Personalized Routes for Mobile Tourism

Damianos Gavalas, Michael Kenteris, Charalampos Konstantopoulos, Grammati Pantziou

***Abstract*— This paper deals with the problem of deriving personalized recommendations for daily sightseeing itineraries for tourists visiting any destination. Our approach considers selected places of interest that a traveler would potentially wish to visit and derives a near-optimal itinerary for each day of visit; the places of potential interest are selected based on stated or implied user preferences. Our method enables the planning of customized daily personalized tourist itineraries considering user preferences, time available for visiting sights in daily basis, opening days of sights and average visiting times for these sights. Herein, we propose a heuristic solution to this problem and discuss its implementation aspects.**

***Index Terms*— Itinerary; Team Orienteering Problem; Route Planning, Maps, Mobile Tourism**

I. INTRODUCTION

TOURISTS that visit a destination for one or multiple days are unlikely to visit every tourist sight; rather, tourists are dealt with the dilemma of which points of interest (POIs) would be more interesting for them to visit. These choices are normally based on information gathered by tourists via the Internet, magazines, printed tourist guides, etc. After deciding of which sights to visit, tourists have to decide on which route to take, i.e. the order in which to visit each POI, with respect to the visiting time required for each POI, the POI's visiting days/hours and the time available for sightseeing in daily basis.

Tourists encounter many problems following this procedure. The information contained in printed guide books is often outdated (e.g. the opening times of some museums might have changed or some other memorial sites might be closed due to maintenance works, etc), the weather conditions might be prohibitive during one of the visiting days to visit an important POI, etc [6]. The selection of the most important and interesting POIs for visiting also requires fusion of information typically provided from separate -often non credible- sources. Usually tourists are satisfied if a fairly attractive or feasible route is derived, yet, they cannot know of any alternative routes which would potentially be better to follow. Some tourist guides do acknowledge such problems and try to propose more generalized tourist routes to a city or an area. Of course these routes are designed to satisfy the likes of the majority of its readers but not those with specialized interests, needs or constraints [3].

Mobile tourist guides may be used as tools to offer solution to these types of problems [9],[4],[13]. Based on a list of personal interests, up-to-date information for the sight and information about the visit (e.g. date of arrival and departure, accommodation address, etc), a mobile guide can suggest near-optimal and feasible routes that include visits to a series of sights, as well as recommending the order of each sight's visit along the route [17]. Generalized tourist routes do not take into consideration the context of the user e.g. the starting or ending point of the user, the available time the user affords, the current time, predicted weather conditions while on journey, etc. Taking into account the parameters of context and location awareness brings forward a challenge for the design of appropriate tourist routes [12]. Kramer et al. [11] analyzed the interests in the profiles of each tourist and concluded that they particularly varied from each other. This conclusion supports the argumentation for deriving personalized instead of generalized tourist routes.

Given a list of sights of some tourist destination in which a user-tourist would potentially be interested in visiting, the problem involves deriving the order in which the tourist should visit the selected POIs, for each day the tourist stays at that destination. We term this problem as the 'tourist itinerary design problem' (TIDP). Interestingly, the TIDP presents similarities to problems which have arisen in the past in the field of operational research; such problems reside upon the mathematical theory of graphs (graph theory) and comprise variations of the well-known travelling salesman problem (TSP).

For instance, the team orienteering problem (TOP) appoints an initial and final point as well as N points for visiting, where each point is associated with a 'score' or 'profit'. Given a particular time margin for each of the M team members, the TOP determines M routes (from the initial to the end point) via a subset of N points, aiming at maximizing the overall profit of visited points [2]. The TOP cannot be solved in polynomial time (NP-complete) [15], hence heuristics deriving near-optimal solutions are the only realistic way to tackle such problems, especially when considering online applications. TOP can be thought of as a starting point to model TIDP whereby the M team members are reduced to the number of days available for the tourist to stay and the profit of a sight signifies the potential interest (or degree of satisfaction) of a

Manuscript received April 22, 2010.

D. Gavalas and M. Kenteris are with the Department of Cultural Technology and Communication, University of the Aegean, Mytilene, Greece (phone: +302251036643; e-mail: {dgavalas@aegean.gr, m.kenteris@ct.aegean.gr}.

C. Konstantopoulos is with the Department of Informatics, University of Piraeus, Piraeus, Greece (email: konstant@unipi.gr).

