

Personalized Tourist Route Generation

Ander Garcia¹, Olatz Arbelaitz², Maria Teresa Linaza¹,
Pieter Vansteenwegen³, and Wouter Souffriau³

¹ Department of eTourism and Cultural Heritage
Visual Communication Technologies Vicomtech
Paseo Mikeletegi 57, E-20009
Donostia - San Sebastian, Spain
agarcia@vicomtech.org

² Department of Computer Architecture and Technology
University of the Basque Country
Donostia - San Sebastian, Spain

³ Centre for Industrial Management
Katholieke Universiteit Leuven
Leuven, Belgium

Abstract. When tourists are at a destination, they typically search for information in the Local Tourist Organizations. There, the staff determines the profile of the tourists and their restrictions. Combining this information with their up-to-date knowledge about the local attractions and public transportation, they suggest a personalized route for the tourist agenda. Finally, they fine tune up this route to better fit tourists' needs. We present an intelligent routing system to fulfil the same task. We divide this process in three steps: recommendation, route generation and route customization. We focus on the last two steps and analyze them. We model the tourist planning problem, integrating public transportation, as the Time Dependent Team Orienteering Problem with Time Windows (TDTOPTW) and we present an heuristic able to solve it on real-time. Finally, we show the prototype which generates and customizes routes in real-time.

1 Introduction

Nowadays, the generation of personalized tourist routes that take into account the profile of the tourist and up-to-date attraction information is a time consuming task. During the pre-trip stage, tourists have to spend their time collecting information about the destination in order to start planning the stay. This is a difficult task because they have to choose the POIs they are going to visit and estimate visits duration and traveling times.

When tourists are at a destination, this task is often done by the staff of the Local Tourist Organizations (LTOs). The staff determines the profile of the tourists (cultural, romantic, family, etc.) and their restrictions (time, money, etc.). Combining this information with knowledge about local attractions (location, price, timetable, etc.), the LTO suggests a personalized route for the

tourist. Finally, they make small changes to this personalized route based on the feedback from the tourist.

A Personalized Electronic Tourist Guide (PET) should perform at least the same task fulfilled by the LTO, on a hand-held device, providing three main functionalities: recommendation, route generation and customization. A PET should select the most interesting POIs for tourists (recommendation); generate routes in real-time maximizing tourists' satisfaction taking into account different restrictions, attractions' attributes (opening and closing times, visit duration, entrance fee, etc.) and traveling times (route generation); and allow further adaptations to the route (customization).

Furthermore, public transportation information should be integrated, since that is identified as one of the most appreciated functionalities of a PET [1,2,3]. In this paper we integrate public transportation and present a prototype that focuses on the route generation and customization.

In Section 2 we review the state of the art regarding PETs and Operations Research (OR) algorithms. In Section 3 we give an overview of the general functionalities of a PET we are interested in. In Section 4 we focus on the route generation algorithm. In section 5 we introduce the customization functionalities. In Section 6 we show our prototype. Finally in Section 8 we summarize the conclusions and some further work.

2 State of the Art

Vansteenwegen [4] and Souffriau [5] present an extensive review of existing Personalized Electronic Tourist Guides (PETs). However, none of these guides uses advanced heuristics to solve the planning problem nor integrates the use of public transportation. Although Dynamic Tour Guide [6] generated and recalculated routes in real-time (less than five seconds), the routing algorithm was very simple and it had important restrictions: it could only create proper solutions for one day routes and a small number of POIs. A PET called P-Tour [7] applied a genetic algorithm to calculate routes. Nevertheless, tourists had to manually enter the POIs they wanted to visit with their details (visiting time, duration and tourist score). Moreover, the system needed nearly ten seconds to obtain a route for just twelve POIs. Again, no public transportation was taken into account. Castillo et al. [8] described a multiagent based system for planning tourist visits. Although they mentioned transportation, they did not integrate selection and routing nor give hints or details about the implementation or performance. Zenker et al. [9] presented ROSE, a mobile application assisting pedestrians to find preferred events and comfortable public transport connections composed by three main services: recommendation, route generation and navigation. They identified the route planning problem to solve and they described it as a multiple destination recommendation with public transportation and time windows. This is the same problem we propose. However, they did not find any algorithm able to solve it.

