Tour Suggestion for Outdoor Activities

Joris Maervoet[1,2,⋆], Pascal Brackman[3], Katja Verbeeck[1],
Patrick De Causmaecker[2,4], and Greet Vanden Berghe[1]

[1] KAHO Sint-Lieven, Computer Science, CODeS,
Gebr. De Smetstraat 1, 9000 Ghent, Belgium
[2] KU Leuven-Kulak, Department of Computer Science, CODeS,
Etienne Sabbelaan 53, 8500 Kortrijk, Belgium
[3] RouteYou.com, Kerkstraat 108, 9050 Ghent, Belgium
[4] iMinds - ITEC - KU Leuven

Abstract. The present article introduces the outdoor activity tour suggestion problem (OATSP). This problem involves finding a closed path of maximal attractiveness in a transportation network graph, given a target path length and tolerance. Total path attractiveness is evaluated as the sum of the average arc attractiveness and the sum of the vertex prizes in the path. This problem definition takes its rise in the design of an interactive web application, which suggests closed paths for several outdoor activity routing modi, such as mountain biking. Both path length and starting point are specified by the user. The inclusion of POIs of some given types enrich the suggested outdoor activity experience.

A fast method for the generation of heuristic solutions to the OATSP is presented. It is based on spatial filtering, the evaluation of triangles in a simplified search space and shortest path calculation. It generates valuable suggestions in the context of a web application. It is a promising method to generate candidate paths used by any local search algorithm, which further optimizes the solution.

Keywords: tour suggestion, transportation network graphs.

1 Introduction

The company RouteYou offers recreational navigation for several outdoor activity modes such as hiking and mountain biking. This involves maintenance of a set of transportation network graphs, in which each arc r has a length l_r and attractiveness $0 < a_r \leq 1$. The latter parameter models suitability to the applicable outdoor activity mode, in terms of the arc's scenic context, physical condition and relation to traffic. The motive behind the present paper is the design of a tour suggestion module, which plans attractive round trips for any of the company's outdoor activity modes. This module requires that the user chooses an outdoor activity mode and specifies a target path length and a starting point. Within some seconds, it returns a closed path, consisting of arcs of optimal attractiveness and satisfies the length and starting point constraints. The desirable

[⋆] Corresponding author.

S. Liang, X. Wang, and C. Claramunt (Eds.): W2GIS 2013, LNCS 7820, pp. 54–63, 2013.
© Springer-Verlag Berlin Heidelberg 2013

path has some constraints because users tend not to accept paths with a considerable number of self-intersections or with recurring subpaths. Paths in clockwise direction are usually prefered in countries with right-hand traffic because they ease turn traffic manoeuvres. Moreover, the user is able to select one or several POI types of preference, such that the output path results from a trade-off of arc attractiveness and the number of contained POIs of the preferred types. Examples of interesting mode - POI type combinations are hiking with mountains and motorcycling with scenic viewpoints. Another scenario is that external organizations set the mode and POI types of preference in a customized planner. This sort of planner aims at promoting a peculiar type of tourism in a specific region e.g. cycling along the Châteaux of the Loire Valley in France. An extension to this module is the generation of multiple suggestions. This means that the user can browse through a set of m path suggestions, for the same set of preferences and constraints. It involves finding the set of m most attractive tours that are spatially different.

The next section gives a literature overview of models applied in the domain of leisure and tourism. Section 3 introduces the OATSP, which formalizes the tour suggestion problem described above. The approach presented in Section 4 enables generating a set of heuristic solutions to the OATSP in a low computational time. The following section discusses two sets of individual tours obtained by this approach. Section 6 is the conclusion of this work.

