Engineering Applications of Artificial Intelligence 72 (2018) 162–169

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

PRoA: An intelligent multi-criteria Personalized Route Assistant

Marina Torres *, David A. Pelta, José L. Verdegay

ETS de Ingenierías Informática y de Telecomunicación, Departamento de Ciencias de la Computación e I.A., Universidad de Granada, 18014, Spain

ARTICLE INFO

Keywords:
Route design
Walking
Personalization
Multi-criteria evaluation

ABSTRACT

Personalization of pedestrian routes becomes a necessity due to the wide variety of user profiles that may differ on preferences or requirements to choose a route. Several software applications offer routes usually based on single criterion like distance or time; however, these criteria do not often fit the pedestrian needs.

Here, we will first focus on the Personalized Routes Problem and then we will approach the specific case of designing accessible and green pedestrian routes.

The proposal is implemented as a freely available Android application (named as PRoA, by intelligent multi-criteria Personalized Route Assistant), which automatically obtains geographical data and information for the decision criteria from open datasets.

The proposal is evaluated using real cases at the city of Granada, Spain.

1. Introduction

Walking is transportation, green and healthy. In the *National Travel Survey: England 2014* (Tranter et al., 2015) one can read that the transportation modes accounting for most trips in 2014 were by car, either as a driver or a passenger (64%), whereas walking accounted for a 22%.

Indeed, walking is the most frequent transportation mode used for very short distance trips: 76% of all trips under one mile are walks. In terms of averages per person, there were 200 walking trips registered in 2014 for a total of 180 miles and 18 min per walking trip.

The Institute for Transportation and Development Policy (ITDP),[1] from New York, states that:

> "For decades, traditional transport planning has focused on improving conditions for private automobiles at the expense of safe sidewalks and bike facilities. Yet, the majority of the world's people rely on cycling, walking, and other forms of human-powered transport [...]. Increasing the use of bicycles and the ease of walking is one of the most affordable and practical ways to reduce CO2 emissions, while boosting access to economic opportunity for the poor".

These environmental benefits, including moderate-intensity physical activity like walking or cycling, had proved to provide substantial health benefits (Pate, 1995).

As the literature indicates, not only the environmental variables are related to physical activity (Saelens et al., 2003) but also the adequate facilities for walking or the accessibility of places to walk (Owen et al., 2004). Craig et al. (2002) states that "Walking to work was significantly related to the environment score", where score is based on 18 neighborhood characteristics (e.g., existence of accessible walking routes, such as sidewalks and paths, and available facilities like parks). In this way, a "good environment" or an "appropriate route characteristics" are significantly related to the decision of walking for example to work or just for leisure. However, those concepts are user-dependent and may vary according to his/her preferences.

Walking is crucial for senior citizens who may found many impediments on their routes such as stairs or steep slopes. Consequently, this kind of people need to conscientiously choose their route according to those constraining characteristics (Borst et al., 2009). Similarly, people with some mobility reduction, pregnant women, or people that want to walk with small kids may be also affected by the same constraints.

Given the proven relationship between the characteristics of a route and the user's decision to walk, the proper design of a route based on user's preferences is essential to promote walking activities. Some existing approaches are outlined next.

For example, Balstrøm (2002) proposed a method to find the faster walking routes in open field and the criteria were defined according to the surface's friction. Walking routes aimed for elderly people were determined by Borst et al. (2009) with a previous evaluation of the links on up to twenty-three physical characteristics (e.g., slopes, stair, shops or zebra crossings). Similarly, Hochmair (2008) uses criteria such as waiting time, turns, or traffic lights to find a multi-modal route

* Corresponding author.
E-mail addresses: marinatorres@ugr.es (M. Torres), dpelta@ugr.es (D.A. Pelta), verdegay@ugr.es (J.L. Verdegay).