The requirements of the problem a PET has to solve can be modeled as the Tourist Trip Design Problem (TTDP) [10,11]. The Orienteering Problem (OP)

[12] is the most basic version of the TTDP. Its generalization to a multiple-day trip is known as the Team OP (TOP). Vansteenwegen et al. [13] present an extensive review of these problems, solution algorithms and applications. When POIs have an associated time window, the problem is called TOP with Time Windows (TOPTW) [14]. Interested readers can find a thorough revision of literature about OPTW and TOPTW in [15,16,17,18].

Fomin and Lingas [19] present the Time Dependent OP (TDOP), an extension of the OP where the time needed to travel from a POI i to a POI j depends on the leave time from i . Implicitly, they can deal with different transportation modes. However, they don't present an algorithm that can be used in real-time applications and no time windows or multiple-day trips are considered.

The model we propose is the Time Dependent TOPTW (TDTOPTW) [20], which combines the previous models and makes possible to integrate one or more public transportation networks to travel between POIs. The travel time required to move from one POI to the next, will vary according to the transportation mode (walking or public transportation) and the leave time. To the authors' understanding, there is no algorithm able to solve the TDTOPTW.

Although examples are present of finding the shortest path in public transportation networks or time dependent networks, they all focus on solving individual queries. Pyrga et al. [21] provide extensive information about the Earliest Arrival Problem (EAP). Ding et al. [22] focused on finding the best departure time to minimize the total travel time on time dependent graphs. A large number of public transportation providers implement these algorithms. For example, the public transportation service provider of San Sebastian has a best path finding application to move between two points (<http://www.dbus.es>).

As examples of more advanced applications, Zografos and Androutsopoulos [23] implemented an algorithm to calculate optimum itineraries in Athens's urban public transport system, including a stop at an intermediate point. In Hong-Kong, a similar system was tested using a different approach [24]. Within the tourism field, PECITAS [25] is a mobile personal navigation system to find the best path between two POIs regarding both the travel time and tourist's preferences. All these applications focus on POI-to-POI cheapest cost problems, while this paper focuses on a higher level selection problem of POIs and a routing problem between multiple POIs.

3 Personalized Electronic Tourist Guide (PET)

Souffriau [5] classifies the different functionalities PETs offer. Regarding the route generation, they can be gathered in three main basic functionalities (Figure 1). In Figure 1 we also represent the inputs and outputs of each step.

- **Recommendation.** Combining the tourist information of the destination with the tourist profile, the system has to create a list of recommended POIs. As a result, POIs will have a different score and visit duration for different tourists. Interested readers will find a deep review on tourist recommendation systems on the recent paper from Kabassi [26].

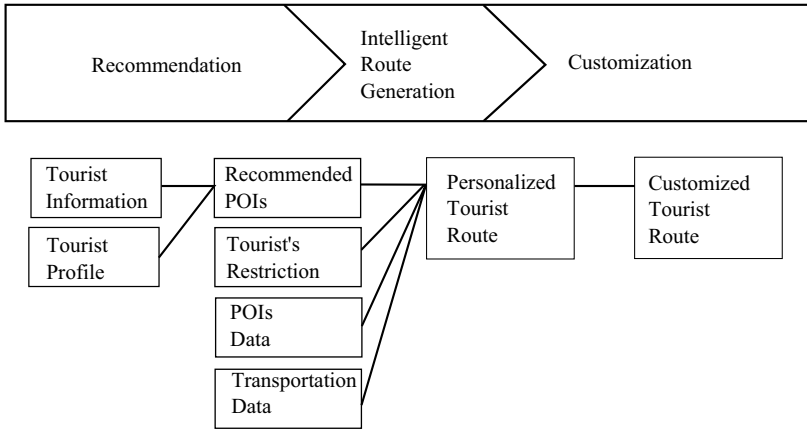


Fig. 1. PET flow

- **Route Generation.** Once the system knows which are the most appealing POIs for the tourist, an intelligent routing engine can combine this information with tourist’s restrictions (available time, number of days of the route, maximum budget, starting POI ...), POI data (location, opening hours, ticket price ...) and transportation data (traveling times, public transportation network data ...) to generate personalized tourist routes. This intelligent routing engine has to apply advanced algorithms from the field of Operations Research (OR) in order to generate routes in real-time.
- **Customization.** Finally, tourists appreciate to have the opportunity to customize the proposed personalized route to better fit their needs. Inserting new visits to the route, removing visits or reordering them are the basic functionalities of a customization engine.