2 Tour Suggestion Models for Leisure and Tourism

The tour suggestion problem for leisure and tourism (TSPLT) involves generating a path through a transportation network visiting some arcs and/or points of interest (POIs). The path should optimally match the end user's preferences or some general recreational preference, given a set of constraints (adapted from the itinerary planning problem formulated by Shcherbina and Shembeleva [15]). It has applications in several subdomains of leisure and tourism. *Recreational point-to-point navigation* aims at generating a path from A to B, which is tailored to a specific recreational navigation mode, such as nordic walking. *Individual city trip planning* involves providing a tour schedule along a selection of POIs, satisfying the personal preferences of an individual intending to visit a city for a certain amount of time. This kind of services are often realized as a web-based application or a mobile client-server application (e.g. [1,16]), generating on-the-fly suggestions for the end user. This is usually not required for applications in collective tourism planning (bus tour planning, cruise itinerary planning). The path generated in the TSPLT may be open or closed. In individual city tour planning for instance, the path is - in most cases - closed. The trip typically both starts at and ends in the tourist's hotel, a parking lot or a train station.

A first type of systems model the problem as a shortest path (SP) problem. This model generates exclusively open paths and focuses on the suitability of the arcs for a certain purpose. In the most common recreational SP approach, the inverse suitability is encoded in single arc weights of a directed weighted graph, corresponding to the transportation network. Path generation involves that common SP algorithms are used to find the path with the lowest cost between two nodes of this graph. Traditional SP algorithms are Dijkstra [3] and its variants [24] and A* [6]. The use of single scenic/attractive weights has often been suggested in this context (e.g. [13,9]). The company presented in the introduction adapted this concept by introducing weight attractiveness for a series of recreational routing modi. Attractiveness also deals with physical conditions and traffic aspects of the roads. Rogers and Langley [12] model the attractiveness of weights by a linear combination of criteria that reflect the end user's preferences. Niaraki and Kim [11] developed an ontology-based technique that generates network weights for personalized routing planning. Tarapata [20] states that single-objective functions are not sufficiently adequate to model real SP problems. He presents a classification of multi-objective shortest path (MOSP) problems, which are used in other real application domains, such as routing with quality-of-service in computer networks. The author identifies six general solution methods to MOSP problems, including mathematical optimization and objective function hierarchization. Hochmair and Navrath [7] argue that the computation of attractive routes is generally beyond the ability of SP algorithms, since an SP algorithm is not able to find a route that maximizes a benefit criterion. However, they demonstrate the practical value of single criterion SP computation for finding this type of routes.

A second type of systems use a problem model that trades off POI selection with time or distance. It originates from the field of Operations Research (OR). It involves selecting a sequence of POIs from an eventually larger input set of POIs. This sequence should meet certain preferences and/or must satisfy a set of constraints. The POI sequence selection is often preceded by a POI filtering mechanism, improving the sequence selection performance. The model takes into account the travel time or distance between candidate POIs, aided by a precalculated travel time/distance matrix or a heuristic estimation function. The resulting path is obtained by concatenating precalculated (shortest) paths between the selected POIs, or by recalculating the complete path using via-points. The latter approach is useful in order to avoid undesirable U-turns and forbidden traffic manoeuvres passing through a selected POI. Godart [5] presented a version of the traveling salesperson problem (TSP) that integrates activity selection and lodging availability for trip planning problems. Deitch and Ladany [2] introduced the bus touring problem (BTP). It requires an undirected graph in which the vertices represent visiting sites and the edges represent connecting scenic routes. Both edges and vertices have associated attractivity values and require traveling/visiting times. The goal is to find a (closed path) bus tour of maximal total attractivity, below a given maximal tour time. The attractivity of recurrently visited vertices and edges is only counted once. The authors show that the BTP can be transformed to the Orienteering Problem (OP). Both Suna and Lee [19] and Maruyama et al. [10] have built tourist trip recommender systems based on variants of the prize collecting TSP (PCTSP). The original model minimizes the total travel cost minus the sum of the benefit criterion values of the POIs along the selected path. Suna and Lee integrate a personal interest factor in the travel cost weights. The two following query models in the field of spatial databases focus on trip planning with *typed* locations. Li et al. [8] introduce the trip planning query, which involves a request for the shortest route from and to a given point that passes through at least one point of any of the specified set of location types. The optimal sequenced route query [14] looks for the shortest path that visits locations according to a specified sequence of POI types. An example of a POI type sequence for leisure is: (1) hair dresser (2) restaurant (3) cinema. Vansteenwegen and Van Oudheusden [23] introduced the tourist trip design problem (TTDP), which is modelled as an OP with time windows. This model starts from a fully interconnected distance/time-weighted graph in which the vertices represent POIs with a personalized [18] score. It involves finding the sequence of POIs that maximizes the total score of the selected POIs, while the total path weight must not exceed a given value. Each POI can only be visited once. Moreover, certain POIs are only available within certain time windows (cf. opening hours). Very good approximate solutions are found with iterated local search by Vansteenwegen et al. [21,22]. They have shown the practicability of the TTDP in city trip planning. All problems in this second class are $\mathcal{NP}$-hard. Both Shcherbina and Shembeleva [15] and Souffriau and Vansteenwegen [17] provide a more detailed overview of the models and functionalities of this type of systems.