[1] https://www.itdp.org/what-we-do/cycling-and-walking/.

https://doi.org/10.1016/j.engappai.2018.03.016
Received 3 October 2016; Received in revised form 29 January 2018; Accepted 22 March 2018

that includes walking and public transportation. The method presented by Quercia et al. (2014) and López-Ornelas et al. (2014) focused on the walkability concept or pleasant routes.

Although these approaches are sound they generally need feedback from users to determine the street's walkability. This is the case of Walkonomics (2011), which despite using some geographical data from different open data sources, is just currently working on four cities (Toronto, San Francisco, London and New York). Another case of dependence on feedback from a user is that proposed by Vadeo (Naranjo and Bayo, 2008). The system is deployed in the city of Valencia, Spain, where the users have indicated obstacles or impediments for handicapped people. Again, the former project cannot be considered enough to achieve a trustful system because it also depends on creating a new community, as well as on the cooperation of its members.

A different approach (that does not requires a new community of users) is Walkability Explorer by Cecchini et al. (2014). The system uses data from OpenStreetMap to determine a so called walkability score. Previously, Leslie et al. (2007) used data from GIS to objectively measure features related to walkability.

Our aim here is threefold: firstly we present the Personalized Routes Problem; secondly we model the case for Personalized Pedestrian Routes and thirdly, we present PRoA: an intelligent personalized route assistant, an Android application available on Google Play (Torres, 2016). PRoA relies on open data and does not need to create a new community to work properly. In other words, PRoA is completely functional from the very beginning.

The paper is structured as follows. First, in Section 2 we provide a general approach to the Personalized Route Problem. Then, in Section 3, we propose a specific case where the chosen criteria are distance, upward slope, downward slope, stairs and green or pedestrian zones leading to the Personalized Pedestrian Route Problem. The routing algorithm is based on A^* and a basic strategy for generating alternative routes is proposed. Section 4 describes PRoA: an Android based application implementing all the features described above. The application's results and benefits can be seen in Section 5, where we show practical examples in the city of Granada, Spain. Finally, Section 6 is devoted to conclusions.

2. The personalized routes problem

The Personalized Routes Problem (PRP) can be stated as follows: given origin ($start$) and destination ($goal$) points, and a set of user preferences and constraints, find the best route from $start$ to $goal$ according to the user's requirements. To solve the problem, three subproblems should be considered.

(a) Map construction

The first subproblem is the construction of the streets' map, which requires gathering information from maps repositories (either public or proprietary). At the end of the process, we will end up with an undirected graph $G = \{N, E\}$ where each *edge* $e \in E$ is delimited by two elements $start_e$ and end_e from N, the set of *nodes*. The edges in E represent street segments from the map. Both edges and nodes have a set of features F that could be numerical values, e.g., elevation, latitude, longitude or length, or nominal characteristics like the edge's street type, e.g., path, ban or pedestrian.

(b) Map evaluation

On the second subproblem, the graph is processed: using the set of features F, a single value (a cost) is calculated for each edge. This step requires to make both an evaluation and an aggregation process based on a set of elements $\{G, F, \Sigma_{OPT}, \Sigma_{CON}, g, W, z\}$. The set $\Sigma_{OPT} \neq \emptyset$ contains the criteria to optimize and it is coupled with the set of objectives or goals g. Each objective in g has an associated weight from W. The weights W represent the importance of reaching its correspondent objective of *minimization* or *maximization* in g. Every criterion in Σ_{OPT} has an associated weight in W. The weights are constrained as follows:

$$\sum_{i=0}^{|\Sigma_{OPT}|} w_i = 1 \qquad w_i \in [0,1]. \tag{1}$$

Finally, the set Σ_{CON} contains criteria for which certain maximum/minimum values are defined by means of the constraints in z.

The scheme of the evaluation and aggregation processes is shown in Fig. 1. There are two phases: (1) the evaluation phase, where we use the set of features F, the criteria Σ_{OPT} and the objectives g to obtain a set of evaluations EV; and (2) the aggregation phase, where we use the evaluations EV, the weights W, Σ_{CON} and z to obtain the aggregated cost C_e for each edge e.