The system we present in this paper focuses on the last two functionalities. First (Section 4) it applies an advanced OR algorithm to generate personalized tourist routes that take advantage of the existing public transportation network. Once a personalized route is generated, it gives to the tourists the opportunity to customize it (Section 5).

4 Route Generation

We have modeled the problem of generating personalized tourist routes with public transportation as a TDTOPTW. After analyzing different approaches, we finally designed a heuristic able to solve it in real-time [20,27], based on a fast heuristic for the TOPTW [17]. The inclusion of public transportation makes the process of building a solution much more complex: each distance calculation between two POIs becomes a Time Dependent Shortest Path problem (TDSP). A small increase in the planned leave time from one POI can cause a significant

increase in the arrival time to the next POI. For instance, when a tourist just misses the bus and has to wait for the next one or walk to the next POI.

Thus, in order to handle the difficulty of the TDTOPTW, we apply a hybrid approach combining two different heuristics. Each of them focuses on a different aspect of the problem.

First we solve in batch the TDSP problems between all the possible pairs of POIs with leave time steps of 1 minute. Then, we calculate an average travel time between each pair of POIs. Since these calculations can be made beforehand, the real-time requirement is eliminated for this part of the calculation and any suitable algorithm can be used to solve the TDSP problems. Delling [28] details the latest algorithms fulfilling this task. We have implemented a time dependent Dijkstra's Shortest Path algorithm modified to cope with public transportation [20]. Based on the conclusions of Pyrga et al. [21], we have applied a time dependent approach with simple transfer times. This way, we handle the extra difficulty added by the public transportation without the real-time requirement.

The complete calculation of the average travel time matrix for an instance with 50 POIs takes around 90 minutes on a PC Intel Core 2 Quad with 2.40 GHz processors and 2 GB Ram. The average travel times are stored in a database for a later fast retrieval.

Once we have calculated these average travel times, we solve the TDTOPTW in real-time as a regular TOPTW. Our design and implementation is based on the algorithm proposed by Vansteenwegen et al. [17] for the (T)OPTW. We give a general description of the algorithm here, for more details we refer to the (T)OPTW article.

The heuristic is based on Iterated Local Search (ILS) [29]. This method combines iteratively a local search with a perturbation phase. The local search heuristic inserts new visits to a route, one by one. For each visit i that can be inserted, the cheapest insertion time ($Shift_i$) is determined. For each of these visits the heuristic calculates a ratio, which relates the tourist score of the POI (the estimated satisfaction for a given tourist) to the time required to visit it. Among them, the heuristic selects the one with the highest ratio for insertion. This process is repeated until no more POIs can be inserted. A perturbation phase removes consecutive POIs from a route. After the removal, the heuristic shifts all visits following the removed visits towards the beginning of the route as much as possible, in order to avoid unnecessary waiting. The perturbation procedure and the local search heuristic are executed until a termination criterion is met. The heuristic returns the incumbent solution as the result.

Finally, once we find a final solution, we run a repair procedure introducing the real travel times between the POIs. Starting from the first POI of the route, we compare the average and the real travel time, taking into account the real leave time. If the real travel time is smaller than the average one, the travel time is adapted by advancing the arrive time to the visit towards the beginning of the route as much as possible. The waiting time and the leave time of the visit are also updated.

Otherwise, if the real travel time is larger than the average one, we arrive later to the visit. If this causes a visit to become infeasible, we remove it from the route and we update the route, moving the rest of the visits towards the beginning.

5 Route Customization

Once the system has generated a personalized route that maximizes the tourist satisfaction, the tourist can decide to change this route to better fit his/her own interests. Although this customization step is voluntary, it greatly increases the flexibility of the system.