G. Pantziou is with the Department of Informatics, Technological Educational Institution of Athens, Athens, Greece (email: pantziou@teiath.gr).

particular tourist visiting the POI within a given time span available for sightseeing daily (therefore, TOP considers the time spent while visiting each POI as well as the time needed to travel from one POI to another).

Nevertheless, TOP does not take into consideration the POIs' visiting days and hours. Therein, the resemblance of TIDP with another operational research problem (travelling salesman problem with time windows, TSPTW) [5] comes forward. TSPTW concerns the minimum cost path for a vehicle which visits a set of nodes. Each node must be visited only once and the visit must be carried out inside an allowed time interval (time window). The correlation of time windows with the POIs visiting days/hours is obvious. However, TSPTW involves planning of only one route (i.e. not M, as many as days available to the tourist to visit POIs), while it requires the vehicle to visit the whole set of nodes. A generalization of TOP and TSPTW is referred to as team orienting problem with time windows (TOPTW) [16] and considers multiple vehicles (i.e. itineraries) that should visit a subset of nodes, each within its allowed time window.

The issue of personalized tourist itineraries has not been looked at in the electronic and mobile tourism literature, with the exception of the algorithms proposed in [14] and [15]. In [14], Souffriau et al. proposed a heuristic solution for the orienteering problem, i.e. they only consider a single tourist itinerary. The algorithm presented in [15] deals with TOPTW; however, it does not take into account neither the opening days of sites nor the time needed to visit a sight, i.e. it makes the unrealistic assumption of zero visiting duration.

The main contribution of this paper lies in modeling and investigating a generalization of TOPTW through introducing a novel heuristic that provides near-optimal solutions to TIDP: the Daily TouRist Itinerary Planning (DailyTRIP). It is noted that some preliminary ideas of our technique have also been presented in [10].

The remaining of this article is organized as follows: The modeling, design and implementation of DailyTRIP are presented in Sections II, III and IV, respectively, while Section V draws conclusions and grounds for future work.

II. DailyTRIP Modeling

DailyTRIP modeling involves the definition and the description of the user model, visit model and the sight (POI) model taking into consideration parameters/ constraints like those listed below:

- User Model:
  - device (e.g. screen resolution, available storage space, processing power, etc);
  - language of content, localization;
  - personal 'demographic' data (e.g. age, educational level);
  - interests (explicit declaration or implicitly collected);
  - disability (e.g. blind, deaf, kinetic disability);
  - budget threshold willing to spend on sightseeing.
- Visit Model:
  - geographical location of accommodation;
  - period of stay (arrival and departure date);
  - time constraints (e.g. available time each day to tour, number and duration of desirable breaks, etc);
  - means of travel (e.g. walking, driving, bus, metro, etc).
- Sight (POI) Model:
  - category (e.g. museum, archaeological site, monument, etc);
  - available multimedia resources (collection of texts, video, audio, etc, localized in different languages;
  - geographical position (coordinates);
  - weight or 'objective' importance (e.g. the Acropolis of Athens is thought to be 'objectively' more important of the Coin Museum of Athens, hence the Acropolis is assigned a larger weight);
  - average duration of visit (e.g. the Archaeological Museum of Athens typically takes longer to visit than the city's Coin Museum due to size difference and the nature of exhibition);
  - rating/comments of users;
  - opening days/hours (time windows), which could be provided by the web service of an administrative body or the Ministry of Culture;
  - whether it is a indoor/outdoor site;
  - whether it is a accessible from people with disabilities;
  - admission price (ticket prices).

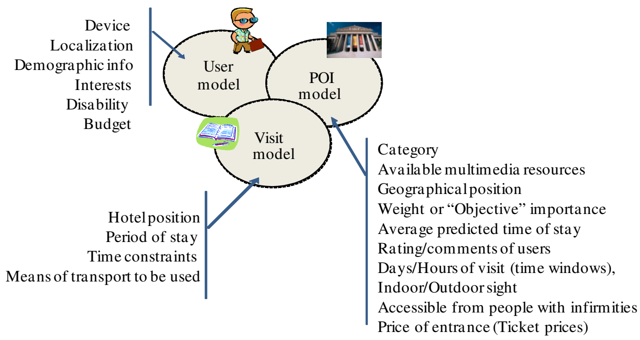


Figure 1. Description of user, visit and sight models in TIDP.