(c) Routes calculation

Finally, having a new graph where every edge e has a cost assigned, a routing algorithm should be applied to obtain the optimal and (possibly) some alternatives routes.

3. The personalized pedestrian routes problem

When the elements of the PRP are defined, a specific problem is obtained. Here we will focus in the personalized pedestrian routes problem, proposing the specific definitions we will use.

(a) Map construction

We depart from a graph G and a set of features F that are obtained from public repositories. Then we define two sets of criteria:

1. Optimization criteria: $\Sigma_{OPT} = \{distance, upward\ slope, downward\ slope, green\ zones\}$, with the following goals $g = \{minimize, minimize, minimize, maximize\}$.
2. Constrained criteria: $\Sigma_{CON} = \{upward\ slope, downward\ slope, stairs\}$ with the constraints $z = \{limit, limit, avoid\}$. These constraints determine if a specific edge should be considered or not in the calculations, either because it exceeds some limit (for example with the *slopes*) or contains an undesirable feature (like *stairs*).

(b) Map evaluation

For every edge $e \in E$, our aim is to obtain a set of evaluations $EV = \{D_e, S_e, G_e\}$ for the criteria in Σ_{OPT}. We depart from the set of features $F = \{distance_e, elevation_{start_e}, elevation_{end_e}, type_e\}$, that is, the length of the edge, the elevation of the start and end nodes of the edge and the edge's type. The $type_e$ value can represent a track (unpaved roads like forest tracks), a path (trails open to all non-motorized vehicles, like hiking trails or bike trails), pedestrian streets (car-free zones), stairs (indicates a street with steps) or none of the above.

The importance of reaching the corresponding goals g associated with the criteria is defined by the weight vector $W = \{w_d, w_{up}, w_{down}, w_g\}$. The calculation of EV requires some intermediate values, namely $slope_e$, d_{max}, $greenCost_e$, which are described below.

The $slope_e$ of an edge e is calculated as

$$slope_e = \begin{cases} \nabla elevation_e / distance_e & \text{if } distance_e \geq 1 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $\nabla elevation_e = |elevation_{start_e} - elevation_{end_e}|$ is the variation of the elevations between the start and the end nodes of the edge e (both in meters). As it may happen that $distance_e < 1$, in such a case we fix the $slope_e = 0$ to avoid problems in the calculations.

The $greenCost_e$ score of an edge e is calculated according to the type of street $type_e$.

$$greenCost_e = \begin{cases} 0.9 & \text{if } type_e \in \{\text{track, path, stairs, pedestrian}\} \\ 0 & \text{if none of the above.} \end{cases} \tag{3}$$

The maximum distance d_{max} is obtained and later used to normalize the distances $distance_e$.

$$d_{max} = max\{distance_e \quad \forall e\}. \tag{4}$$

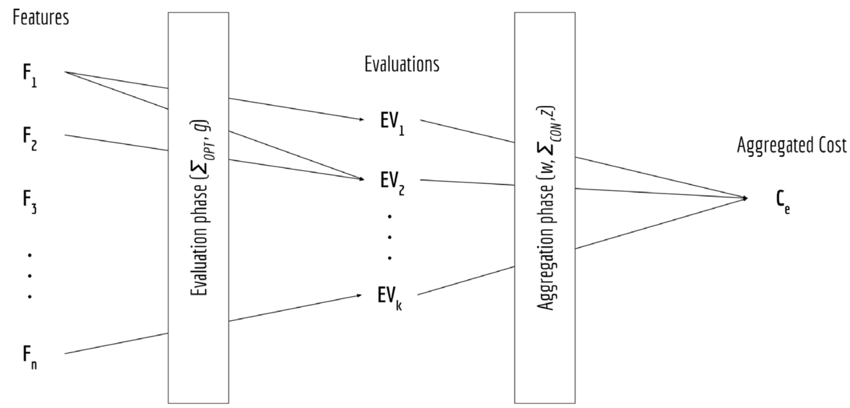


Fig. 1. Personalized routes problem: evaluation and aggregation stages.