3 The Outdoor Activity Tour Suggestion Problem

The BTP model suits the requirements of company's tour suggestion module best. The objective function takes into account both edge and vertex attractiveness, whereas tour time resides under the constraints. However, the tour suggestion module requires a target path length instead of a maximal travel time. This gives rise to a distance window constraint. In order to model the particular tour suggestion problem, we introduce the OATSP.

This model requires a directed graph $G = (V, A)$, for which

- any arc $r \in A$ is an ordered pair $(v_1, v_2) \in V^2$,
- any arc $r \in A$ has an associated attractiveness $0 < a_r \leq 1$ and length l_r,
- reverse arcs have equal attractiveness and length,
- each vertex v has an associated prize $0 \leq p_v \leq 1$, and,
- a closed path C is a series of arcs, which are circularly subsequent in G. Each element of A appears at most once in C. C_i denotes the arc at the i-th position in C. The sets $A_C \subseteq A$ and $V_C \subseteq V$ consist of the arcs and vertices visited by the closed path C. The set A'_C contains any arc $C_i = (v_1, v_2)$ of C that is not preceded by its reverse arc $C_j = (v_2, v_1)$ with $j < i$.

Given a prefered path length l_p, length tolerance t and starting vertex v_s, a solution to the OATSP is a closed path C in G satisfying

1. C_1 starts in v_s
2. $(1-t) \cdot l_p \leq \sum_{r \in A_C} l_r \leq (1+t) \cdot l_p$
3. $\varphi \cdot \dfrac{\sum_{r \in A'_C} (a_r \cdot l_r)}{\sum_{r \in A_C} l_r} + \sum_{v \in V_C} p_v$ is maximal

The parameter φ determines the relative importance of arc to node attractiveness. This problem is a variant of the PCTSP [4]. This model suits the requirements of the tour suggestion module. It is able to suggest closed paths of a target length and optimal arc and POI attractiveness, by setting the vertex prizes to the degree of membership to the specified categories. Since recurrent visits to reverse arcs and nodes are penalized by the objective function, solutions tend to avoid U-turns and recurring reverse subpaths. The model does not allow recurring arcs. It does allow multiple vertex and reverse arc visits since the starting vertex or any of the POIs may be located on a subgraph with low connectivity. Planar self-intersections can be avoided by assigning a low threshold value to the prizes of vertices that do not belong to any of the specified categories.

4 Approach

A multiple tour suggestion module has been designed for the web application of the company. It enables generating a set of m heuristic solutions to the OATSP within a low computational time, in an environment without precalculated paths nor distances available. The main algorithm starts by determining a *feasibility window (FW)*. This square area discerns the arcs and nodes reachable in a round trip through v_s given the path length window constraint. The algorithm assumes that, within the FW, (a) only a few (ranging from none to the order of tens) vertices v have $p_v > 0$, (b) there is a diversity of attractiveness a_r amongst the arcs r, and, (c) vertices with $p_v > 0$ have a high probability to be located along trajectories of higher attractiveness than vertices in their local neighbourhoods. These conditions often hold for the tour suggestion module presented, where the user selects few POI categories and relatively low values for l_p. Moreover, it assumes low values of φ.

