is the case for the City Trip Planner. Almost all users are (very) positive about the required input (S1), the response time (S2), the quality of the proposal (S3 and S4) and the user friendliness (S5 and S6). It is interesting to note that most people disagree with S1, as Davies, Cheverst, Mitchell, and Efrat (2001) reported that a time-consuming input of preferences is a source of negative reactions when proposing custom-built tours.

Some users give suggestions for extra functionalities such as enabling to save the trip for the next visit or include booking of restaurants and hotels. Others suggest to extend the website to other cities and translate the content to other languages, as it is currently only available in English and Dutch. Overall, apart from the technical malfunctions reported, the remarks concerning the functionality are positive.

5. Conclusions

The City Trip Planner is a web-based tourist expert system that proposes custom-made city trips, tailored to the user's interests and context. The system is implemented as a web site for five historic cities in Flanders, Belgium. The different tourist offices supply and maintain a detailed database containing information about the different points of interest in their city. A tourist states his context and interest, based on which a selection of visits is filtered. The proposed tourist trip design algorithm maximises the tourist's trip satisfaction, by selecting the most interesting visits while respecting all constraints. Usage statistics and user feedback during the first two months clearly indicate that the City Trip Planner is appreciated and proposes high quality trips to tourists. Based on the high number of users of the system and the feedback from the local tourist offices, it can be concluded that tourists are looking for expert systems like this to plan their trip. Extending the system to take into account weather forecasts, budget limitations or public transportation will certainly increase the usefulness.

This web application allows planning multiple day visits in advance. When actually making the trip, unexpected events may make the planning infeasible. A mobile device could overcome this problem since it could provide immediately an adapted plan. The limited computational resources of a mobile device can become a problem. However, the mobile device only has to take the current day into account, and can discard POIs that are planned to be visited on other days. In this way, substantially less computational resources are required to alternate the plan.

Also, the interest estimation system could be extended to incorporate more advanced recommendation techniques, such as collaborative filtering or content-based recommendation (Schiaffino & Amandi, 2009).

Acknowledgement

Pieter Vansteenwegen was a post-doctoral research fellow of the Fonds Wetenschappelijk Onderzoek–Vlaanderen (FWO).

References

Baeza-Yates, R., & Ribeiro-Neto, R. (1999). *Modern information retrieval*. Addison-Wesley.

Castillo, L., Armengol, E., Onaindía, E., Sebastiá, L., González-Boticario, J., Rodríguez, A., et al. (2008). Samap: An user-oriented adaptive system for planning tourist visits. *Expert Systems with Applications, 34*, 1318–1332.

Dasarathy, B. (1991). *Nearest neighbor (NN) norms: NN pattern classification techniques*. IEEE Computer Society Press.

Davies, N., Cheverst, K., Mitchell, K., & Efrat, A. (2001). Using and determining location in a context-sensitive tour guide. *IEEE Computer: Special Issues on Location Aware Computing, 34*, 35–41.

Dorigo, M., & Gambardella, L. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation, 1*, 53–66.

Feo, T., & Resende, M. (1989). A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters, 8*, 67–71.

Lee, C.-S., Chang, Y.-C., & Wang, M.-H. (2009). Ontological recommendation multi-agent for tainan city travel. *Expert Systems with Applications, 36*, 6740–6753.

Righini, G., & Salani, M. (2009). Decremental state space relaxation strategies and initialization heuristics for solving the orienteering problem with time windows with dynamic programming. *Computers and Operations Research, 4*, 1191–1203.

Schiaffino, S., & Amandi, A. (2009). Building an expert travel agent as a software agent. *Expert Systems with Applications, 36*, 1291–1299.

Souffriau, W., Maervoet, J., Vansteenwegen, P., Vanden Berghe, G., & Van Oudheusden, D. (2009). A mobile tourist decision support system for small footprint devices. In Bio-Inspired Systems: Computational and Ambient Intelligence. In *Proceedings of 10th International Work-Conference on Artificial Neural Networks (IWANN2009), Special Session on Advances in AI models adaptable to mobile devices, number 5517 in Lecture Notes in Computer Science* (pp. 1248–1255), Salamanca, Spain. ISBN 978-3-642-02477-1.

Souffriau, W., Vansteenwegen, P., Vanden Berghe, G., & Van Oudheusden, D. (2010). A path relinking approach for the team orienteering problem. *Computers & Operations Research, 37*, 1853–1859.

Souffriau, W., Vansteenwegen, P., Vertommen, J., Vanden Berghe, G., & Van Oudheusden, D. (2008). A personalised tourist trip design algorithm for mobile tourist guides. *Applied Artificial Intelligence, 22*(10), 964–985.

ten Hagen, K., Kramer, R., Hermkes, M., Schumann, B., & Mueller, P. (2005). Semantic matching and heuristic search for a dynamic tour guide. In *Information and Communication Technologies in Tourism*. Springer.

Vansteenwegen, P., & Van Oudheusden, D. (2007). The mobile tourist guide: An OR opportunity. *OR Insight, 20*(3), 21–27.

Vansteenwegen, P., Souffriau, W., Vanden Berghe, G., & Van Oudheusden, D. (2009). Iterated local search for the team orienteering problem with time windows. *Computers & Operations Research, 36*, 3281–3290.