Now, for each edge e the set $EV = \{D_e, S_e, G_e\}$ is calculated as:

$$\begin{aligned} D_e &= distance_e/d_{max} \\ S_e &= (slope_e)^2\, distance_e \\ G_e &= (1 - greenCost_e)D_e. \end{aligned} \tag{5}$$

As $slope_e \in [0,1]$, S_e does not need any normalization and will be referred as *upward slope* or *downward slope* criteria depending on the direction the edge is visited.

The values in EV are combined in the aggregation phase using the weights W, leading to an aggregated cost C_e for each edge e. The cost of an edge $C_e = \{C_e^{up}, C_e^{down}\}, 0 \le C_e^{up}, C_e^{down} \le 1$ is calculated as:

$$\begin{aligned} C_e^{up} &= w_d\ D_e + w_{up}\ S_e + w_g\ G_e \qquad \text{for upward direction} \\ C_e^{down} &= w_d\ D_e + w_{down}\ S_e + w_g\ G_e \qquad \text{for downward direction.} \end{aligned}$$

As an edge can be traversed upward or downward, the weight applied to S_e is different for both cases. That is why two values C_e^{up} and C_e^{down} should be considered. According to the *constraints* in $z = \{limit, limit, avoid\}$ applied to the criteria Σ_{CON}, the $C_e = \{C_e^{up}, C_e^{down}\}$ value will be:

1. if the edge has stairs, then $C_e = \{\infty, \infty\}$.
2. if the slope $slope_e$ is greater than the maximum slope allowed for the upward and/or downward direction, then $C_e^{up} = \{\infty\}$ and/or $C_e^{down} = \{\infty\}$.

(c) Routes calculation

The routes calculation is based on the A^* algorithm which is shown in Algorithm 1. The algorithm requires as input the origin (*start*) and destination (*goal*) nodes or locations and two functions (shown underlined) $cost(start_e, end_e)$ and $h(node)$.

The former function determines the cost C_e of the edge connecting the nodes $(start_e, end_e)$. This could be either C_e^{up} or C_e^{down} depending if the slope from $start_e$ to end_e is upwards or downwards.

For the A^* to be optimal, an admissible heuristic estimation is needed. The function $h(node)$ shown in Algorithm 2, calculates such estimation from $node$ to $goal$. The estimation depends on the *distance*, *downward slope* and *upward slope* criteria, while currently, there is no estimation value for the criterion *green zone*.

A single run of the A^* algorithm finds the optimal solution r^* according to the user preferences and constraints. In order to obtain alternatives routes, we use an approach that consist on using the same A^* algorithm but previously banning a set of nodes.

In the second run of the algorithm, and given the optimal route r^*, we initialize the $ClosedSet$ with those nodes from r^* belonging to the second third of the route. As a result, we will obtain a new route r'. If we want another route r'', the $ClosedSet$ will be initialized with the nodes in the second third part of the routes from r^* and r'. In this way, A^* does not expand or evaluate those *banned* nodes.

