Pers Ubiquit Comput (2017) 21:137–155
DOI 10.1007/s00779-016-0971-3

ORIGINAL ARTICLE

Scenic route planning for tourists

Damianos Gavalas[1,6] · Vlasios Kasapakis[2,6] · Charalampos Konstantopoulos[3,6] · Grammati Pantziou[4,6] · Nikolaos Vathis[5,6]

Received: 21 March 2016 / Accepted: 31 August 2016 / Published online: 6 October 2016

Abstract Tourists visiting unknown destinations become increasingly dependent on mobile city guides to locate tourist services and retrieve informative content about nearby points of interest (POIs). Several mobile guides already support the provision of personalized tour recommendations to assist tourists in making feasible plans and visiting the most interesting POIs within their available time. However, existing tourist tour planners only regard available attractions as sites lacking physical dimensions (i.e., POIs are treated as 'points'). This restricts the modeling of POIs as attractions that may be entered/exited from a certain location (e.g., the main entrance). Although this is adequate for scheduling visits at museums, galleries, small squares or parks with single entry points, it fails to capture practical properties of typical tourist visiting styles in urban destinations. Tourists commonly appreciate strolling through pedestrian zones, market areas or urban areas of architectural, cultural and scenic value rather than only visiting sites of restricted access or taking the fastest route to move among city landmarks. Herein, we introduce *Scenic Athens*, a context-aware mobile city guide for Athens (Greece) which provides personalized tour planning services to tourists. Far beyond than just providing navigational aid, *Scenic Athens* derives near-optimal sequencing of POIs along recommended tours, taking into account a multitude of travel restrictions and POI properties, so as to best utilize time available for sightseeing. Unlike similar tools, our application incorporates scenic (walking) routes (in addition to point POIs), thereby supporting more experiential exploration of tourist destinations. This broader perception of tourist attractions substantially increases the complexity of the entailed optimization problem's modeling. A user evaluation study validated the recommendation value, usability and perceived utility of the proposed application.

Keywords Tourist trip design problem · Scenic routes · Orienteering problem · Time window · Context awareness · Web service · Mobile application · Android · Performance test · User evaluation

1 Introduction

Tour planning is a challenging task for individuals visiting unfamiliar urban destinations. Firstly, tourists need to narrow down to a potential set of points of interest (POI),

✉ Damianos Gavalas
dgavalas@aegean.gr

Vlasios Kasapakis
v.kasapakis@aegean.gr

Charalampos Konstantopoulos
konstant@unipi.gr

Grammati Pantziou
pantziou@teiath.gr

Nikolaos Vathis
nvathis@softlab.ntua.gr

1 Department of Product and Systems Design Engineering, University of the Aegean, Hermoúpolis, Syros, Greece

2 Department of Cultural Technology and Communication, University of the Aegean, Mytilene, Lesvos, Greece

3 Department of Informatics, University of Piraeus, Piraeus, Greece

4 Department of Informatics, Technological Educational Institution of Athens, Athens, Greece

5 School of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece

6 Computer Technology Institute and Press 'Diophantus' (CTI), Patras, Greece

among the many available, aligned with their personal interests and trip constraints. Thereafter, they are expected to allocate POIs among daily tours, decide upon a reasonable sequencing of POI visits along each tour and schedule transportation plans for moving from a POI to another [20].

Visits to museums, galleries, religious and archeological sites are certainly part of tourists' routine in urban destinations. However, field studies revealed that tourists seek to maximize the time spent wandering around the urban space, engaging all their body senses while 'on the move' [14, 19]. Unlike commuters or permanent city residents, most tourists would trade a time-efficient walking shortcut or transit transfer in favor of a more indirect, scenic or roundabout walking route that offers more opportunities for amorphous exploration and discovery [16]. Apart from offering a glance over everyday life activities, these walking routes allow strollers to appreciate the scenic value as well as the cultural and architectural elements of historical districts.

Today, several ICT tools exist assisting the way arounds of tourists, commonly in the form of mobile city guides [10], which may be used to locate tourist services and retrieve informative content about nearby POIs. Several mobile city guides tackle the problem of tourist tour planning, commonly termed as tourist trip design problem (TTDP) [4]. TTDP refers to scheduling feasible plans for tourists interested in visiting multiple POIs. Solving the TTDP entails deriving daily tourist tours comprising ordered sets of POIs that match tourist preferences, thereby maximizing tourist satisfaction (typically termed 'profit'), while taking into account a multitude of parameters and constraints (e.g., distances among POIs, time estimated for visiting each POI, POIs' opening hours) and respecting the time available for sightseeing in daily basis.

However, existing Web/mobile tourist tour planners adopt a narrow notion of POIs as sites which lack physical dimensions (i.e., POIs are treated as 'points'). Their recommended tours typically comprise visits at museums, galleries, religious monuments, castles, small squares or parks with single entry points. When including visits to market areas or scenic neighborhoods, they, again, regard those attractions as points and assume the users to start and end their visit at a central location in order to continue on to their next visit. Besides not advising specific walking paths, the above-mentioned assumption is certainly unrealistic especially for relatively large areas or elongated routes (e.g., long riversides) especially because the starting/ending points may be far located one from one another and the return path to the central location does not offer any 'profit' to the tourist.

Herein, we propose an elaborate TTDP modeling which captures practical properties of typical tourist visiting styles allowing to schedule both visits to 'point' POIs and walking routes through pedestrian zones, market areas or districts of architectural, cultural and scenic value (see Fig. 1). Hereafter, we will use the term 'POI' to refer to point attractions and the term 'scenic route' to refer to walking paths of tourist value. We have designed a metaheuristic algorithm which solves this TTDP formulation deriving high-quality solutions and satisfies the real-time requirements of Web/mobile tour planners even for large datasets. Note that, in addition to highlighting walking routes of touristic importance, our formulation may also serve for preventing tourists from passing through tourist unfriendly (or outright dangerous) parts of a city. This novel algorithmic approach is used in *Scenic Athens*, a context-aware mobile city guide for Athens (Greece) which provides personalized tour planning services and supports the experiential exploration of tourist destinations. *Scenic Athens* offers a variety of visual means to display the recommended tours and also provides directions to move from a POI (or the end point of a scenic route) to the next scheduled attraction. It is noted that a preliminary version of this work appears in [6].

The remainder of this article is structured as follows: Sect. 2 reviews relevant approaches both in the algorithmic domain and in the tourist tour planning software tools. Section 3 presents our tour planning algorithm. Section 4 discusses the system implementation details of *Scenic Athens* as well as results compiled from extensive performance tests. Section 5 reports the methodology and findings of a user evaluation study. Section 6 concludes our work and suggests directions for future research.

2 Related work

2.1 Algorithmic approaches to TTDP

TTDP has received considerable attention in the recent years with several mainly heuristic algorithmic methods proposed to solve it [4]. Those methods approach the problem from different perspectives, resulting in diverse problem models, which consider different problem variables and constraints. The input parameters considered in most TTDP models are: a set of candidate POIs, each associated with a number of attributes (e.g., type, location, opening days/hours and entrance fee); the number of tours to be generated, based upon the period of stay of the user at the tourist destination; the 'profit' of each POI, denoting its relevant importance; the anticipated visit duration of an average user at a POI; the travel times among POIs, which are constant when considering exclusively walking transfers; the daily time budget B that a tourist wishes to spend on visiting sights (the overall daily tour duration, i.e., the sum of visiting times plus the overall time spent for moving from a POI to another, should be kept below B).

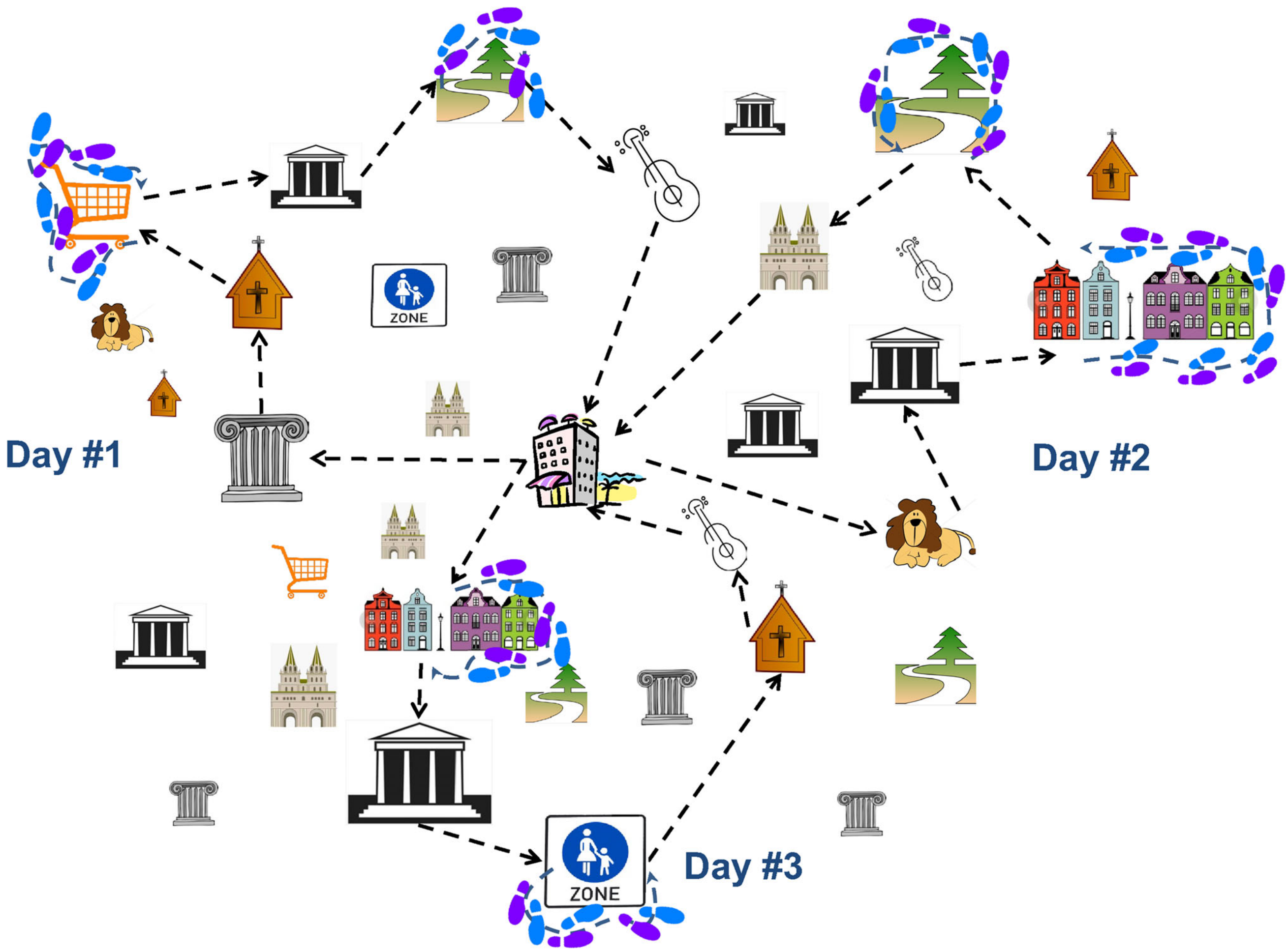

Fig. 1 Illustration of the TTDP, wherein the attractions included in the daily tours may be POIs or scenic routes (*icons* represent the categories of POIs or scenic routes, while their *size* indicates their profit)