The customization is based on six basic operations a PET should provide. The difficulty of these operations is directly related to the public transportation network. As the leave time from the previous POI changes, the travel time to the next POI does too. Thus, all the affected traveling times have to be recalculated. This process varies in function of the executed operation:

- **Add a visit.** When a tourist wants to add a new visit to a route, there are two possibilities to determine the insertion position. In the first one, the system finds the best insertion point, minimizing the total time (travel plus wait plus visit times). The second possibility consists on the tourist choosing the insertion order within a day. In any case, the system has to update the order of the following visits. The system has to insert the new visit and update the travel and wait times starting from the previous visit.
- **Remove a visit.** In this case the system has to remove a visit. Thus, all the following visits of the day are moved towards the beginning. Starting from the visit preceding the removed one, traveling and waiting times have to be recalculated until the end of the day is reached.
- **Move a visit towards the beginning of the route.** When a tourist wants to visit a POI earlier than scheduled, the system swaps this visit and the previous one. Starting from the one before the previous visit, traveling and waiting times have to be recalculated until the end of the day is reached. Obviously, this move can be repeated to schedule the visit even earlier.
- **Move a visit towards the end of the route.** When a tourist wants to visit a POI later than scheduled, the system swaps this visit and the next one. Starting from the previous visit, traveling and waiting times have to be recalculated until the end of the day is reached.
- **Move a visit to the previous/next day.** When a tourist wants to move a visit to the previous/next day, first the system has to remove the visit from the actual day. Once this operation is completed, the system finds the best insertion position (smallest required extra time) for the POI on the previous/next day.
- **Customization exception.** Once a tourist starts customizing a route, it is possible that a violation of some of the restrictions occurs. As the customization is a hand-made process realized by a tourist, in these cases the system shows an alert asking for confirmation before committing the change.

We have created the following exceptions to ask for explicit confirmation during the customization process:

- Maximum travel time/ budget exceed alert. This alert notifies that the maximum values introduces as restrictions have been exceeded.
- High wait/travel alert. This alert is launched if the proportion between visiting time and wait/travel time is high (higher than 40%).

We describe the prototype integrating the previous operations in Section 6.

6 Prototype

The prototype has been initialized with data about the city of San Sebastian. San Sebastian is a Spanish tourist city located near the French border. It has around 200.000 inhabitants and 50 POIs. Most tourists visit the city by combining public transportation with short walks. The public transportation network includes 467 bus stops, 26 lines and more than 65.000 direct bus connections between stops. The prototype has two different frontends. The first one is targeted for desktop interaction on the pre-trip stage. The second one is targeting mobile devices with a touch friendly user interface.

In relation to the application's interaction with the user, first of all, tourists have to enter their restrictions. Moreover, the system has to know their profile in order to assign the correct tourist score to each POI. Although there are advanced approaches available on the literature [26], this step is out of the scope of this paper. Thus, we have followed a simple approach, having four different profiles (nature, culture, gastronomy, leisure). For each POI and profile, the system administrator has to assign the estimated tourist score (0-100) and the estimated visit duration. Tourists have the opportunity to overwrite these scores. They can update both the tourist score and the visit duration. Besides, they can mark some POIs as must-seen/must-avoid POI. The system automatically assigns these POIs a tourist score of 100/0.

With this input data, the system generates a personalized tourist route in real time (less than 5 seconds). This route takes advantage of the public transportation network of the city, shortening the traveling times and increasing the mobility of tourists. Regarding the traveling information, the travel between the previous POI and the actual POI is divided in transportation links. Each transportation link stores details about how the traveling is done. If a tourist goes on foot the distance between POIs and the estimated travel times are stored. Otherwise, for each public transportation involved, the traveling distance from/to the previous/actual POI/stop are stored, containing details about the public transportation line, the estimated waiting and traveling times.

The following example details how to arrive from a hotel outside the city center (NH) to the cathedral of San Sebastian (Buen Pastor):

- Walk 2 minutes from hotel NH to bus stop Avda Tolosa 5
- Wait 5 minutes for bus number 5
- Take the bus and travel 12 minutes until bus stop Buen Pastor Cathedral
- Walk 1 minute to Buen Pastor Cathedral

Apart from the functionalities presented on the previous sections, once the customization process is finished, a tourist can access the details of the whole route. Besides depicting all POIs on a map, the system provides full details for each visit. As an additional functionality, tourists can generate and download a pdf of the whole route. Finally, tourists can navigate the destination using a map based interface. We show all POIs on a map and tourists can choose to access details about any POI.

6.1 Architecture

We have designed and implemented a client server architecture. The client consists on a thick client executed on a Web browser. The server is constituted by a database (MySQL), and an application Server (Apache Tomcat). The client is a Web client based on Google Web Toolkit (GWT). GWT is a framework from Google allowing to develop a Web application on Java. GWT's compiler transforms the Java code on the corresponding HTML and JavaScript code compatible with main browsers (including iPhone's and Android's browsers).