Notably, the above stated parameters/constraints are not exhaustive. From those parameters the below listed elements may be easily derived:

- The topological distance (or Manhattan distance) among the POIs and also among the accommodation and the POIs, based on their geographical coordinates and the local map.
- The number of routes that must be generated are based upon the period of stay of the user at the tourist destination.
- The anticipated duration of visit of a user at a POI derives from the average duration and the user's potential interest (concluded by examining the user's profile).
- The ability to visit open air sites in a particular day during the user's visit, e.g. outdoor sites are not recommended to visit during a rainy day (meteorological forecasts can be retrieved from an Internet web service).

The problem's definition also includes the 'profit' of a POI, calculated as a weighted function of the objective and subjective importance of each POI (subjectivity refers to the users' individual preferences). Our algorithmic solution maximizes the overall profit, i.e. it enables the construction of personalized routes which include the most important (for each tourist) sights under specific constraints (opening hours, weather conditions, time available for sightseeing). The most crucial constraint in seeking sound algorithmic solutions is the daily time limit *T* which a tourist wishes to spend on visiting sights; the overall daily route duration (i.e. the sum of visiting times plus the overall time spent moving from a POI to another which is a function of the topological distance) should be kept below *T*.

III. DailyTRIP: A Heuristic for Deriving Near-Optimal Personalized Daily Tourist Itineraries

A. *Problem Statement*

The TIDP problem involves a complete graph $G=(V,E)$, $|V| = n$, where each node i, $i=0,\dots,n-1$, in V corresponds to a POI and each edge (i,j) in E corresponds to the shortest path (in terms of Manhattan distance $d_{i,j}$) linking individual POIs i and j.

Each POI $i \in V$ is associated with a weight w_i which denotes the 'objective' importance of the POI and a profit vaue p_i, which reflects the importance of that POI for a particular user and depends on her personal preferences. Each POI i is also associated with a set of days $D_c(i)$ when visiting is not feasible (e.g. Mondays and during some bank holidays) and the anticipated visit duration of the user at the POI $t_v(i)$; similarly to the profit, $t_v(i)$ also depends on the user's personal preferences (for instance, someone interested in archaeology is expected to take longer to visit an archaeological museum than others).

The cost of each edge (i,j) $c_{i,j}$, namely the cost of visiting j after visiting i, is a weighted function of travelling time from i to j $t_{i,j}$ (the latter depends on the Manhattan distance $d_{i,j}$ between i and j and the means of travel), the profit of the arriving node p_j and the duration of visit at the arriving node $t_v(j)$: $c_{i,j} = a_1 \cdot t_{i,j} - a_2 \cdot p_j + a_3 \cdot t_v(j)$, where a_1, a_2 and a_3 are weight coefficients. This formula signifies that being on node i, the next itinerary stop j has to be a node of relatively high profit that takes short to arrive and visit. Notably, $c_{i,j} \neq c_{j,i}$.

Travelers typically plan to visit the area for a set of days, D. Users also define a starting and ending time for their daily itineraries, Tstart and Tend, which denote what time she prefers to depart from his starting point S and arrive at her end point (destination) E. Hence, a daily time budget devoted to visiting sights may be easily calculated: T = Tend - Tstart. Without loss of generality, we assume that that the starting and end points of the |D| daily itineraries coincide, i.e. S ≡ E (typically these will coincide with the user's accommodation H).

Summarizing, the objective of DailyTRIP is to derive |D| itineraries Ii that maximize the overall profit $\sum_{i=1}^{|D|}\sum_{j=1}^{|I_i|} p_j$, ensuring that the time needed to complete each itinerary does not exceed the user-defined daily time budget T, i.e. $T(I_i) \leq T$.

B. *The DailyTRIP algorithm flow*

DailyTRIP comprises the following execution phases:

Phase 1: Definition of the problem's model

The first phase first involves the definition of problem's space, i.e. the nodes of G, the nodes' weight w_i and the travelling time matrix $t_{i,j}$ that denotes the time needed to travel between node pairs; notably, $t_{i,j} \neq t_{j,i}$, since the route $i \to j$ differs from the route $j \to i$ due to considering one-way roads. Taking into consideration personalization issues (e.g. in a simplified scenario, user preferences upon POIs' categories), the cost matrix (i.e. the cost values $c_{i,j}$ associated with the two-directional edges) as well as the nodes' profit p_i and visit duration $t_v(i)$ with respect to a specified user are also computed.