Algorithm 1 A* Algorithm Pseudocode.

```
function A*(start, goal)
    OpenSet ← {start}
    ClosedSet ← {}
    gCost[] initialized with default value of infinity
    gCost[start] ← 0
    hCost[start] ← gCost[start] + h(start, goal)
    while OpenSet is not empty do
        current ← node in OpenSet with lower hCost value
        if current = goal then
            return path(current)
        end if
        for each neighbor of current do
            if neighbor is not in ClosedSet then
                newGCost ← gCost[current] + cost(current, neighbor)
                if neighbor not in OpenSet then
                    OpenSet ← OpenSet ∪ {neighbor}
                end if
                if newGCost < gCost[neighbor] then
                    gCost[neighbor] ← newGCost
                    hCost[neighbor] ← gCost[neighbor] + h(neighbor)
                    cameFrom[neighbor] ← current
                end if
            end if
        end for
        OpenSet ← OpenSet\{current}
        ClosedSet ← ClosedSet ∪ {current}
    end while
    return no path found
end function
```

Algorithm 2 Cost estimation to go from a $node$ to the $goal$ node.

```
function h(node, goal)
    if node = goal then
        return 0
    end if
    dist ← euclideanDistance(node, goal)
    hDistance ← dist/d_max
    slope ← (elevation_goal − elevation_node)/dist
    if dist < 1 then
        hSlope ← 0
    else
        hSlope ← slope² dist                ▷ hSlope < 0 if downward slope
    end if
    h ← hDistance w_d + max{hSlope, 0} w_up − min{hSlope, 0} w_down
    return h
end function
```

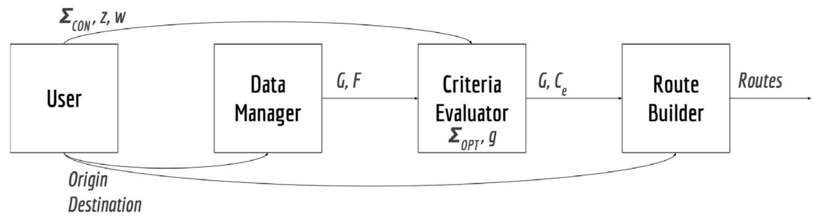


Fig. 2. Workflow diagram for a PRoA session.

This approach can be repeated as many times as alternatives routes are desired. However, it should be clear that it is not always possible to obtain a different solution. For example, if we banned two nodes connecting a street which is the only way to connect two separated areas, then the problem becomes unfeasible. Banning of the nodes in the second third part of the route is arbitrary but produces good results in practice.

The generation of alternative routes is far from trivial as stated in Bader et al. (2011). It should be noted that after exploring other options for generating routes, we kept this very basic approach because of two main reasons: firstly, it does not require many more additional computational resources (time, memory) which is an important issue because we are dealing with a mobile phone-based application; and secondly it does not require the user to define additional information that he/she may not know (like selecting a zone to avoid when he/she is trying to obtain a route in an unknown city).

4. PRoA: the application

PRoA is an Android application that implements the map construction, map evaluation and routes construction stages described in the previous section.

It is based on Android 5.0 APIs and is compatible with older versions (the minimum compatible version is 4.0.3). Some features are requested from third party services: OpenStreetMap Server Side Scripting, mapQuest Open Elevation Service, Google APIs (Maps Android, Directions, Elevation, Places) and AndroidPlot. PRoA also uses SQLite as a database manager, allowing the user to run the application offline once the maps have been downloaded.

4.1. PRoA system

PRoA has three different components: the Data Manager, the Criteria Evaluator and the Route Builder, each one associated with every stage of the personalized route problem.

A PRoA session leads to the workflow shown in Fig. 2. First, the user specifies the route plan (origin and destination locations). Then, he/she should indicate the importance of achieving each criterion's objective and the constraints to define W, Σ_{CON} and z. Then, the Data Manager, using the origin and destination locations, constructs the street graph G needed and the set of features F. This component obtains the data from the local database or, if it is not complete or unavailable, it requests the absent information using Web services. Then, the needed information is locally stored (in the device). Once the street graph G is completed, the Criteria Evaluator calculates the aggregated costs $C_e\ \forall e \in E$. After that, the Route Builder finds the optimal route (from origin to destination) and up to two alternatives, with the method discussed in Section 3.