The objective in TTDP modeling is to derive a set of near-optimal daily, disjoint itineraries (ordered visits to POIs), each comprising a subset of available (candidate) POIs so as to maximize tourist satisfaction (i.e., the overall collected profit); the derived tours should respect user constraints/POI attributes and satisfy the daily time budget available for sightseeing. Note that several problem parameters may be adapted according to user preferences. For instance, POI profits may be calculated as a weighted function of the objective and subjective importance of each POI (subjectivity refers to the users' individual preferences and interests on specific POI categories). Similarly, the time spent for visiting a POI derives from its average visiting duration and the user's potential interest for that particular POI.

The baseline combinatorial optimization problem for TTDP is the orienteering problem (OP) [23]. In the OP, given a starting node s, a terminal node t and a positive time limit (budget) B, the goal is to find a path from s to t (or tour if $s \equiv t$) with length at most B such that the total profit of the visited nodes is maximized.

Clearly, the OP may be used to model the simplest version of the TTDP wherein the POIs are associated with a profit (i.e., user satisfaction) and the goal is to find a single tour that maximizes the profit collected within a given time budget (time allowed for sightseeing in a single day). Extensions of the OP have been successfully applied to model more complex versions of TTDP. The OP with time windows (OPTW) considers visits to locations within a predefined time window; this allows modeling opening days/hours of POIs. The time-dependent OP (TDOP) considers time dependency in the estimation of time required to move from one location to another; therefore, it is suitable for modeling multimodal transports among POIs. The team orienteering problem (TOP) is the extension of the OP to multiple tours.

The TOP with time windows (TOPTW) has been mostly commonly studied among the aforementioned OP variants [9, 12, 22], since it is useful for modeling several real-life optimization problems. Given the complexity of the problem, the TOPTW literature mostly involves metaheuristics that involve, (a) an insertion step (adds a visit to one of the m tours) iteratively performed until the solution (or a set of solutions) cannot be further improved and (b) a diversification step that aims at escaping from local optima. Those two steps are repeated until a termination criterion is met. The iterated local search (ILS) algorithm [22] is considered most suitable for real-time TTDP applications among alternative TOPTW algorithms as it represents a fair compromise in terms of speed (less than 7 s for up to 200 POIs and $k = 4$ daily tours) versus deriving routes of reasonable quality (on average, less than 5 % gap from the best known solution).

The arc orienteering problem (AOP), introduced by Souffriau et al. [21], is the arc routing version of the OP and is applicable to TTDP variants whose modeling requires profits to be associated with the arcs (instead of the nodes) of the network as some links may be more beneficial to be traversed than others. For instance, arc values could indicate the scenic value or the gradient (i.e., the difficulty to walk) of a route segment.

The combination of the OP and the AOP has been proposed by [23] under the name mixed orienteering problem (MOP). In the MOP, profits are associated with the nodes as well as with the arcs of the graph. Therefore, it can be used to formulate TTDP variants wherein, further to typical attractions, certain routes may be of tourist interest. In this paper, we formulate our optimization problem as a mixed team orienteering problem with time windows (MTOPTW), i.e., the extension of the MOP to multiple tours wherein nodes may be visited (and arcs may be traversed) within a time window. Due to its apparent hardness and the real-time requirements of TTDP applications, we employ an ILS metaheuristic approach to tackle MTOPTW so as to derive approximate solutions relatively fast. The ILS metaheuristic has been introduced in [7] and has been found to be more efficient than an alternative approach based on simulated annealing.

2.2 Web and mobile TTDP solvers

Web and mobile tourist assistant tools have proliferated in the last few years offering a variety of services spanning from vacation planning to mobile tourist guides [10] and tourism recommender systems [3, 18]. Among them, four Web/mobile prototypes are known to offer tour planning services: CT-Planner4,[1] CityTripPlanner,[2] TripBuilder[3] and eCOMPASS.[4] These tools, essentially TTDP solvers, automate the creation of a single or multiple tours via a set of POIs taking into account their respective profit, visiting time and opening hours as well as the walking travel times among POIs and the trip details (visiting days, start/end times). The derived tours are personalized, i.e., they are tuned according to user-defined preferences.

CT-Planner4 [11] is a Web-based tourist tour planner for seven (7) Japanese cities. Recommended tours are personalized with respect to: (a) user focus (e.g., entertainment, culture, nature) and taste (e.g., less known vs. well known or calm vs. energetic), which adjust POI profits and (b) preferred moving speed and reluctance to walk, which adjust walking travel times. The tour planning engine of CT-Planner4 relies on genetic algorithm which solves the selective traveling salesman problem (STSP) [13], i.e., it derives a single tourist itinerary. The solver starts with n random initial tours and ends up to a tour plan with the highest utility score (i.e., profit) through iteratively performing a crossover/mutation procedure.

CityTripPlanner [24] is a Web/mobile city tour planner which already covers 76 destinations across all continents. Recommended tours are displayed on a map and list view. The user is allowed to edit derived tours, remove unwanted POIs and/or adjust visiting time scheduled for particular POIs (in all cases, edits trigger recalculation of tours). The start/end locations may be selected among a fixed set of hotels (i.e., user's lodging) and landmarks. CityTripPlanner is known to utilize the ILS TOPTW algorithm [22] combined with a GRASP heuristic as its core tour planning engine.

Brilhante et al. [1] presented TripBuilder, a Web tool which employs a two-step process upon POI collections retrieved from the Wikipedia and albums of geo-referenced photos from Flickr. Firstly, the step of deriving an optimal set of POIs (based on POI popularity and user preferences) is modeled as an instance of the generalized maximum coverage problem [2]. Then, the selected POIs are sequenced along a sightseeing itinerary by a heuristic algorithm addressing an instance of the traveling salesman problem.

eCOMPASS [5] is a Web/mobile tourist tour planner which considers the option of using public transit for moving around. Namely, eCOMPASS incorporates multimodality (i.e., time dependency) within its routing logic aiming at deriving near-optimal sequencing of POIs so as to minimize waiting time at transit stops. Along this line, eCOMPASS utilizes an efficient heuristic which tackles the time-dependent TOPTW (TDTOPTW). eCOMPASS also allows users to define arbitrary start/end locations for their daily tours and assists scheduling lunch breaks at restaurants conveniently located along the recommended tours.

The aforementioned city tour planning software tools already incorporate an array of useful services. However, they all regard tourist tours as a sequence of nonstop visits to 'point' POIs of restricted or free access and assume that tourists always take the shortest walking or transit route to move from a POI to another. Nevertheless, they neglect the

[1] http://ctplanner.jp/ctp4/index-e.html.

[2] http://www.citytripplanner.com/.

[3] http://tripbuilder.isti.cnr.it/.

[4] http://ecompass.aegean.gr/.

scheduling of walking (scenic) routes in between POI visits, hence compromising their utility in realistic tourist visiting styles. The *Scenic Athens* application introduced in this article addresses this issue, taking into account scenic routes in tour planning, thereby allowing tourists to make the best out of their available sightseeing time. We adopt a generic model for scenic routes, which includes a time window. This property allows to model routes whose value varies throughout the day or the week. For instance, market areas could be assigned working days/hours time windows. Similarly, the scenic value of walk along the riverside would be more appreciated under daylight. Besides, time windows (or negative profit values) are also useful to denote dangerous routes/areas. Aside scenic route planning, *Scenic Athens* incorporates several interesting features: in addition to displaying tourist tours via a list and/or a map view, it offers augmented reality and Google street views on recommended attractions; route planning decisions optionally consider weather forecast for the trip dates (e.g., visits to museums could be scheduled on rainy days, leaving walking routes for clear-sky days); start/end locations of daily tours may be chosen by various means (user's accommodation, selected city landmarks, current position, arbitrary location pointed on a map interface); start/end location and time may vary among tours; different walking paces are supported (relaxed/medium/intense); recommended tours may be edited, while POIs and scenic routes may be rated so as to adjust their profit value.

3 The MTOPTW heuristic for scenic route planning

In this section, we introduce the mixed team orienteering problem with time windows (MTOPTW), i.e., the extension of the MOP to multiple tours. The problem models realistic TTDP variants where profits are associated not only with POIs (network nodes) but also with routes (network edges), as certain routes may be more attractive than others. Specifically, we consider a complete windy undirected graph[5] $G = (V, E)$, where $V = \{u_1, u_2, \ldots, u_N\}$ denotes the vertex set and E the edge set. A travel cost is assigned to each edge $\{u, v\}$, namely $T_{u,v}$ which might be different from $T_{v,u}$. Also, each vertex u is associated with a visit duration T_u and visiting a vertex u (or traversing an edge $\{u, v\}$) offers a profit P_u ($P_{u,v}$). For convenience, we denote a node or edge with positive profit as a tourist attraction (TA). Each vertex u (edge $\{u, v\}$) is associated with an opening time O_u^d ($O_{u,v}^d$) and a closing time C_u^d ($C_{u,v}^d$) for each different day of the week $d \in \{0, 1, \ldots, 6\}$. The visit at a vertex u or the traversal of an edge $\{u, v\}$ at a specific day d can only start after its opening time and should end before its closing time. Furthermore, K walks $W_0, W_1, \ldots, W_{K-1}$ should be obtained; sl_i and el_i denote the starting and ending location (i.e., vertex), while st_i and et_i denote the starting and ending time of each walk W_i, respectively. Specifically, each walk comprises an ordered set of vertices, which may either be nodes with positive profit (i.e., POIs) or end points of edges with positive profits (i.e., the scenic routes' ends): $W_i = \left(w_0^i, w_1^i, \ldots, w_{l_i-1}^i\right)$; it holds that $w_0^i = \mathrm{sl}_i$, $w_{l_i-1}^i = \mathrm{el}_i$ and the visit at each vertex w_m^i (the traversal of each edge $\{w_m^i, w_{m+1}^i\}$) satisfies its time window, i.e., the visit (traversal) starts after its opening time and ends before the closing time. The profit of the derived solution is equal to the sum of the profits of the visited vertices and the traversed edges. If a vertex is visited (or an edge is traversed) more than once, its profit is credited only once. The goal of the MTOPTW is to construct a feasible solution which maximizes the overall collected profit.