Phase 2: Reduction of the problem's space

The initial set of sights around the tourist destination is sorted in decreasing order of profit p, where the value of p mainly depends on its category (i.e. whether the POI is museum or an architectural monument) and the user's preference upon this category. To reduce the computational effort required to reach valid solutions (i.e. to reduce the problem's space) we discard:

- nodes (POIs) with profit p_i smaller than a threshold value p_{min}
- POIs located too far from the origin point H, i,e, every node v for which $t_{H,v} > t_{max}$, where t_{max} is an upper time limit (see Figure 2a).

An alternative approach would be to exclude the relatively low-profit POIs located far from H, i.e. exclude every POI i for which $a_1 * p_i - a_2 * d_{i.H} < t$, where a_1 and a_2 are weight coefficients and t a threshold value.

Phase 3: Selection of first daily itinerary nodes

DailyTRIP determines the |*D*| POIs that will be the first to include in the |*D*| daily itineraries I_i, where i=1.. |*D*|. We select the set of |*D*| nodes $\{N_i\}$, where i=1.. |*D*|, located furthest apart from one another, i.e. those for which the minimum distance from one another is the maximum among any other permutation of |*D*| nodes. For instance, in the example topology of Figure 2b, assuming that |*D*|=3, we select the nodes i, j and k that: $\max_{i,j,k} min\,\{d_{i,j}, d_{i,k}, d_{j,k}\}$. Then, the |*D*| daily itineraries are initialized, each incorporating one of those nodes: $I_i = \{N_i\}, \forall i = 1..|D|$.

Phase 4: Construction of itinerary trees

On each of the following algorithm's steps, itineraries I_i are considered interchangeably incorporating a new node N not yet included in any of the I_i. In particular, for each I_i, the candidate node N with the minimum connection cost $c_{j,N}$ to any of the nodes $j \in I_i$ joins I_i (through accepting the $j \to N$ edge), given that the daily time budget T condition is not violated for this itinerary. Notably, as the candidate node N may be connected to any of the I_i nodes (i.e. not necessarily to the edge nodes of the itinerary), I_i grows as a tree structure rather than a multipoint line. The time T_i corresponding to the completion of the itinerary I_i is calculated first by temporarily connecting H with the I_i node nearest to H, then converting the

I_i itinerary tree to a multipoint line (through a post-order tree traversal) and finally calculating: $T(I_i) = t_{H,1} + \sum_{k=1}^{|I_i|}(t_v(k) + d_{k,k+1})$. Namely, $I_i = I_i \cup N$, if $T(I_i) \leq T$.

Hence, on each step itineraries Ii grow, typically approaching the start/end point H, until no further insertion is feasible (see Figure 2c). Upon completion, each itinerary is connected to the 'hotel' node H, i.e. the edge j→N is accepted, where j is the itinerary's node nearest to H (see Figure 2d).

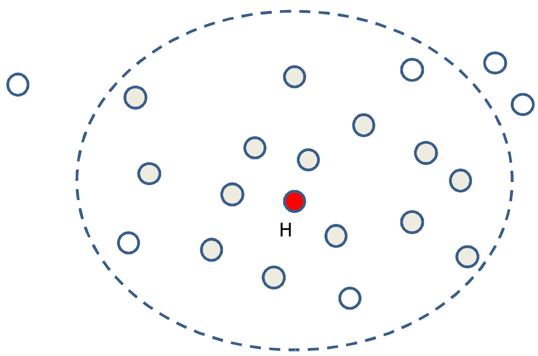
It is noted that the acceptance of candidate nodes also depends on the corresponding POIs' scheduled visiting days. In particular, for each joining node i that may not be visited during the days Dc(i), the 'excluded' days of the itinerary I joined by i is adapted excluding those days: $\mathrm{D_c(I)} = \bigcup_{\mathrm{i=1}}^{|\mathrm{I}|} \mathrm{D_c(i)}$, signifying that during those days the itinerary is not feasible either. Apparently, a POI i may join an itinerary I if the intersection of their valid days (i.e. those when visiting is feasible) is not null and also this intersection includes at least one of the D days of visit, namely if $\mathrm{D_v(I)} \cap \mathrm{D_v(i)} \cap \mathrm{D} \neq \emptyset$.