4.2. The data

The Data Manager component obtains data from OSM (OpenStreetMap, 2004), a project that creates and distributes open and free geographical data. The OSM dataset has been compared to similar datasets and results have proved its good data quality (Haklay, 2010). Also, Zielstra and Hochmair (2012) concluded that routes planned with OSM data "*resulted in shorter shortest paths for pedestrians than commercial datasets*" in a study of the pedestrian routes based on OSM data on German and U.S. cities.

The data requested to OSM is:

- *Nodes*: the node is a point defined by its latitude, longitude and identification number.
- *Streets*: streets in OSM are an ordered list of nodes. Note that a node from a street could be an intersection of streets or a middle point of the street that connects two segments with different orientation.
- *Street type*: if it is pedestrian, road, path, track, steps, etc.

The elevation data is not provided by OSM, instead it is requested to other services based on OSM datasets that also include the elevation map or Google Maps Elevation API. Based on the previously described information, the Data Manager component calculates other important attributes:

- *Edges*: an edge in the graph is a segment of a street in OSM, that is, a connection between two nodes. A node in OSM is not always a street intersection so an edge of the graph can be connected to one edge (same street continues) or more edges (intersection).
- *Distance*: the length (in meters) of an edge is calculated as the distance between its connecting nodes.
- *Elevation variation*: the elevation variation (in meters) of each edge is calculated as the difference on the elevation data of its two nodes.

The system is designed with a local database (DB) allowing an offline usage. The DB is updated each time the user first request a route on a specified area of the map. For this purpose, the map's data is split into tiles. A node from the graph belongs to one and only one tile making easier to index the database. This method ensures data consistency because the tile is always fully requested and stored on the database so no data is missing if the tile is already stored.

The tiles requested to OSM are all the tiles that contain information inside the Bounding Box (an OSM concept described as the area defined by two longitudes and two latitudes) formed by the origin and destination points and all tiles adjacent to this Bounding Box. This approach limits the data available to build the route at Route Builder component, and because of this limitation the tiles needs to be of a considerable size. Fig. 3 shows an example of the Bounding Box concept between the origin and destination points, the required tiles and the graph used for a specified plan route. After some empirical testing, we defined size of the tiles is 0.01 degrees variation on latitude and 0.01 degrees variation on longitude.

4.3. PRoA screenshots

The interface has been consciously designed trying to fit all information on a smart-phone screen as well as on a tablet screen and making it easy to interact with the application. The functionality is divided into three tabs that correspond to the plan input data, the importance criteria specification and the routes visualization.

Screen captures are shown in Fig. 4. In the first tab (a), the user indicates the origin and destination points placing markers on a map

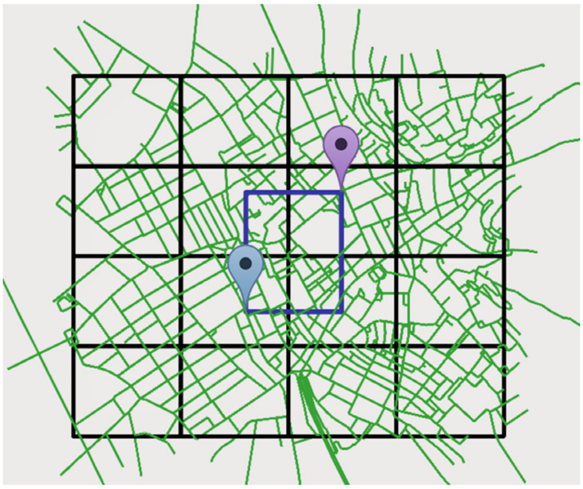


Fig. 3. Four tiles are needed to cover the bounding box area defined by the two markers. The 12 adjacent tiles are also requested to construct the shown street graph.

or writing the address on a search field that offers "auto-complete" functionality and suggestions.

Then, the user can select one of the three predefined profiles available (see Fig. 4(b)) or define a personalized one. The predefined profiles are "avoid slopes", "reduce distance" and "pass through green zones". The weights values for these profiles are summarized in Table 1.