For solving MTOPTW, we propose an iterated local search [15] metaheuristic. Among other operations, the proposed metaheuristic repeatedly executes local search steps where the neighborhood of a solution should be examined in order to yield new solutions promising higher eventual profit. Thus, it is important to keep the size of this neighborhood relatively small so that each local search step can be executed fast. For that reason, the input graph is preprocessed and transformed to a graph on which the solutions obtained are equivalent to those on the original graph. Specifically, first, we separate all profitable nodes from adjacent profitable edges. For instance, in Fig. 2a, node u is adjacent to two profitable edges (i.e., it is the end point of two scenic routes as well as a TA itself) and two nonprofitable ones. In the new transformed graph (see Fig. 2b), we introduce a dummy node and edge u' and (u', u), respectively, with $P_{u'} = 0$, $T_{u'} = 0$, $(O_u^d, C_u^d) = (-\infty, +\infty)$ and $P_{u',u} = P_{u,u'} = 0$, $T_{u',u} = T_{u,u'} = 0$, $\left(O_{u,u'}^d, C_{u,u'}^d\right) = \left(O_{u',u}^d, C_{u',u}^d\right) = (-\infty, +\infty)$, again, for any day d. Now, the profitable edges are all connected to the dummy node while the ones with no profit are connected to the original node u. Clearly also, any solution on this new graph can be easily transformed to a solution on the original graph of exactly the same itinerary on the real nodes. Next, all adjacent profitable edges should be parted (see Fig. 2c, d) by inserting a corresponding number of dummy nodes and edges whose profit and time attributes are set as above. Admittedly, this preprocessing increases the number of edges and nodes in the graph; however, since the tourist attraction routes (edges) are relatively few in a graph modeling a metropolitan city, this increase will be

[5] G is called a windy graph, if G is an undirected graph and there are two costs associated with each edge, representing the cost of traversing it in each possible direction.

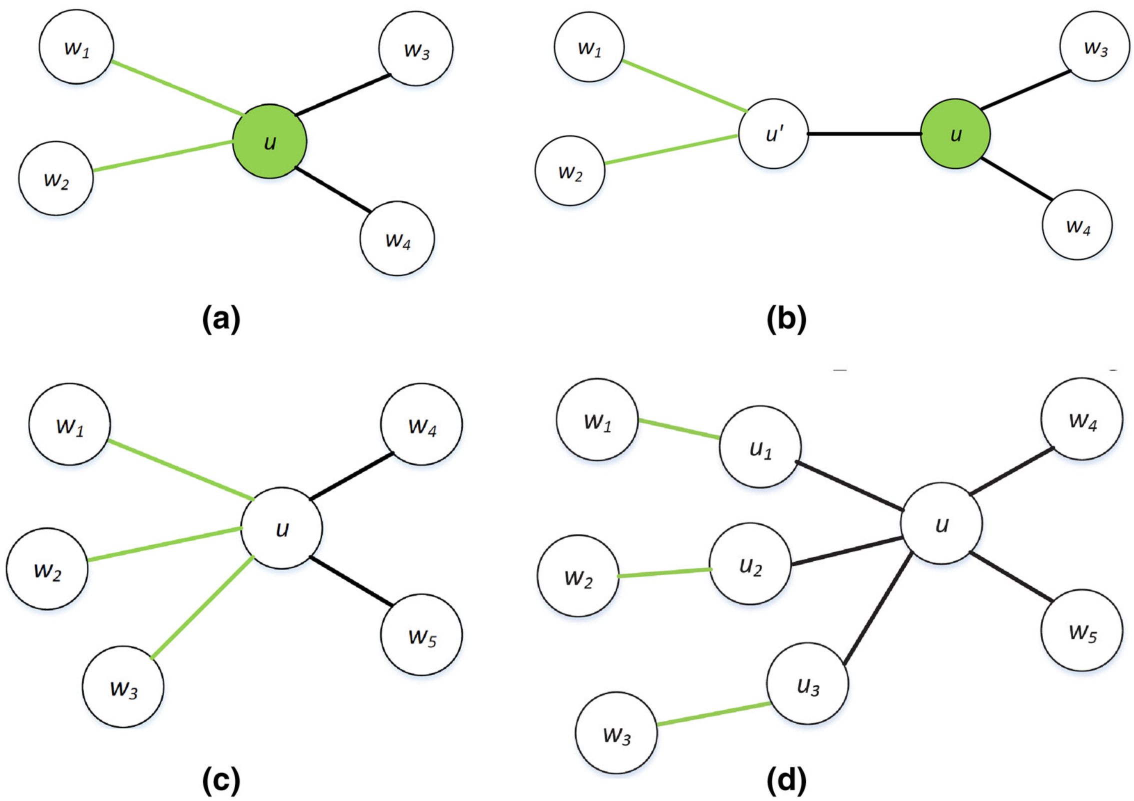

Fig. 2 Preprocessing steps (profitable edges or nodes are green colored): **a** profitable edges adjacent to a profitable node; **b** graph transformation eliminating the adjacency among profitable nodes and profitable edges; **c** profitable edges incident to the same node; **d** elimination of the adjacency among profitable edges

relatively small and easily offset by the simplification of solution neighborhoods in the local search steps.

After this preprocessing phase, the new graph is given as an input to the ILS metaheuristic. In the following, we give the details of the proposed metaheuristic.

3.1 The iterated local search metaheuristic

The iterated local search [15] is a well-known metaheuristic method where a sufficient number of local search/perturbation steps are applied until a near-optimal solution can be found. In the local search step, a neighboring solution of the current one can be obtained by inserting a nonincluded TA between two consecutive nodes of the walk which are not connected through a profitable edge. For instance, in Fig. 3, we consider the neighboring solutions of the single walk solution with node representation $W_i = (\mathrm{sl}_i, u, v, w, \mathrm{el}_i)$. We consider that the nonincluded TA is the edge $\{y, z\}$. Similarly, we can handle the case that the nonincluded TA is a node. Figure 3a–f shows the neighboring solutions produced when inserting the edge $\{y, z\}$ between two consecutive nodes of the walk. In Fig. 3a, b, $\{y, z\}$ is inserted between sl_i and u, considering both directions of this edge. Similarly, Fig. 4c, d depicts the insertion of $\{y, z\}$ between u and $\{v, w\}$ and in Fig. 3e, f, $\{y, z\}$ is inserted after $\{v, w\}$. Notice also that $\{y, z\}$ cannot be inserted between v and w, since $\{v, w\}$ is a profitable edge.

Inspired by [22], each node in a walk is associated with its arrival, starting and leaving (leave) time. For easily checking the feasibility of a new TA insertion in an existing walk, we also hold the latest time the arrival at each node can take place (maxArrive) so that the walk can remain feasible, that is no subsequent visit is canceled due to this time shift. Considering the walk $W_i = \left(w_0^i, w_1^i, \ldots, w_{l_i-1}^i\right)$ on day d, the time attributes of each included node are given by the recursive formulas:

$$\mathrm{arrive}\left(w_0^i\right) = \mathrm{start}\left(w_0^i\right) = \mathrm{leave}\left(w_0^i\right) = \mathrm{st}_i,$$
$$\text{and for each } k = 0, 1, \ldots, l_i - 1$$
$$\mathrm{arrive}\left(w_k^i\right) = \max\left(\mathrm{leave}\left(w_{k-1}^i\right), O^d_{w^i_{k-1}, w^i_k}\right) + T_{w^i_{k-1}, w^i_k},$$
$$\mathrm{start}\left(w_k^i\right) = \max\left(\mathrm{arrive}\left(w_k^i\right), O^d_{w^i_k}\right)$$
$$\text{and } \mathrm{leave}\left(w_k^i\right) = \mathrm{start}\left(w_k^i\right) + T_{w^i_k}.$$

The maxArrive time of the nodes is calculated recursively from the ending location to the start as follows:

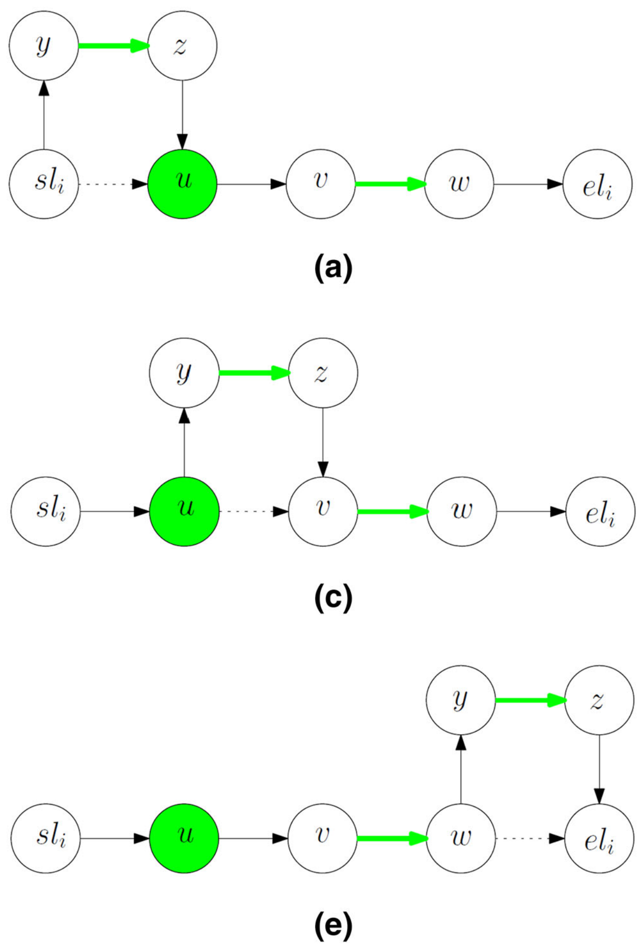

Fig. 3 Local search step

$$\mathrm{maxArrive}\left(w^i_{l_i-1}\right) = \mathrm{et}_i \quad \text{and} \quad \text{for each}$$
$$k = l_i - 2, l_i - 3, \ldots, 1, 0$$
$$\mathrm{maxArrive}\left(w^i_k\right) = \min\Big(C^d_{w^i_k}, C^d_{w^i_k, w^i_{k+1}} - T_{w^i_k}, \mathrm{maxArrive}\left(w^i_{k+1}\right) - T_{w^i_k} - T_{w^i_k, w^i_{k+1}}\Big)$$