Phase 5: Rearrangement of itinerary trees

Phase 5 is optional and aims at improving the solutions derived in the previous phase, i.e. either increasing the overall profit or maintaining the same profit while reducing the itinerary completion time *T(I)* (see Figure 2d). Improved solutions are searched for every itinerary by: (a) substituting each itinerary tree node by any node not included in any itinerary at the end of the previous phase, (b) by swapping nodes included on different itineraries. In any case, the new itinerary solutions should satisfy the daily time budget constraint.

Phase 6: Traversal of itinerary trees

Notably, the outcome of the previous phases is not a set of itineraries, but rather a set of itinerary trees. Hence, the last phase of DailyTRIP involves the conversion of the |*D*| trees to multipoint lines I_i through a post-order traversal of the corresponding trees.



(a)

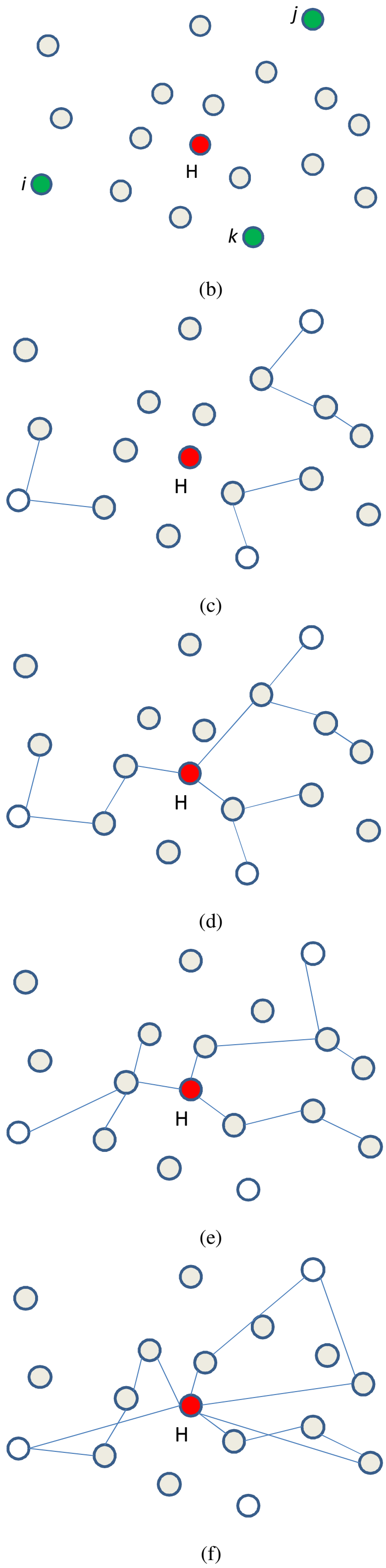


Figure 2. Execution phases of DailyTRIP.

IV. Implementation Details and Evaluation of DailyTRIP

DailyTRIP has been developed using JSP/MySQL web technologies and Google Maps as the main user interface. The user first provides some personal demographic data and preferences upon tourist content items, i.e. she may state preference in visiting museums, archaeological sites, monuments, etc. Further, she points the location *H* of her accommodation, the period of visit, the hours available for visit, the means of transport and the radius around the hotel she is willing to move in order to visit a POI. The user is then shown a list of the initially selected POIs based on her preferences, which she is allowed to modify adding/ removing POIs.

The algorithm filters the POIs left out from the problem's space (due to their distance from the user's accommodation, their incompatibility with the user's preferences or their intentional removal by the user) and populates the travelling time matrix $t_{i,j}$ for the remaining nodes through first computing the distance matrix entries and considering the average expected velocity v of the selected means of transport. Distances amongst pairs of nodes are found by means of using the shortest-route functionality of the Google Maps API [7] which refers to Manhattan distances and takes into account one-way roads.

Our implementation is based on the following assumptions: (a) each daily itinerary starts and ends at the same node, which coincides with the user's accommodation; (b) among all possible routes between a pair of nodes we only consider the shortest route in terms of length, although this might not be shortest in time; (c) the user is assumed to move with constant velocity regardless of the traversed edge or the time of day (admittedly, this is a valid assumption only for tourists walking around a city); (d) the POIs are assumed to be open for visiting during the hours available to the tourist for sightseeing.