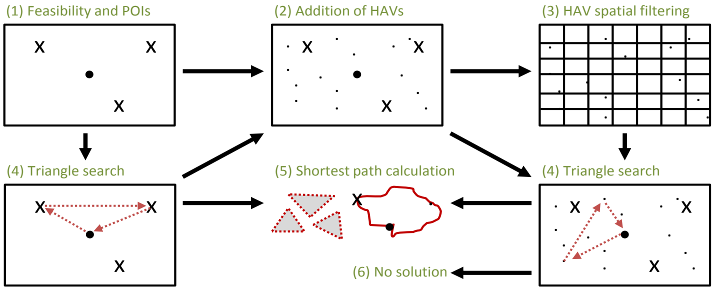


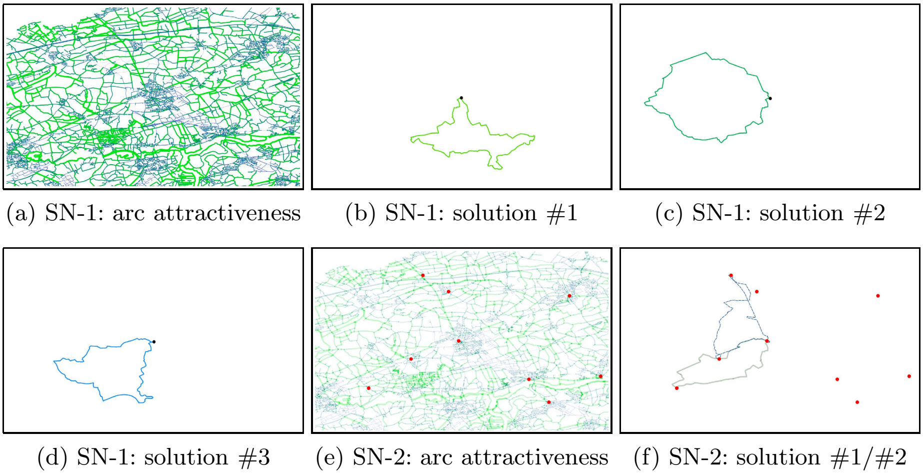
Fig. 1. State diagram of the fast heuristic algorithm for the OATSP

Fig. 1 shows the state diagram of the main algorithm. In state (1), the FW is generated in the geographic coordinate plane, such that v_s is in the center and the projected width and height measured through v_s, equals l_p/e_1. If the FW contains more than n_1 feasible POIs ($p_v > 0$), the algorithm continues with triangle search. Otherwise, the window is enriched by highly attractive vertices (HAVs), in state (2). HAVs are auxiliary POIs representing the most attractive arcs in the FW. This phase looks for the maximal value a_{min} for which there exist at least n_3 arcs r with $a_r \geq a_{min}$. Any vertex that is connected by an arc r

with $a_r \geq a_{min}$, is an HAV. The maximal attractiveness of the connecting arcs of an HAV v is denoted a_v. Next, all HAVs are promoted as POIs, with p_v set to $\varphi a_v/\#HAVs$. If the total number of POIs does not exceed n_2, the algorithm continues with triangle search. Otherwise, the HAVs are subjected to spatial filtering (state (3)). This involves that all points are categorized into one of the boxes of the $g_1 \times g_1$ grid constructed over the window. Only one auxiliary POI per box is kept, absorbing the sum of p_v-values of the other points in the box. Triangle search in state (4) involves the brute-force evaluation of any directed triangle made up by v_s and any other two POIs within the window. This evaluation consists of two steps. First, a triangle undergoes a fast feasibility check. A triangle is feasible if its direction is clockwise (in case of right-hand traffic) and if its perimeter is between $(1 \pm t) \cdot l_p/e_2$. Next, an evaluation function computes a score for a feasible triangle in this simplified window representation. This function returns a weighted sum of the prizes p_v for any of the POIs involved. Prizes of POIs located at one of the triangle vertices are given double weights. POIs located on one of the triangle edges generate p_v, which decreases as the elliptical distance to the closest triangle edge increases. If all angles are greater than x_1, a global bonus $(*x_2)$ is granted. When no feasible triangle after HAV promotion, the algorithm does not return any solution (state (6)). Otherwise, the triangle of highest score is passed to the SP calculation state. Given a triangle abc, this fifth state entails finding the concatenation of paths of lowest cost between (a, b), (b, c) and (c, a) in a graph with arc weights equal to l_r/a_r. In order to avoid U-turns and recurring subpaths, the weights of both the forward and available reverse arcs of the resulting subpath are drastically increased, after each subrouting. If the concatenated path does not meet the length window constraint, the paths between (a, b), (b, c) and (c, a) are calculated in a changed order and again concatenated. If it still does not meet the constraint, it is rejected after which the second best triangle is processed, and so on.