For the estimation of the arrival time at a node w^i_k, we should take into account the leave time from the preceding node along the route, the travel time as well as the opening time of the route between these two nodes. Notice also that the starting time is the time when the actual visit at the node starts and may be different from the arrival time (the node may be closed at the time of arrival and one would wait until the opening time). Finally, the maximum allowed arrival time at the node w^i_k depends on its closing time, its visit duration, the closing time of the route leading to the next node along the walk, the travel time along that route, as well as the maximum allowed time of the next node which has been already estimated.

Based on these relations, the insertion of a new TA node or edge in a walk W_i can be quickly checked for feasibility. For instance, assume that a TA edge $\{y, z\}$ is tried for insertion in the walk between the two successive nodes w^i_k and w^i_{k+1}. Both traversal directions of the edge, i.e., from y to z and from z to y, should be examined. The insertion of $\{y, z\}$ with the direction from y to z after a node w^i_k is feasible if and only if the following conditions are satisfied:

1. leave $(w^i_k) \leq C^d_{w^i_k, y}$,
2. the arrival time at y is at most C^d_y,
3. the leaving time from y is at most $C^d_{y,z}$,
4. the arrival time at z is at most C^d_z,
5. the leaving time from z is at most $C^d_{z, w^i_{k+1}}$,
6. the new arrival time at w^i_{k+1} is at most $\mathrm{maxArrive}(w^i_{k+1})$.

Similar checks should be done for the tentative insertion of a new TA node. If the new (node or edge) insertion

Fig. 4 Screens from the *Scenic Athens* mobile (Android) application

is feasible, the difference between the new arrival time at w^i_{k+1} and the former one is the time shift of the arrival at w^i_{k+1}, that is $\text{shift} = \text{newArrive}(w^i_{k+1}) - \text{arrive}(w^i_{k+1})$. Notice also, that the insertion of a new TA in a walk always results in a positive time shift as this insertion is done over a non-TA edge whose corresponding route is a shortest path. In contrast, negative shifts would be possible, if the new insertion was allowed over an TA edge whose route is not necessarily the shortest between the ends of the edge.

Algorithm 1 implements the local search step for the insertion of a new TA at a specific walk W_i. Specifically, the local search technique considers all TAs not yet included in any walk. For each of these TAs, every possible insert position along W_i is examined and the position of the lowest shift is kept. Also, for all candidate TA edges, both directions are checked for insertion. Then, the insertion in W_i achieving the highest ratio of profit over shift is performed. After this insertion, the time attributes of some of the nodes along W_i may have been

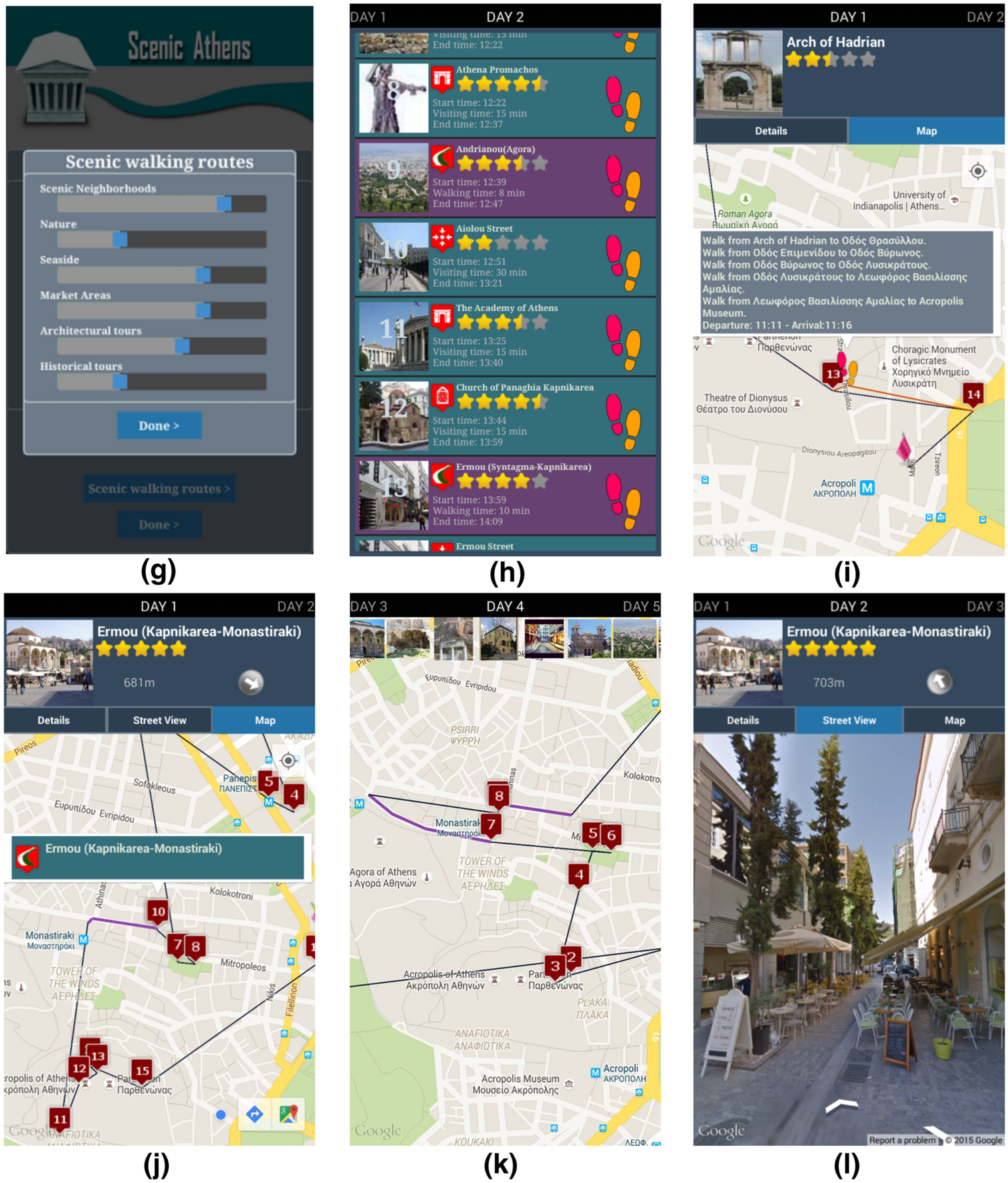

Fig. 4 continued

changed and new values are estimated again only for these nodes.

Overall, the ILS (Algorithm 2) loops over a number of iterations. In each iteration, all walks are repeatedly extended until no further insertion is possible. Also, after each solution improvement, the best-so-far solution is updated accordingly. For escaping from the local optima, the last step in each iteration is the perturb step where a randomly selected chain of consecutive TAs is removed from each walk of the solution.

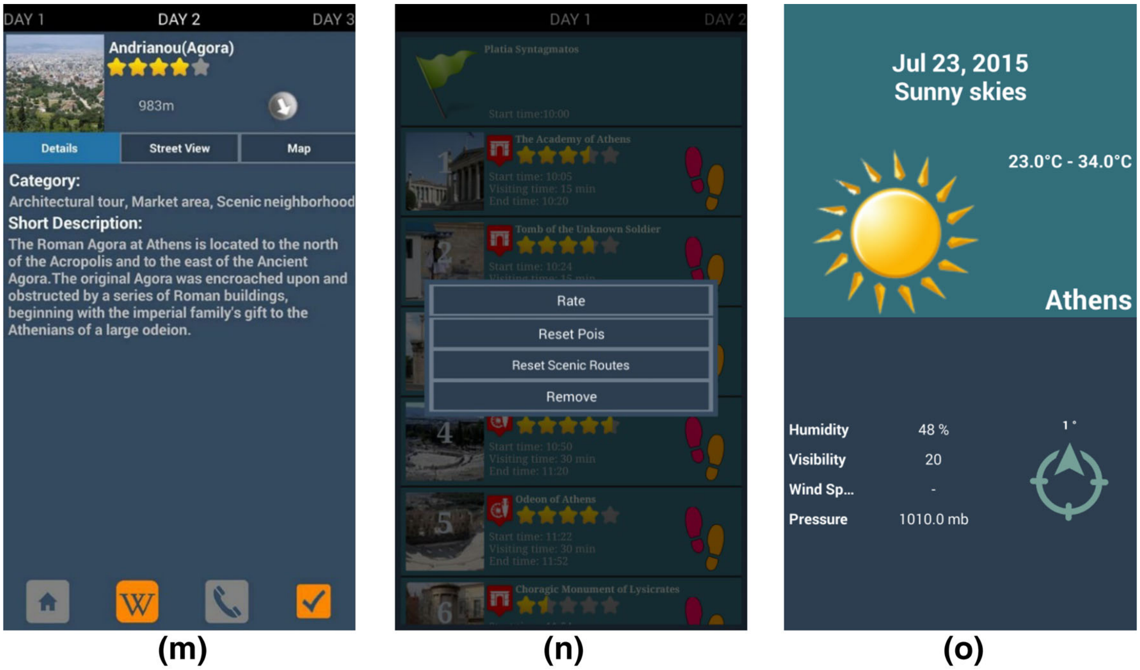

Fig. 4 continued

Algorithm 1 insertBestTAIn (W_i)