The output of DailyTRIP is sketched on a Google Maps interface, with each itinerary drawn on separate screen and the order of visiting POIs denoted by the alphabetical order of characters representing POIs (see Figure 3). The maps derived by the web application are then converted to static images using the Google Static Maps API [8] in order to display on mobile phone screens.

Currently, we work on implementing the ILS algorithm [15]. Initial tests have shown that DailyTRIP derives better solutions in terms of the overall collected profit, while not significantly surpassed in terms of performance (time required to derive a valid solution). This is because ILS uses a 'greedy' approach [1] wherein the POI with higher profit is iteratively chosen to join an itinerary; thus, ILS fast spends the available daily time budget and thereby cannot afford to include important ('profitable') POIs within the itinerary solutions. On the other hand, DailyTRIP suggests a compromise in terms of performance and deriving improved solutions.

The high performance of DailyTRIP suggests it is suitable for online usage. In particular the algorithm requires less than 2,5 sec to derive a solution (excluding the time required to draw the solution on Google Maps) that deviates less than 7% from the optimal solution, considering problem spaces spanning up to 25 nodes.

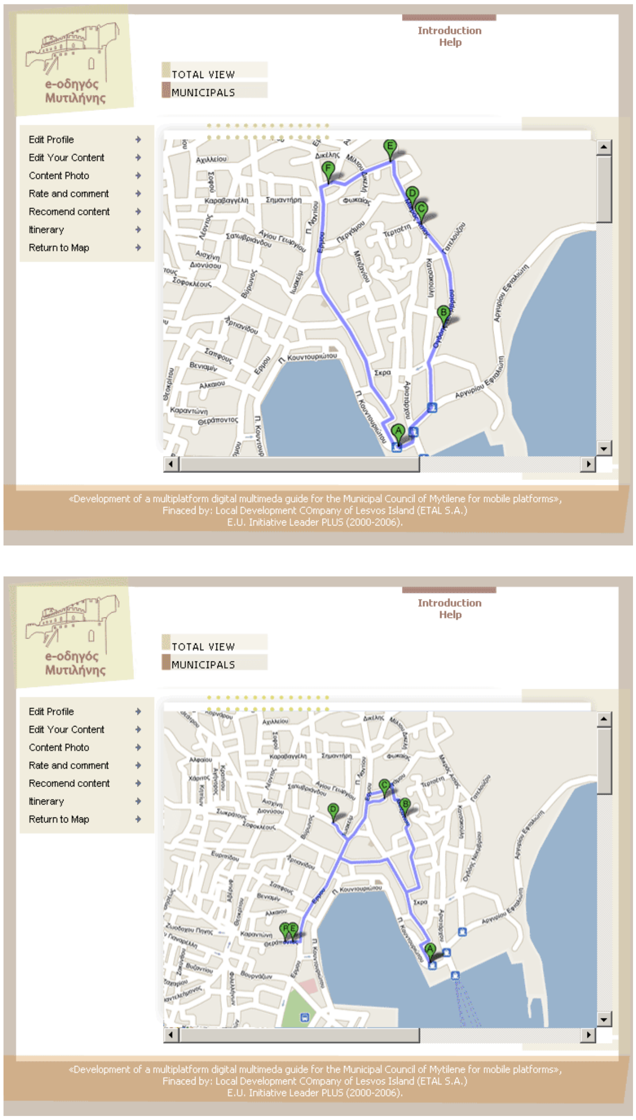


Figure 3. Output of DailyTRIP for two daily itineraries on Google Maps (point 'A' denotes the user's accomodation location, i.e. the start/end point of the two itineraries).

V. Conclusions and Future Research

This paper introduced DailyTRIP, a heuristic approach for deriving personalized recommendations of daily tourist itineraries for tourists visiting any tourist destination. DailyTRIP considers selected POIs that a traveller would potentially like to visit and derives a near-optimal itinerary for the traveller for each day of visit. Our approach takes into account user preferences, time available for visiting sights in daily basis, opening days of sites and average visiting times for these sites. The objective of DailyTRIP is to maximize the overall profit associated with visited POIs (where individual profits are calculated as a function of the POIs' 'objective' importance and the user's potential interest for the POI) while not violating the daily time budget for sightseeing. Our algorithm has been implemented and proved suitable for online applications (real-time design of itineraries).