We have followed the best practices promoted by Google [30], which can be summarized as the adoption of the following elements: dependency injection, Model-View-Presenter (MVP) pattern, bus event and browser history. On the next subsections we briefly introduce both backends of the prototype.

6.2 Desktop Client

The desktop client offers two main functionalities: generation of personalized tourist routes and their customization. Besides, it allows to navigate through the destination, to load previously customized routes, to update user's profile and to update the tourist score and visit duration.

Once tourists have chosen their profile and updated tourist score for POIs and visit duration (optional), they have to enter their restrictions. Then, the system generates a personalized tourist route. Visits are presented ordered in a map (Figure 2(a)). The list of visits is also shown on the sidebar. We assign a letter to each visit according to its order (A,B, C, ...).

Selecting a visit, more information about it and its POI (arrival time, visit duration, leave time) can be seen. Moreover, tourists can start customizing the route using five different buttons (Figure 2(b)). Four of the buttons represent a direction: Up, to move a visit towards the beginning of a route; Down, to move a visit towards the end of a route; Left, to move a visit to the previous day; and Right, to move a visit to the next day. And finally, the last one, the one with a minus character label, allows to remove the visit. Finally, a button with a plus character allows to add new visit. Tourists have to choose whether they want to add the new visit at a certain order, at the end of the day, or at the optimal order (the one minimizing total route time). Once the customization process is over, the tourists can access a detailed summary of the route and generate a downloadable pdf.

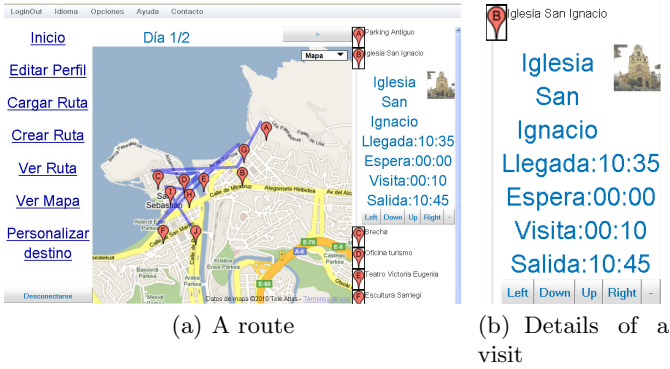


Fig. 2. Details of the desktop prototype

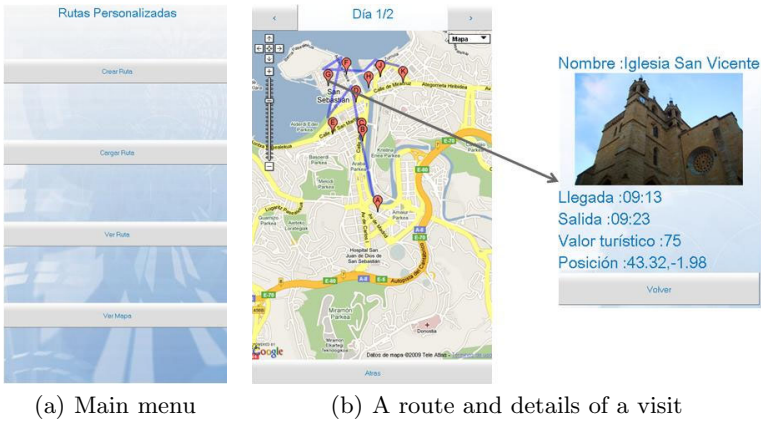


Fig. 3. Details of the mobile prototype

6.3 Mobile Client

The mobile client offers the same functionality, except for the customization. It has been taken out in order to improve the user friendliness of the prototype. The interface has been adapted to use with touch screens. Figure 3(a) shows the main screen with its button enlarged. Figure 3(b) shows a map with a personalized route and the details about one visit. Due to the reduced screen size, on the mobile client the details are shown on a new window.