The multiple suggestion extension is realized by the integration of a simple *competitive learning* algorithm in the triangle search. It relies on a triangle distance function which is defined as follows. The two variable vertices of a triangle are categorized into one of the boxes of the $g_2 \times g_2$ grid constructed over the window. The *distance* between two (clockwise) triangles is the sum of the Manhattan distance (MD) between the first vertices and the MD between the second vertices in the grid. Two triangles are called *resemblant* if one of the MDs is lower than 2. During the brute-force triangle evaluation, the algorithm manages a store of maximally m prototype solutions. Suppose the current solution does not resemble any prototype solution. If there are less than m prototype solutions, the current solution enters the store as a new prototype. If the store is full and the current solution is better than the worst prototype, it replaces the prototype. Suppose the current solution resembles one or more prototype solutions. If it is better than the closest prototype, it replaces the prototype. In the end, the m prototype solutions are passed to the SP calculation state.

Parameter Overview. The parameters e_1 and e_2 should be set to the minimal and average ratio of a SP length to the Euclidean distance in a graph with arc weights l_r/a_r. The lower limit n_1 and upper limit n_2 determine the number


(a) SN-1: arc attractiveness (b) SN-1: solution #1 (c) SN-1: solution #2

(d) SN-1: solution #3 (e) SN-2: arc attractiveness (f) SN-2: solution #1/#2

Fig. 2. Arc attractiveness map and the three/two best tours in the solution store for SN-1/SN-2. Each subfigure is shown in the FW, centered around v_s. Highly attractive arcs are depicted by thick green lines, and less attractive arcs by thin blue lines. The red dots indicate the POIs of the type 'interesting church'. The best solution of SN-2 is indicated by a solid line.

of POIs used for triangle search. Too low numbers can result in a shortage of feasible triangles, whereas high numbers increase the computational time. n_3 determines the number of HAVs used for triangle search, but depends on the dispersion of arcs of equal attractiveness in the FW. x_1 and x_2 should be experimentally determined in order to reward less acute-angled triangles, since very acute angles result in poor tours.

5 Results

The approach introduced in the previous section has been tested for the outdoor activity mode 'attractive cycling'. The transportation network graph for this mode contains both paved and unpaved roads. The OATSP settings are $\varphi = 1$ and $t = 0.30$. The remaining parameters are set as follows: $e_1 = 1.41$, $e_2 = 1.6$, $n_1 = 5$, $n_2 = 60$, $n_3 = 20$, $x_1 = 40°$, $x_2 = 1.5$, $g_1 = g_2 = 10$, and $m = 10$.

A first experiment, called SN-1, assigns the centre of a medium-sized city in Belgium as starting vertex v_s and sets $l_p = 30km$. No POI category of interest is specified, so initially any vertex prize $p_v = 0$. Fig. 2 shows the arc attractiveness map within the FW and the 3 out of 10 tours of highest scores in the solution store. Most of the arcs within the city are substantially less attractive than the arcs in the neighbourhood. The arcs in the south of the window, either along the rivers or within a woody region, have the highest attractiveness. Each of the top-3 tours visits this region. Any of the 10 tours in the store were assessed by amateur cyclists as attractive tours. They valued the solution diversity in the store highly. Experiment SN-2 adds the POI category 'interesting church' to the

Table 1. Top-5 result characteristics for both experiments

SN-1	Δeval.f.	tour length	OATSP obj.f.
# 1	0.4477	37.6 km	0.8126
# 2	0.3937	34.9 km	0.8316
# 3	0.2998	31.8 km	0.7800
# 4	0.2619	28.6 km	0.7297
# 5	0.2280	30.4 km	0.6826

SN-2	Δeval.f.	tour length	OATSP obj.f.
# 1	4.9709	25.1 km	3.7618
# 2	4.9073	30.1 km	3.7265
# 3	4.3079	34.1 km	2.7078
# 4	4.0000	32.3 km	2.7393
# 5	4.0000	25.6 km	2.7660

specifications of SN-1. Each closest vertex to a POI of this category gets $p_v = 1$. Fig. 2 shows these POIs, and the 2 out of 5 tours of highest scores in the solution store. One of the POIs is located very close to and to the west of v_s. This gives tours in the west containing this POI, priority over the other tours.