BestInsertionPoint ← ∅
BestTA ← ∅
BestRatio ← $-\infty$
for each candidate TA cp do
  TempLowestShift ← ∞
  TempBestInsertionPoint ← ∅
  for each included TA ip in W_i do
    if the insertion of cp after ip is feasible then
      if shift < TempLowestShift then
        TempLowestShift ← shift
        TempBestInsertionPoint ← ip
  if $\frac{profit_{cp}}{TempLowestShift} > BestRatio$ then
    BestInsertionPoint ← TempBestInsertionPoint
    BestTA ← cp
    BestRatio ← $\frac{profit_{cp}}{TempLowestShift}$
Insert the BestTA after BestInsertionPoint in W_i
Update *arrive(w), start(w), leave(w), maxArrive(w)* of each node *w* along W_i, if needed

Algorithm 2 The ILS Metaheuristic

$W \leftarrow \{W_0, W_1, \ldots, W_{K-1}\}$ the set of Walks to be determined
$DelW \leftarrow \emptyset$
for numOfIters do
  while $W \neq \emptyset$ do
    for each W_i in W do
      if insertBestTAIn(W_i) is feasible then
        if total solution profit after insertion > bestProfit then
          bestProfit ← total solution profit after insertion
          bestSolution ← current solution
      else $DelW \leftarrow DelW \cup \{W_i\}$
    $W \leftarrow W - DelW$
  perturb()
return bestSolution;bestProfit

3.2 Example execution of the iterated local search metaheuristic

In the following, we give the details of determining the best insertion point of the edge $\{y, z\}$ in the route in Fig. 3. There are three possibilities for inserting the edge $\{y, z\}$ as depicted in Fig. 3a–c. Table 1 lists the values of time parameters assumed in this example.

Before the insertion of the edge $\{y, z\}$ in this particular route, the arrival, start and leave time of each node along the route are calculated as follows:

$$
\begin{aligned}
\mathrm{arrive}(\mathrm{sl}_i) &= \mathrm{start}(\mathrm{sl}_i) = \mathrm{leave}(\mathrm{sl}_i) = 10{:}00,\\
\mathrm{arrive}(u) &= \max\{\mathrm{leave}(\mathrm{sl}_i), O_{\mathrm{sl}_i,u}\} + T_{\mathrm{sl}_i,u} = 10{:}10,\\
\mathrm{start}(u) &= \max\{\mathrm{arrive}(u), O_u\} = 10{:}30,\\
\mathrm{leave}(u) &= \mathrm{start}(u) + T_u = 10{:}50,\\
\mathrm{arrive}(v) &= \max\{\mathrm{leave}(u), O_{u,v}\} + T_{uv} = 10{:}55,\\
\mathrm{start}(v) &= \mathrm{leave}(v) = \mathrm{arrive}(v) = 10{:}55,\\
\mathrm{arrive}(w) &= \max\{\mathrm{leave}(v), O_{v,w}\} + T_{v,w} = 11{:}10,\\
\mathrm{start}(w) &= \mathrm{leave}(w) = \mathrm{arrive}(w) = 11{:}10,\\
\mathrm{arrive}(\mathrm{el}_i) &= \max\{\mathrm{leave}(w), O_{w,\mathrm{el}_i}\} + T_{w,\mathrm{el}_i} = 12{:}00.
\end{aligned}
$$

The maxArrive values are calculated starting from the end location and iterating backwards along the route:

Table 1 Example time parameters

Parameter	Value
st_i	10:00
et_i	14:00
$T_{\mathrm{sl}_i,u}$	10 min
$T_{u,v}$	5 min
$T_{v,w}$	15 min
T_{w,el_i}	50 min
$T_{y,z}$	30 min
T_u	20 min
T_v, T_w, T_{sl_i}, T_{el_i}	0 min
(O_u, C_u)	(10:30, 13:00)
(O_v, C_v), (O_w, C_w)	$(-\infty, +\infty)$
$(O_{\mathrm{sl}_i}, C_{\mathrm{sl}_i})$, $(O_{\mathrm{el}_i}, C_{\mathrm{el}_i})$	(10:00, 14:00)
$(O_{\mathrm{sl}_i,u}, C_{\mathrm{sl}_i,u})$, $(O_{u,v}, C_{u,v})$, $(O_{w,\mathrm{el}_i}, C_{w,\mathrm{el}_i})$	$(-\infty, +\infty)$
$(O_{v,w}, C_{v,w})$	(10:00, 18:00)
$(O_{y,z}, C_{y,z})$	(11:30, 13:00)
$(T_{\mathrm{sl}_i,y}, T_{u,y}, T_{w,y})$	(60, 50, 30 min)
$(T_{z,u}, T_{z,v}, T_{z,\mathrm{el}_i})$	(55, 45, 25 min)
$(O_{\mathrm{sl}_i,y}, C_{\mathrm{sl}_i,y})$, $(O_{u,y}, C_{u,y})$, $(O_{w,y}, C_{w,y})$	$(-\infty, +\infty)$
$(O_{z,u}, C_{z,u})$, $(O_{z,v}, C_{z,v})$, $(O_{z,\mathrm{el}_i}, C_{z,\mathrm{el}_i})$	$(-\infty, +\infty)$

Table 2 Time parameter values when inserting the edge $\{y, z\}$ is between nodes u and v

	sl_i	u	y	z	v	w	el_i
Arrive	10:00	10:10	11:40	12:10	12:55	13:10	14:00
Start	10:00	10:30	11:40	12:10	12:55	13:10	14:00
Leave	10:00	10:50	11:40	12:10	12:55	13:10	14:00
maxArrive	10:40	10:50	11:40	12:10	12:55	13:10	14:00

$$\begin{aligned}\mathrm{maxArrive}(\mathrm{el}_i) &= C_{\mathrm{el}_i} = 14{:}00,\\ \mathrm{maxArrive}(w) &= \min\{C_w, C_{w,\mathrm{el}_i} - T_w, \mathrm{maxArrive}(\mathrm{el}_i)\\ &\quad - T_w - T_{w,\mathrm{el}_i}\} = 13{:}10,\end{aligned}$$

Similarly, $\mathrm{maxArrive}(v) = 12{:}55$, $\mathrm{maxArrive}(u) = 12{:}35$, $\mathrm{maxArrive}(\mathrm{sl}_i) = 12{:}25$.

The first tentative insertion of edge $\{y, z\}$ is between the nodes sl_i and u. However, this insertion turns out to be unfeasible. The arrival time of nodes u, v, w and el_i is affected by this insertion. Specifically, the new arrival time at node u is 12:45 which is larger than maxArrive(u) (=12:35); therefore, this insertion is not feasible.

The second attempt for inserting the edge $\{y, z\}$ is between the nodes u, v. The new values of time parameters due to this insertion are shown in Table 2.

It can be easily seen that this insertion is feasible since visiting of each node or edge take places between the opening and closing time of the node or edge and before the maxArrive time of the node. The shift value for this insertion is given by the difference new value of arrive(v)—previous value of arrive(v), namely 120 min (=12:55–10:55).

Table 3 Time parameter values when inserting the edge $\{y, z\}$ is between nodes w and el_i

	sl_i	u	v	w	y	z	el_i
Arrive	10:00	10:10	10:55	11:10	11:40	12:10	12:35
Start	10:00	10:30	10:55	11:10	11:40	12:10	12:35
Leave	10:00	10:50	10:55	11:10	11:40	12:10	12:35
maxArrive	11:40	11:50	12:15	12:30	13:00	13:35	14:00

The next insertion of $\{y, z\}$ over the edge $\{v, w\}$ is not permitted since the latter is a profitable edge. Thus, the last possibility is to insert the edge $\{y, z\}$ between the nodes w and el_i. The parameter values are shown in Table 3.

The insertion is feasible, and the shift value is 35 min which is the difference between the new and old arrival time at node el_i (=12:35–12:00). Apparently, this is the lowest shift value derived by this sequence of insertions, and thus, the edge $\{y, z\}$ is finally inserted between the nodes w and el_i.

4 Implementation

4.1 Content database

We have compiled a collection from the urban area of Athens (Greece), which comprises 113 attractions (POIs), 18 scenic routes and 100 hotels. POIs are classified in the following categories: museums and art galleries, parks, archeological sites, squares, churches and religious heritage, monuments and landmarks. The metadata stored for each POI include: title; geo-coordinates (latitude, longitude); category; profit; visiting time; opening/closing hours for each week day; indication whether it is 'open air' (i.e., unsuitable to visit in rainy or heat wave days) or not; entrance fee for adults and children; indication for accessibility facilities; short description; photograph(s); address; telephone; official Web site URL; Wikipedia entry URL; average user rating; overall number of uploaded user ratings.

Scenic routes are classified into scenic neighborhoods, market areas, architectural tours, historical tours, nature and seaside. The metadata stored for each of the scenic routes include title; category; Wikipedia entry URL; path coordinates; short description; photograph(s); average user rating; and overall number of uploaded user ratings.

Profits have been set in a 1–100 scale, and visiting times vary from 5 min (e.g., for some outdoor statues) to 2 h (e.g., for some not-miss museums or long scenic routes).

About half of the POIs are outdoors and always visitable (24 h time windows) while the remainder are associated with relatively wide, largely overlapped time windows (typically around 8 h). Around ¾ of the scenic routes are always visitable while the remainder are associated with shorter time windows (e.g., market areas or routes which are considered more enjoyable under daylight). Most of the selected hotels are situated around main attraction areas, while some hotels are situated in the city suburbs.

4.2 *Scenic Athens* architecture and application workflow

Scenic Athens adopts a coarse-grained service-oriented architecture (SOA) approach, wherein the business logic is composed of loosely coupled Web services, combined together to deliver personalized services to thin mobile clients. The mobile client application (https://www.youtube.com/watch?edit=vd&v=xwjGGz0Yp3o) has been developed on the Android 4.4 platform (Android API versions $\geq$3.0 are also supported) and is available from http://zarcrash.x10.mx/ScenicAthens.apk).