7 Evaluation

We have evaluated the performance of our approach against test instances for the city of San Sebastian. Results are able to obtain routes in real time (worst case of 0.2 seconds for a 2 day 8 hour per day route) in a scenario with 50 POIs, 26 public transportation lines and 467 stops, and are comparable to the best

results available on the literature [20,27]. Moreover, most available examples are not able to generate routes in near real-time and the times they require to obtain a solution for comparable test instances are an order of magnitude higher.

We plan to test the tourist accuracy of the routes proposed by the prototype with real tourist for the city of San Sebastian during the summer of 2010.

8 Conclusions and Future Work

We have presented an intelligent routing system able to generate and customize personalized tourist routes in real-time and taking into account public transportation. Although the whole process is divided in three steps, we have focused only on the last two: route generation and route customization.

We have modeled the tourist planning problem, integrating public transportation, as the Time Dependent Team Orienteering Problem with Time Windows (TDTOPTW). We have designed a heuristic able to solve it in real-time, precalculating the average travel times between each pair of POIs in a preprocessing step. Moreover, the system includes the basic functionalities required to customize the generated route. They allow to move, add and remove visits in a route.

Finally, we have shown a prototype which generates and customizes routes in real-time. This prototype has a desktop and a mobile frontend and has been successfully evaluated against test instances. Furthermore we plan to fulfil tests with real tourists during the summer of 2010 in the city of San Sebastian in order to evaluate the quality of the generated personalized tourist routes from the tourists' point of view.

Future works consists on extending the system to more cities with a different public transport network topology. The next one consists on integrating an advanced recommendation system in a wholly functional PET. Finally, we would like to insert social network capabilities, allowing to store, share and add travel experiences to better help tourists on the destination.

Acknowledgments

The authors would like to thank the Basque Government for partially funding this work through the neurebide and etourgune projects and the Centre for Industrial Management of the Katholieke Universiteit Leuven for hosting Ander Garcia as a guest researcher during 2009. Pieter Vansteenwegen is a post-doctoral research fellow of the "Fonds Wetenschappelijk Onderzoek - Vlaanderen (FWO)". Finally, authors would like to thank Javier Vallejo from "Compañía del Tranvia de San Sebastian" for providing real data about the public transportation network of the city.

References

1. Beer, T., Fuchs, M., Höpken, W., Werthner, H., Rasinger, J.: Caips: A context-aware information push service in tourism. In: *Information and Communication Technologies in Tourism 2007*, Ljubljana, Slovenia, pp. 129–149. Springer, New York (2007)

2. Schmidt-Belz, B., Laamanen, H., Poslad, S., Zipf, A.: Location-based mobile tourist services—first user experiences. In: *Information and Communication Technologies in Tourism 2003*, Ljubljana, Slovenia, pp. 115–123. Springer, Wien (2003)
3. Stroobants, R.: *Mobile tourist guides*. Technical report, Katholieke Universiteit Leuven, Belgium (2006)
4. Vansteenwegen, P.: *Planning in tourism and public transportation*. PhD thesis, Centre for Industrial Management, Katholieke Universiteit Leuven, Belgium (2008)
5. Souffriau, W.: *Automated Tourist Decision Support*. PhD thesis, Centre for Industrial Management, Katholieke Universiteit Leuven, Belgium (2010)
6. Kramer, R., Modsching, M., ten Hagen, K., Gretzel, U.: Behavioural impacts of mobile tour guides. In: *Information and Communication Technologies in Tourism 2007*, Ljubljana, Slovenia, pp. 109–118. Springer, Vienna (2007)
7. Maruyama, A., Shibata, N., Murata, Y., Yasumoto, K., Ito, M.: A personal tourism navigation system to support traveling multiple destinations with time restrictions. In: *International Conference on Advanced Information Networking and Applications*, vol. 2, pp. 18–21 (2004)
8. Castillo, L., Armengol, E., Onaindía, E., Sebastiá, L., González-Boticario, J., Rodríguez, A., Fernández, S., Arias, J., Borrajo, D.: Samap. an user-oriented adaptive system for planning tourist visits. *Expert Systems With Applications* 35(2), 1318–1332 (2008)
9. Zenker, B., Ludwig, B., Schrader, J.: Rose - assisting pedestrians to find preferred events and comfortable public transport connections. In: *ACM Mobility Conference 2009*. ACM, New York (2009) (accepted)
10. Souffriau, W., Vansteenwegen, P., Vertommen, J., Vanden Berghe, G., Van Oudheusden, D.: A personalized tourist trip design algorithm for mobile tourist guides. *Applied Artificial Intelligence* 22(10), 964–985 (2008)
11. Garcia, A., Linaza, M., Arbelaitz, O., Vansteenwegen, P.: Intelligent routing system for a personalised electronic tourist guide. In: *Information and Communication Technologies in Tourism 2009*, Amsterdam, The Netherlands, pp. 185–197. Springer, Heidelberg (2009)
12. Tsiligirides, T.: Heuristic methods applied to orienteering. *Journal of the Operational Research Society* 35(9), 797–809 (1984)
13. Vansteenwegen, P., Souffriau, W., Van Oudheusden, D.: The orienteering problem: a survey. *European Journal of Operational Research* (2009) (accepted)
14. Savelsbergh, M.W.P.: Local search in routing problems with time windows. *Annals of Operations Research* 4(1), 285–305 (1985)
15. Righini, G., Salani, M.: Decremental state space relaxation strategies and initialization heuristics for solving the orienteering problem with time windows with dynamic programming. *Computers & Operations Research* 36(4), 1191–1203 (2009)
16. Tricoire, F., Romauch, M., Doerner, K., Hartl, R.: Heuristics for the multi-period orienteering problem with multiple time windows. *Computers & Operations Research* 37(2), 351–367 (2010)
17. Vansteenwegen, P., Souffriau, W., Vanden Berghe, G., Van Oudheusden, D.: Iterated local search for the team orienteering problem with time windows. *Computers & Operations Research* 36(12), 3281–3290 (2009)
18. Montemanni, R., Gambardella, L.: Ant colony system for team orienteering problems with time windows. *Foundations of Computing and Decision Sciences* 34 (2009)
19. Fomin, F., Lingas, A.: Approximation algorithms for time-dependent orienteering. *Information Processing Letters* 83(2), 57–62 (2002)