Experiment Statistics. In SN-1, 764 HAVs were added to the FW and reduced to a set of 84. It took 752 ms to evaluate[1] $84 \cdot 83$ triangles (out of which 1256 were feasible). SN-2 only went through states (1), (4) and (5), so no HAVs were added to a set of 9 POIs. It took only 2 ms to evaluate $9 \cdot 8$ triangles. Only 5 out of 13 feasible triangles were considered sufficiently diverse by the competitive learning algorithm and kept in the solution store. The triangle evaluation scores, tour lengths and OATSP objective function scores of the 5 best SN-1 and SN-2 results in the solution store are given in Table 1. Specifically in SN-1, the OATSP objective function equals the attractiveness average of arcs visited by the tour. The triangle evaluation score correlates with the objective function. Only the solution of rank 1 has been overestimated. The average arc attractiveness along the (a, b) subpath of this solution is remarkably lower than the average along the other subpaths. In the case of SN-2, the OATSP objective function takes into account both arc and node attractiveness. Only POIs were assigned to the triangle vertices, and therefore each tour scores at least two. Since no HAVs were used to calculate the triangle evaluation score, this function is only effective in predicting the number of POIs contained by a tour.

6 Conclusion

The OATSP, introduced in the present article, involves finding attractive closed paths in a transportation network graph, tailored for a specific outdoor activity mode. An experiment showed that the presented algorithm is able to generate a set of heuristic solutions to the OATSP, satisfying the constraints. The core of this algorithm is a brute-force triangle evaluation in the FW, containing the POIs with $p_v > 0$ and a set of auxiliary POIs, constrained to the number of g_1^2. Although a quadratic number of triangles is evaluated, the evaluation runs in low computational time. The triangle that receives the best evaluation is passed to an SP calculation module, prioritizing paths along attractive arcs. The triangle evaluation function has been found an efficient heuristic for the OATSP objective value of the resulting path. A simple competitive learning algorithm

[1] Using PHP (CLI) 5.3.6 in Ubuntu 11.10, with an Intel Core i7-920 Processor.

enables generating m tours of high objective value, which are spatially different. This algorithm of low computational impact showed to be effective.

Further work includes improving the introduced algorithm and a comparison with the route quality and computational time of the results generated by a local search method. Potential improvements are the introduction of precalculated paths in the triangle evaluation, the replacement of the brute-force technique by a heuristic selection of triangles, and, the evaluation of polygons instead of triangles.

Acknowledgment. The research has been carried out as part of the industrial PhD project "Structural heuristics for personalized routes" funded by the IWT (090726) and the company RouteYou.