4.2.1 *Offline phase*

The pairwise travel (walking) times among all locations stored in the content database (including the end points of scenic routes) are computed using GraphServer.[6] POIs/scenic routes data and walking times are stored in memory structures on the server side to ensure high responsiveness to user queries. POIs/scenic routes metadata are also stored locally in the *Scenic Athens* mobile application; having received a recommended tour description (i.e., sequence of POIs/scenic routes IDs and travel times), these metadata are utilized to visualize the recommended tours and the details (metadata) of recommended POIs. This approach prevents unnecessary interactions with the server and significantly improves the application's performance, as perceived by the user.

4.2.2 *Online phase*

Aside from the customized tour planning, the *Scenic Athens* application offers to the user two additional services, both accessible from the first screen of the application (see Fig. 4a). The *Around Me* service (see Fig. 4b) allows the user to filter nearby hotels, restaurants or POIs by applying a—up to 6 km—distance threshold from her location. Additional filters may be applied based on the categorization of POIs and scenic routes (e.g., choose to only show museums and churches). The user is also provided an augmented reality view of nearby (up to 1.3 km) POIs (see Fig. 4c).

Prior to requesting a tour plan, the user feeds in the trip details principally the trip dates and the start/end locations/times of the daily tours (see Fig. 4d). The start/end locations may be selected among a predefined list of hotels or city landmarks; alternatively, it may be set by pointing an arbitrary location on a map or by obtaining a location fix via GPS (Fig. 4e). Optionally, the user may configure her profile by rating her preference on POI and/or scenic route categories (see Fig. 4f, g). In effect, those ratings are used to adjust default POI/routes profit values accordingly.[7]

Upon the submission of a user request, the tour planning Web service returns a JSON response comprising a description of the daily tourist tours. The recommended tours may be visualized using a list, map and hybrid itinerary views, in addition to the augmented reality view. The list view (see Fig. 4h) illustrates recommended POIs (green background) and scenic routes (purple background) as to their visiting order, while also showing their title, photograph, category, rating, estimated arrival/departure time and visiting duration (walking time in case of scenic route). A 'footsteps' icon placed on each list item may be tapped to yield walking directions from the previous tour element (see Fig. 4i). The map view (see Fig. 4j) presents the POI locations, using a different color code to denote the path followed along scenic routes. The hybrid itinerary view is a combination of the map view and a horizontal swipe image gallery (see Fig. 4k). The augmented reality view requires a GPS location fix and displays the locations of the POIs/routes included in the currently selected tour. In the case of scenic routes, a Google street view is also available, wherefrom the user can navigate along the recommended scenic route from its path start location (see Fig. 4l).

The user may retrieve further information for selected POIs or scenic routes along with a distance estimate (from the user's current location) and a compass icon pointing toward to that location (see Fig. 4m). A 'check-in' button may be tapped to indicate that the POI/route has actually been visited. The users are offered the option to remove (i.e., 'blacklist') any POI or scenic route from the proposed tours; removed POIs and scenic routes may be restored later on. Upon a POI removal (or check-in), the tour planning service is re-invoked, ensuring that this POI will not be recommended again. The users can also rate the POIs and scenic routes, contributing on their respective profit adjustment on future requests (see Fig. 4n). Finally, a

[6] http://graphserver.github.io/graphserver/.

[7] The adjustment of profit values of individual POIs/routes also takes into account the average rating score received by other mobile users for this POI/route. Ratings provided by other users are taken into account only when the number of ratings received for a particular POI/route is more than 10 so that statistical validity is guaranteed.

Table 4 Performance test results

Dataset (#POIs/#scenic routes)	# Daily tours	Overall time (ADSL, ms)	Overall time (3G, ms)	Execution time (ms)	Profit
50/10	1	337	484	181	801.98
	2	357	573	183	1270.99
	3	366	512	185	1655.63
	4	385	511	187	1988.62
100/20	1	377	511	218	1463.68
	2	422	622	243	2231.34
	3	434	568	245	2793.85
	4	446	598	245	3268.49
200/40	1	557	652	376	1539.5
	2	693	872	494	2508.26
	3	733	862	525	3192.43
	4	740	899	526	3773.15

weather forecast may be requested for a specific day of the trip (see Fig. 4o). It should be noted that *Scenic Athens* has been implemented following an iterative design approach. Several features, among the ones described above (e.g., the 'what POIs are around,' the 'check-in' and the weather forecast), have been developed along this process based on the suggestions of a restricted number of users (focus group).

4.3 Performance tests

This subsection documents results yield from performance tests conducted upon our scenic tour planning algorithm. The experimental testbed has been set-up as follows: the tour planning Web service has been hosted on an Intel(R) Core(TM) i7-4790 K CPU @4.00 GHz server; a client application has been deployed on a remote computer connected to the Internet either through an ADSL or through a 3G connection (the latter 'simulated' the execution of the client application on a smartphone device).

The performance results are presented in Table 4. The 'overall time' measurements account for the time spent from the moment that a query (i.e., tour request) is issued from the client until a JSON-formatted tour description is received back. Note that this does not include the time required to retrieve POI metadata (e.g., POI descriptions and photos) from the client's local storage and render the tour in a list/map representation. We also report the net execution time required to compute tour planning on the server side, thus also providing an indication of the associated network latency. We have conducted tests which involved three different datasets; the first dataset included 50 POIs and 10 scenic routes, the second 100 POIs and 20 scenic routes and the third 200 POIs and 40 scenic routes. For each dataset, we performed 100 separate invocations of the tour planning service, each for a different tour start/end location. The values presented in the table represent the average measurements over those 100 invocations. The number of daily tours requested varied from 1 to 4, while the time window for each daily tour has been set to 10:00–15:00. The table also presents the average profit associated with the derived tours (i.e., the sum of profit values of selected POIs and scenic routes). All the datasets-related files are accessible from: http://dgavalas.ct.aegean.gr/public/scenic_instances/.

The test results provide evidence of the real-time performance achieved by our tour planning service since the overall query service time remains below 1 s, even when considering the computation of four tours on large datasets (200/40 POIs/scenic routes), with clients using relatively slow (3G) network connections.

5 User evaluation

We have executed user evaluation trials, held in Athens (Greece) in October 2015, utilizing the user-centric evaluation framework of Pu et al. [17]. Participants belonged to two main evaluator groups:

- Locals (permanent city residents or students, familiarized with the main attractions and scenic routes of Athens).
- Tourists, largely unfamiliar with the tourist destination.

A few among the locals have been personal contacts of the authors, while the rest have been 'friends of friends.' Tourists have been recruited in the vicinity of major city attractions. The evaluators have been handed a flyer summarizing the functionality of the app and the objectives of the field trials along with a URL to download the app. They have been allowed to download the app after having registered contact information on an online form. The

Table 5 Evaluation questionnaire

	Locals	Tourists
Demographics, usage style		
	Questions asking identity (local/tourist), age, gender, device used, familiarity with analogous tools (mobile tourist guides)	
	Separate questions about style of usage [how many tour recommendations I received, how many places I checked-in, how faithfully I followed the recommended tour(s)]	
Usability		
Q1	The process of receiving a personalized tour was straightforward	
Q2	The result of my actions has always been consistent with my expectations	
Q3	Setting up my preferences and trip details was easy	
Q4	Overall, the application has performed (i.e., responded) reasonably fast	
Q5	Overall, the *Scenic Athens* application has been easy to use	
Recommendation value, functionality, perceived utility		
Q6	The recommended attractions and routes matched my interests	
Q7	(a) The recommended attractions and scenic routes have been the ones that I would recommend to a tourist visiting Athens	(b) The recommended attractions and scenic routes have been the ones that I would like to have been recommended
Q8	The incorporation of scenic routes into the tour (further to visitable sites) increased the tour's quality	
Q9		The sequencing of POIs and scenic routes along the tours was reasonable (it always took short time to get from an attraction to another)
Q10	It often occurred to reject (delete) recommended POIs	
Q11	Check the features you have mostly found attractive and useful, if any (map view of nearby attractions, list view of the tour, map view of the tour, augmented reality, street view, compass showing the direction toward an attraction)	
Q12	Check the features you have mostly found disappointing or less useful, if any (map view of nearby attractions, list view of the tour, map view of the tour, augmented reality, street view, compass showing the direction toward an attraction)	
Q13	Overall, an application like *Scenic Athens* would be useful to have when visiting any unfamiliar destination	
Q14	I would be willing to pay (a small amount of money) to have an app like this when visiting an unknown tourist destination	

individuals who downloaded the app have been contacted (through emails) a few days later and requested to answer anonymously an online questionnaire. A total of 13 locals (10 men, 3 women) and 6 tourists (4 men, 2 women) aged between 25 and 47 years responded to our request, tested the *Scenic Athens* application using a variety of mobile devices and provided feedback. Among the evaluators, only two have had previous experience with a city guide.

The questionnaire is shown in Table 5.[8]

5.1 Usage style

The participants received a sufficient number of tour recommendations (4.8 and 6.3, on average, for the locals and the tourists, respectively). This indicates that the tourists explored the customized tour planning feature more thoroughly. The average trip duration for the tourists has been 2.7 days, while their average number of check-ins in POIs has been 6.1. Last, the majority of the tourists argued that, to some extent, they actually followed the recommended tours although some found it impossible to stick to the indicated departure/arrival times ('You can't reasonably follow such tours 'faithfully'; this kind of app, you use it only to receive a rough schedule.').

5.2 Usability

The evaluation results relevant to usability are illustrated in Fig. 5. Most of the users, both locals and tourists, found the process of receiving a personalized tour application straightforward (Q1). A user who expressed neutral opinion justified his view by the lack of familiarity with similar applications.

Users (especially locals) appeared more neutral with regard to the predictability of their actions (Q2). A local user stated: 'A sufficient level of familiarization is required to understand what you can do with this application and to get to know the application capabilities to a full extent.'

Several participants found it relatively difficult to set-up their preferences and trip details, such as the start/end time of daily tours (Q3), mainly because they did not think of pressing their smartphone's settings soft button or they did that on the first screen. As for the application's performance (Q4), only 2 among the 19 evaluators have been unsatisfied (most probably due to their poor device

[8] A pdf version of the questionnaire is available from: http://zarcrash.x10.mx/ScenicEvaluationSurvey.pdf.