20. Garcia, A., Arbelaitz, O., Vansteenwegen, P., Souffriau, W., Linaza, M.: Hybrid approach for the public transportation time dependent orienteering problem with time windows. In: Corchado, E., Graña Romay, M., Manhaes Savio, A. (eds.) HAIS 2010. LNCS, vol. 6077, pp. 151–158. Springer, Heidelberg (2010)
21. Pyrga, E., Schulz, F., Wagner, D., Zaroliagis, C.: Efficient models for timetable information in public transportation systems. *Journal of Experimental Algorithmics* 12, 1–39 (2008)
22. Ding, B., Yu, J., Qin, L.: Finding time-dependent shortest paths over large graphs. In: EDBT 2008: Proceedings of the 11th international conference on Extending database technology, pp. 205–216. ACM, New York (2008)
23. Zografos, K.G., Androustopoulos, K.N.: Algorithms for itinerary planning in multimodal transportation networks. *IEEE Transactions on Intelligent Transportation Systems* 9(1), 175–184 (2008)
24. Chiu, D.K.W., Lee, O.K.F., Leung, H.F., Au, E.W.K., Wong, M.C.W.: A multimodal agent based mobile route advisory system for public transport network. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences, HICSS 2005, p. 92.2 (2005)
25. Tumas, G., Ricci, F.: Personalized mobile city transport advisory system. In: Information and Communication Technologies in Tourism 2009, Amsterdam, The Netherlands, pp. 173–183. Springer, Wien (2009)
26. Kabassi, K.: Personalizing recommendations for tourists. *Telematics and Informatics* 27(1), 51–66 (2010)
27. Garcia, A., Vansteenwegen, P., Arbelaitz, O., Souffriau, W., Linaza, M.: Integrating public transportation in personalised electronic tourist guides. *Computers & Operations Research*, Special issue: Transport Scheduling (2010) (under review)
28. Delling, P.: Engineering and Augmenting Route Planning Algorithms. PhD thesis, Fakultät für Informatik Universität Fridericiana zu Karlsruhe (2009)
29. Lourenço, H.R., Martin, O., Stutzle, T.: Iterated local search. In: Kochenberger, F.G., Glover, G. (eds.) *Handbook of Metaheuristics*, pp. 321–353. Kluwer Academic Publishers, Dordrecht (2003)
30. Ryan, R.: Google web toolkit architecture: Best practices for architecting your gwt app. In: Google IO Developer Conference 2009 (2009)