Our future research will focus on variations of DailyTRIP algorithm that will incorporate additional TIDP problem parameters and constraints, e.g. weather conditions while on travel, financial budget (for transport and POIs admission charges), etc. We will also investigate the use of a combination of means of transport, e.g. walking and bus service, taking into account various aspects of alternative transport services (e.g walking time to the nearest metro station, day and time-dependent metro service frequencies, etc). Methods for fast itinerary updates will also be considered, wherein derived itineraries are subject to modifications due to sudden weather changes, taking longer than anticipated to visit or arrive at POIs, etc. Last, DailyTRIP will incorporate location-awareness, deriving itineraries that start at the user's current position.

REFERENCES

[1] Black P.E. (2005). Greedy algorithm. Dictionary of Algorithms and Data Structures, U.S.National Institute of Standards and Technology, http://www.itl.nist.gov/div897/sqg/dads/HTML/greedyalgo.html. Last accessed: December 10, 2010.

[2] Chao, I.-M., Golden B. L., Wasil E. A. (1996). The Team Orienteering Problem. European Journal of Operational Research, 88(3): 475-489.

[3] Cheverst K., Davies N., Mitchell K., Friday A., Efstratiou C. (2000). Developing a Context-Aware Electronic Tourist Guide: Some Issues and Experiences. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: 17-24.

[4] Cheverst K., Mitchell K., Davies N. (2002). The Role of Adaptive Hypermedia in a Context-Aware Tourist GUIDE. Communications of the ACM 45(5): 47-51.

[5] Dumas Y., Desrosiers J., Gelinas E., Solomon M. M. (1995). An Optimal Algorithm for the Traveling Salesman Problem with Time Windows. Operational Research, 43(2): 367-371.

[6] Dunlop M., Ptasinski P., Morrison A., McCallum S., Risbey C., Stewart F. (2004). Design and Development of Taeneb City Guide - From Paper Maps and Guidebooks to Electronic Guides. Proceedings of the International Conference Information and Communication Technologies in Tourism (ENTER'2004): 58-64.

[7] Google Maps. Google Maps API Concepts, http://code.google.com/apis/maps/documentation/. Last accessed: December 10, 2010.

[8] Google Static Maps API, http://code.google.com/apis/maps/documentation/staticmaps/. Last accessed: December 10, 2010.

[9] Kenteris M., Gavalas D., Economou D. (2009). An Innovative Mobile Electronic Tourist Guide Application. Personal and Ubiquitous Computing, 13(2):103-118.

[10] Kenteris M., Gavalas D., Pantziou G., Konstantopoulos C. (2010). Near-Optimal Personalized Daily Itineraries for a Mobile Tourist Guide, Proceedings of the 15th IEEE Symposium on Computers and Communications (ISCC'2010): 862-864.

[11] Kramer R., Modsching M., ten Hagen K. (2006). A City Guide Agent Creating and Adapting Individual Sightseeing Tours Based on Field Trial Results. International Journal of Computational Intelligence Research 2(2): 191-206.

[12] Krüger A., Malaka R. (2004). Artificial Intelligence Goes Mobile. Applied Artificial Intelligence, 18: 469-476.

[13] Malaka R., Zipf A. (2000). DEEP MAP - Challenging IT Research in the Framework of a Tourist Information System., Proceedings of the International Conference on Information and Communication Technologies in Tourism (ENTER'2000).

[14] Souffriau W., Maervoet J., Vansteenwegen P., Vanden Berghe G., Van Oudheusden D. (2009). A Mobile Tourist Decision Support System for Small Footprint Devices, Proceedings of the 10th International Work-Conference on Artificial Neural Networks (IWANN'2009): 1248-1255.

[15] Souffriau W., Vansteenwegen P., Vertommen J., Berghe G.V., Oudheusden D. (2008). A Personalized Tourist Trip Design Algorithm for Mobile Tourist Guides. Applied Artificial Intelligence, 22(10): 964-985.

[16] Vansteenwegen P. (2008). Planning in tourism and public transportation. PhD thesis, Katholieke Universiteit Leuven.

[17] Vansteenwegen, P., Van Oudheusden D. (2007). The Mobile Tourist Guide: An OR Opportunity. Operational Research Insight, 20(3): 21-27.