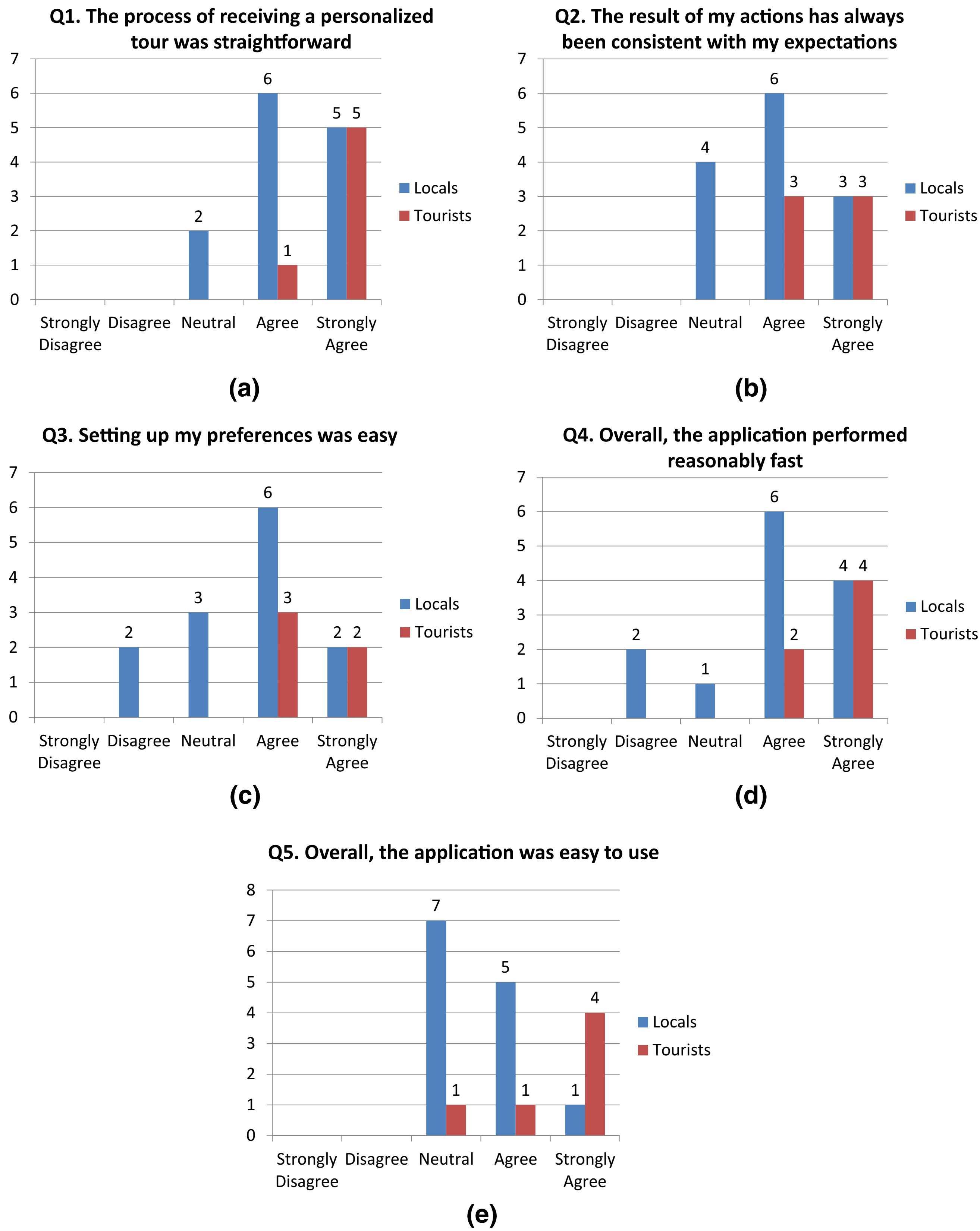

Fig. 5 Usability statistics

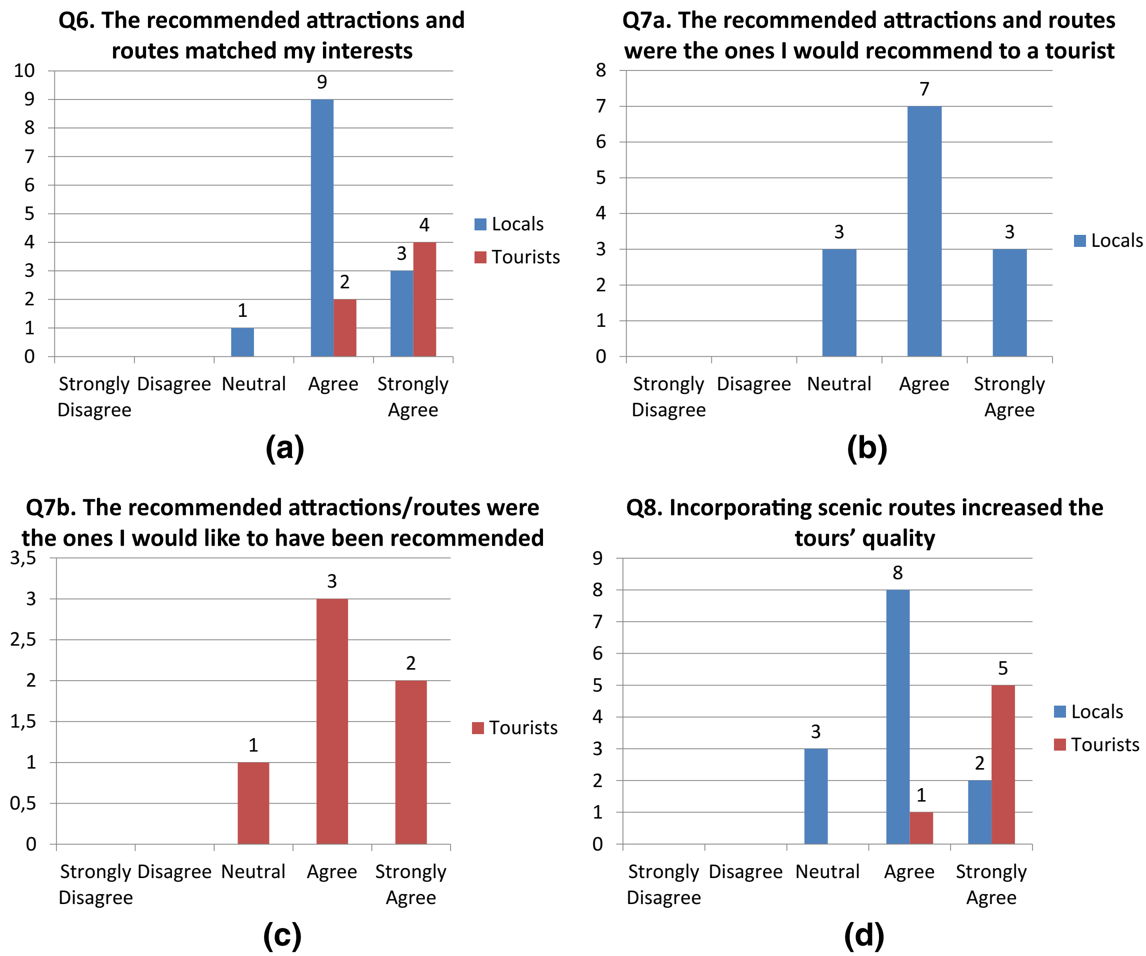

Fig. 6 Recommendation value statistics

resources or bad signal reception which increased network latency in the interaction with the server) stated that the application was not performing reasonably fast (responding to Q4). Overall, respondents felt relatively comfortable in using the application and consuming the tourist content (Q5), although some reported that they found difficult to grasp the function of the augmented reality view, navigate among the tours recommended for different days or modify the trip details of a particular daily tour. In conclusion, it appears that several of the reported usability problems could be resolved by increasing the 'visibility' of some otherwise missed features or through creating semitransparent 'first time' instruction pages.

5.3 Recommendation value, functionality, perceived utility

The statistics relevant with the evaluation of recommended tours' value are illustrated in Fig. 6. To a greater or lesser degree, the users argued that the recommended attractions and routes matched their interests (Q6). Most locals would recommend the POIs and scenic routes to a tourist (Q7a) ('The application recommended a lot of interesting sites, some of which, even myself as a local, didn't know they exist'); nevertheless, a couple of respondents pointed out that they would also recommend some additional walking routes of the 'unexplored' Athens. Accordingly, most of the tourist users confirmed that the recommendations have been the ones they would like to receive when visiting Athens (Q7b), although a respondent argued that the content could be expanded so as to include more POI categories (such as fun parks or family-friendly sites). The unique feature of *Scenic Athens*, i.e., the inclusion of scenic routes in the recommended tours, has been appreciated as rather important by most users (a tourist commented 'I only requested the app to suggest walking tours. Haven't visited any site requiring entrance, really!'), which increased the value and meaningfulness of the tours' quality (Q8). Nevertheless, some users possibly felt uncertain about that since they have not had experience with alike applications solely offering POI recommendations.