References

1. Cheverst, K., Davies, N., Mitchell, K., Friday, A., Efstratiou, C.: Developing a context-aware electronic tourist guide: some issues and experiences. In: Proceedings of the Conference on Human Factors in Computing Systems, pp. 17–24. ACM (2000)
2. Deitch, R., Ladany, S.: The one-period bus touring problem: Solved by an effective heuristic for the orienteering tour problem and improvement algorithm. European Journal of Operational Research 127(1), 69–77 (2000)
3. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik 1, 269–271 (1959)
4. Fischetti, M., Salazar-González, J.J., Toth, P.: The generalized traveling salesman and orienteering problems. In: The Traveling Salesman Problem and Its Variations, Combinatorial Optimization, vol. 12, pp. 609–662. Springer US (2004)
5. Godart, J.: Combinatorial optimisation based decision support system for trip planning. In: Buhalis, D., Schertler, W. (eds.) Information and Communication Technologies in Tourism, pp. 318–327. Springer (1999)
6. Hart, P.E., Nilsson, N.J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE Trans. Syst. Sci. Cybernetics 4(2), 100–107 (1968)
7. Hochmair, H.H., Navratil, G.: Computation of scenic routes in street networks. In: Car, A., Griesebner, G., Strobl, J. (eds.) Proceedings of the Geoinformatics Forum, Salzburg, Austria, pp. 124–133. Wichmann Verlag, Heidelberg (2008)
8. Li, F., Cheng, D., Hadjieleftheriou, M., Kollios, G., Teng, S.-H.: On Trip Planning Queries in Spatial Databases. In: Medeiros, C.B., Egenhofer, M., Bertino, E. (eds.) SSTD 2005. LNCS, vol. 3633, pp. 273–290. Springer, Heidelberg (2005)
9. Malaka, R., Zipf, A.: Deep map - challenging IT research in the framework of a tourist information system. In: Information and Communication Technologies in Tourism, pp. 15–27 (2000)
10. Maruyama, A., Shibata, N., Murata, Y., Yasumoto, K., Ito, M.: A personal tourism navigation system to support traveling multiple destinations with time restrictions. In: Proceedings of AINA 2004, pp. 18–22. IEEE Computer Society (2004)
11. Niaraki, A.S., Kim, K.: Ontology based personalized route planning system using a multi-criteria decision making approach. ESWA 36(2), 2250–2259 (2009)
12. Rogers, S., Langley, P.: Personalized driving route recommendations. In: Proceedings of the AAAI Workshop on Recommender Systems, pp. 96–100. Madison (1998)

13. Sanders, P., Schultes, D.: Engineering Fast Route Planning Algorithms. In: Demetrescu, C. (ed.) WEA 2007. LNCS, vol. 4525, pp. 23–36. Springer, Heidelberg (2007)
14. Sharifzadeh, M., Shahabi, C.: Processing optimal sequenced route queries using voronoi diagrams. Geoinformatica 12(4), 411–433 (2008)
15. Shcherbina, O., Shembeleva, E.: Modeling recreational systems using optimization techniques and information technologies. Annals of OR, 1–21 (2011)
16. Souffiau, W., Maervoet, J., Vansteenwegen, P., Vanden Berghe, G., Van Oudheusden, D.: A Mobile Tourist Decision Support System for Small Footprint Devices. In: Cabestany, J., Sandoval, F., Prieto, A., Corchado, J.M. (eds.) IWANN 2009, Part I. LNCS, vol. 5517, pp. 1248–1255. Springer, Heidelberg (2009)
17. Souffriau, W., Vansteenwegen, P.: Tourist Trip Planning Functionalities: State–of–the–Art and Future. In: Daniel, F., Facca, F.M. (eds.) ICWE 2010. LNCS, vol. 6385, pp. 474–485. Springer, Heidelberg (2010)
18. Souffriau, W., Vansteenwegen, P., Vertommen, J., Vanden Berghe, G., Van Oudheusden, D.: A personalised tourist trip design algorithm for mobile tourist guides. Applied Artificial Intelligence 22(10), 964–985 (2008)
19. Sun, Y., Lee, L.: Agent-based personalized tourist route advice system. In: SPRS Congress Istanbul 2004, Proceedings of Commission II, pp. 319–324 (2004)
20. Tarapata, Z.: Selected multicriteria shortest path problems: An analysis of complexity, models and adaptation of standard algorithms. Int. J. Appl. Math. Comput. Sci. 17(2), 269–287 (2007)
21. Vansteenwegen, P., Souffriau, W., Vanden Berghe, G., Van Oudheusden, D.: A guided local search metaheuristic for the team orienteering problem. European Journal of Operational Research 196(1), 118–127 (2009)
22. Vansteenwegen, P., Souffriau, W., Vanden Berghe, G., Van Oudheusden, D.: Iterated local search for the team orienteering problem with time windows. Comput. Oper. Res. 36(12), 3281–3290 (2009)
23. Vansteenwegen, P., Van Oudheusden, D.: The Mobile Tourist Guide: an OR Opportunity. OR Insight 20(3), 21–27 (2007)
24. Zhan, B.F., Noon, C.E.: Shortest path algorithms: An evaluation using real road networks. Transportation Science 32(1), 65–73 (1998)