The statistics relevant with the functionality and perceived utility of *Scenic Athens* are illustrated in Fig. 7. It

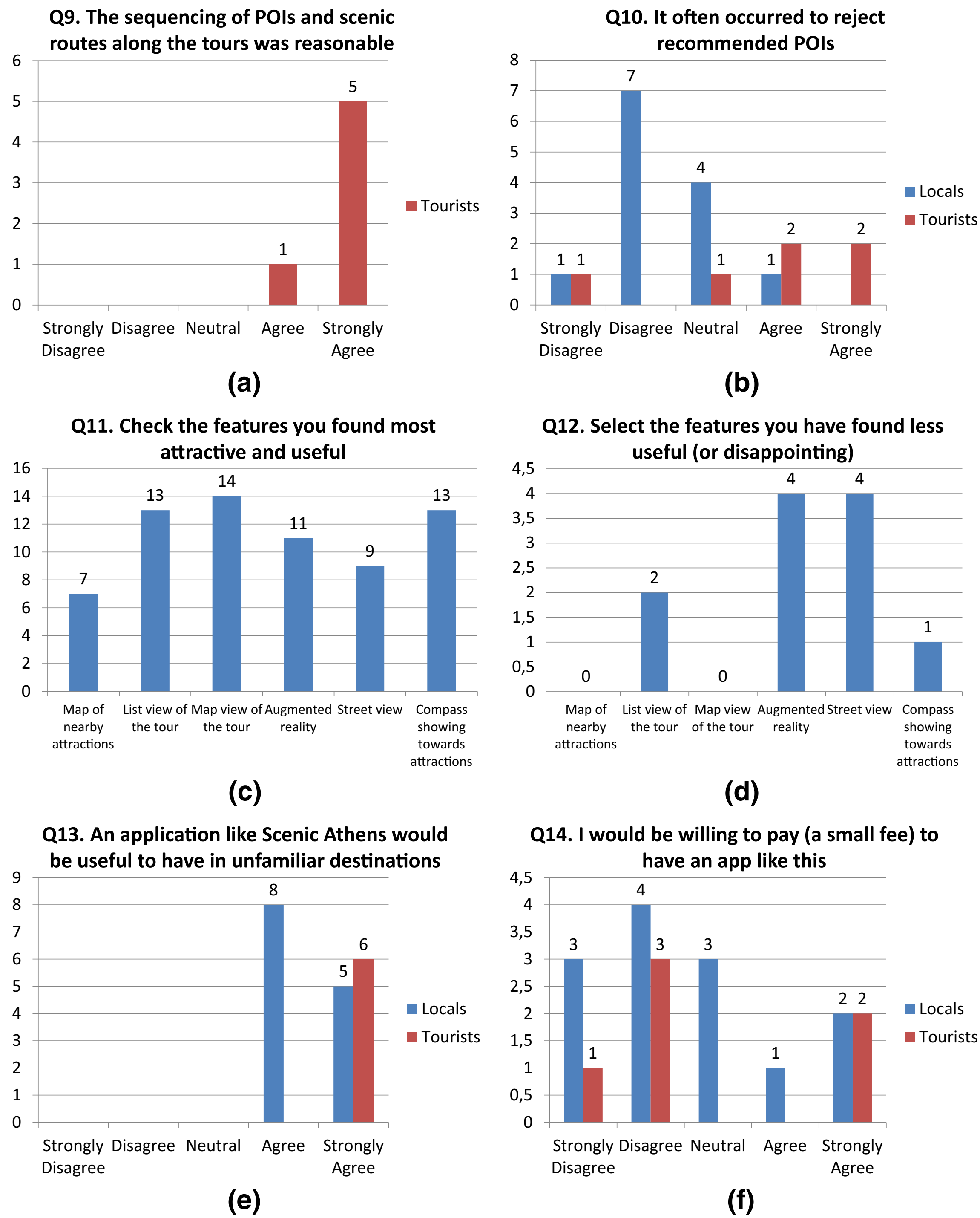

Fig. 7 Functionality and perceived utility statistics

appears that most tourists made the most out of their time, as the sequencing of POI visits and scenic routes ensured short walking distances among them (Q9). The majority of the users (especially the locals) did not often reject (delete) recommended POIs (Q10) (one respondent stated that 'I did not even know I could do that!' while another suggested that 'It'd be nice to be able to shuffle the sites in a way that makes more sense to you'). This agrees with the responses collected with respect to the usage of the customized tour planning feature, confirming that the tourist evaluators actually used the application in realistic conditions. It is also concluded that, even if the recommendations largely match the users' interests, the option to remove POIs or scenic routes is quite important to have so as to allow further personalization of recommended tours.

As for the usefulness and attractiveness of the application features (Q11 and Q12), users received positively the main means for displaying the recommended tours (i.e., the list and the map view). The use of the rotating compass has been appreciated since it contributed to the sense of orientation. The augmented reality view received several enthusiastic comments (especially from users with no prior exposure to this technology) but also criticism ('I felt frustrated chasing after the floating markers'). The street view has also been criticized for taking long to load and offering no real insights on the actual scenic value of routes.

All the evaluators valued the utility of alike applications in unfamiliar destinations (Q13). However, only a few would pay a small fee for a similar application ('This kind of applications should be available for free, as they support tourism,' 'There are lots of tourist guides available in Google Play for almost every metropolis on earth'). Nevertheless, we argue that *Scenic Athens* represents a promising business case as evidenced by the viability and outspread of similar commercial projects,[9] which offer only a fraction of the functionality presented herein. Alternatively, alike applications could also be seen as an added-value service to the 'tourist package' as a whole (potentially raising funding from local authorities, tourist businesses or cultural institutes).

The evaluators also contributed several suggestions to further improve the application. Among others, *Scenic Athens* could be extended so as to:

- be able to function offline to save roaming charges;
- consider public transit transfers among POIs;
- recommend routes accessible by people with motoring disabilities;
- take into account queuing delays on the entrances of popular POIs;
- support TripAdvisor-like features for reviewing/commenting on POIs;
- allow saving a trip diary (POIs visited, impressions, user photos, etc.).

6 Conclusion and future research

In this article, we presented a novel tourist tour planning approach which opposes the perception of urban tourism as the 'disembodied visual experience of navigating between sights' [8] and supports the discovery of natural, cultural, architectural, historical and market assets of tourist destinations.

To materialize this approach, we employed an efficient metaheuristic to tackle the MTOPTW problem so as to allow incorporating scenic routes into recommended tours. MTOPTW solutions are appropriate to highlight the scenic value as well as the cultural and architectural elements of urban districts. Thus, we argue that MTOPTW captures realistic requirements of tourist destination visitors and, thus, compares favorably to any other relevant approach found today in the TTDP-related literature. This novel algorithmic approach is used in *Scenic Athens*, a context-aware mobile city guide for Athens (Greece) which provides personalized tour planning services and supports the experiential exploration of tourist destinations. We have conducted several performance tests which demonstrated that the algorithmic engine of *Scenic Athens* definitely meets the real-time requirements of tourist tour planning applications. A user evaluation study validated the quality of tour recommendations as well as the usability and perceived utility of *Scenic Athens*.

In the future, we plan to implement versions of *Scenic Athens* incorporating more elaborate formulations of TTDP. For instance, we could consider (a) multiple user-defined constraints (further to the time budget allowed per daily tour), such as a limited money budget for POI entrance fees; (b) 'max-n type' for each day and for the whole trip (e.g., a maximum number of museums to visit on the first day or maximum walking time per day).

Acknowledgments This work has been supported by the EU FP7/2007–2013 Programme under Grant Agreements No. 288094 (eCOMPASS—'eco-Friendly Urban Multi-Modal Route Planning Services for Mobile Users') and No. 621133 (HoPE—'Holistic Personal public Eco-mobility').

References

1. Brilhante IR, Macedo JA, Nardini FM, Perego R, Renso C (2015) On planning sightseeing tours with TripBuilder. Inf Process Manage 51(2):1–15
2. Cohen R, Katzir L (2008) The generalized maximum coverage problem. Inf Process Lett 108(1):15–22
3. Gavalas D, Konstantopoulos C, Mastakas K, Pantziou G (2014) Mobile recommender systems in tourism. J Netw Comput Appl 39:319–333
4. Gavalas D, Konstantopoulos C, Mastakas K, Pantziou G (2014) A survey on algorithmic approaches for solving tourist trip design problems. J Heuristics 20(3):291–328
5. Gavalas D, Kasapakis V, Konstantopoulos C, Pantziou G, Vathis N, Zaroliagis C (2015) The eCOMPASS multimodal tourist tour planner. Expert Syst Appl 42(21):7303–7316
6. Gavalas D, Kasapakis V, Konstantopoulos C, Mastakas K, Pantziou G, Vathis N, Zaroliagis C (2016) Scenic Athens: a personalized scenic route planner for tourists. In: Proceedings of the 21st IEEE symposium on computers and communications, pp 1151–1156
7. Gavalas D, Konstantopoulos C, Mastakas K, Pantziou G, Vathis N (2016) Efficient metaheuristics for the mixed team orienteering problem with time windows. Algorithms MDPI 9(1):6

[9] http://www.mtrip.com/, http://www.citytripplanner.com/.

8. Haldrup M (2004) Laid-back mobilities: second-home holidays in time and space. Tour Geogr 6(4):434–454
9. Hu Q, Lim A (2014) An iterative three-component heuristic for the team orienteering problem with time windows. Eur J Oper Res 232(2):276–286
10. Kenteris M, Gavalas D, Economou D (2011) Electronic mobile guides: a survey. Pers Ubiquit Comput 15(1):97–111
11. Kurata Y, Hara T (2014) CT-Planner4: toward a more user-friendly interactive day-tour planner. In: Proceedings of the 2014 international conference on information and communication technologies in tourism, pp 73–86
12. Labadi N, Melechovský J, Wolfler Calvo R (2011) Hybridized evolutionary local search algorithm for the team orienteering problem with time windows. J Heuristics 17:729–753
13. Laporte G, Martello S (1990) The selective travelling salesman problem. Discret Appl Math 26(2):193–207
14. Larsen J (2001) Tourism mobilities and the travel glance: experiences of being on the move. Scand J Hosp Tour 1(2):80–98
15. Lourenço HR, Martin OC, Stützle T (2003) Iterated local search. Handbook of metaheuristics. Springer, Berlin, pp 320–353
16. McKean J, Johnson D, Walsh R (1995) Valuing time in travel cost demand analysis: an empirical investigation. Land Econ 71:96–105
17. Pu P, Chen L, Hu R (2011) A user-centric evaluation framework for recommender systems. In: Proceedings of the 5th ACM conference on recommender systems, pp 157–164
18. Ricci F (2010) Mobile recommender systems. Inf Technol Tour 12(3):205–231
19. Shoval N, Isaacson M (2007) Tracking tourists in the digital age. Ann Tour Res 34(1):141–159
20. Souffriau W, Vansteenwegen P (2010). Tourist trip planning functionalities: state of the art and future. In: Proceedings of the 10th international conference on current trends in web engineering, pp 474–485
21. Souffriau W, Vansteenwegen P, Berghe GV, Van Oudheusden D (2011) The planning of cycle trips in the province of East Flanders. Omega 39(2):209–213
22. Vansteenwegen P, Souffriau W, Vanden Berghe G, Van Oudheusden DV (2009) Iterated local search for the team orienteering problem with time windows. Comput Oper Res 36:3281–3290
23. Vansteenwegen P, Souffriau W, Van Oudheusden D (2011) The orienteering problem: a survey. Eur J Oper Res 209(1):1–10
24. Vansteenwegen P, Souffriau W, Vanden Berghe G, Van Oudheusden D (2011) The city trip planner: an expert system for tourists. Expert Syst Appl 38(6):6540–6546