Alternatively, the user may define a personalized profile setting the criteria importance as shown in Fig. 4(c). The user can further personalize his/her interests using the additional settings shown in

Table 1
Predefined weights profiles.

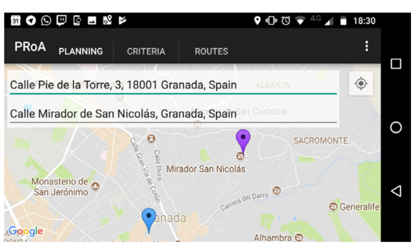
Profile	w_d	w_{up}	w_{down}	w_g
Avoid slopes	0.1	0.45	0.45	0.0
Reduce distance	1.0	0.0	0.0	0.0
Pass through green zones	0.3	0.0	0.0	0.7

Fig. 4(d), where slope limitations and enabling/disabling stairs in the routes can be defined.

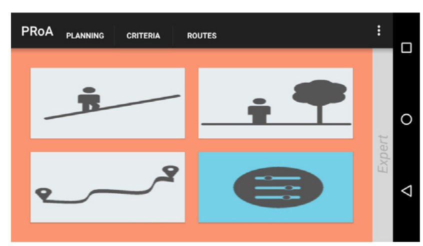
Fig. 4(e) shows the alternatives provided by PRoA for the criteria importance distribution shown in Fig. 4(c). The optimal route and alternative ones are depicted on the map and identified by colors. The user can select or unselect each route in order to show or hide its path and elevation data on the provided line chart like the one shown in Fig. 4(f). When an alternative is selected, the user can request the directions to see the list of instructions on how to follow the route. The routes can be saved and loaded later. This feature allows an easy comparison between the routes obtained by different criteria importance distribution.

5. Analysis and results in Granada, Spain

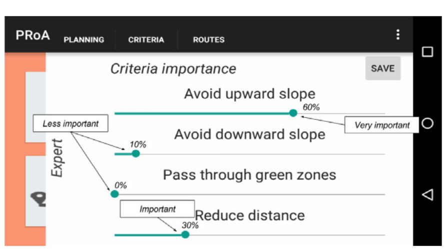
The assessment of PRoA is performed through a set of real world examples developed in Granada, Spain. Three analysis are presented. In the first one we compare the routes provided by PRoA with those provided by "de-facto" routing application Google Maps. Then, in the second part, we show a number of examples with slopes limitation, green zones and stairs avoidance. In this case, comparisons against other tools are not possible as such features (as long as we know) are not considered by any other application. Finally, in the third part, we show a brief performance analysis.



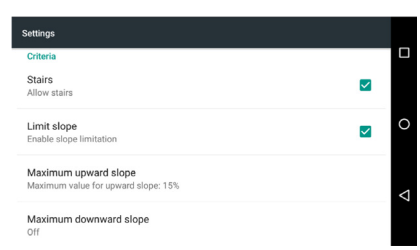
(a) Route definition.



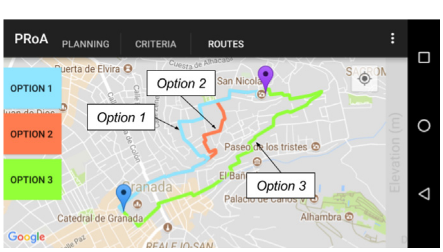
(b) Predefined profiles.



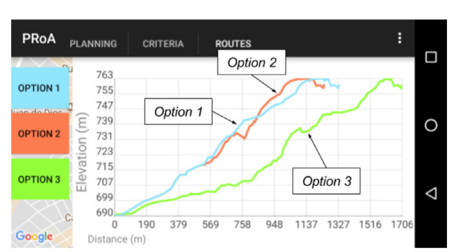
(c) User profiles.



(d) Additional settings.



(e) Results.



(f) Elevation profile.

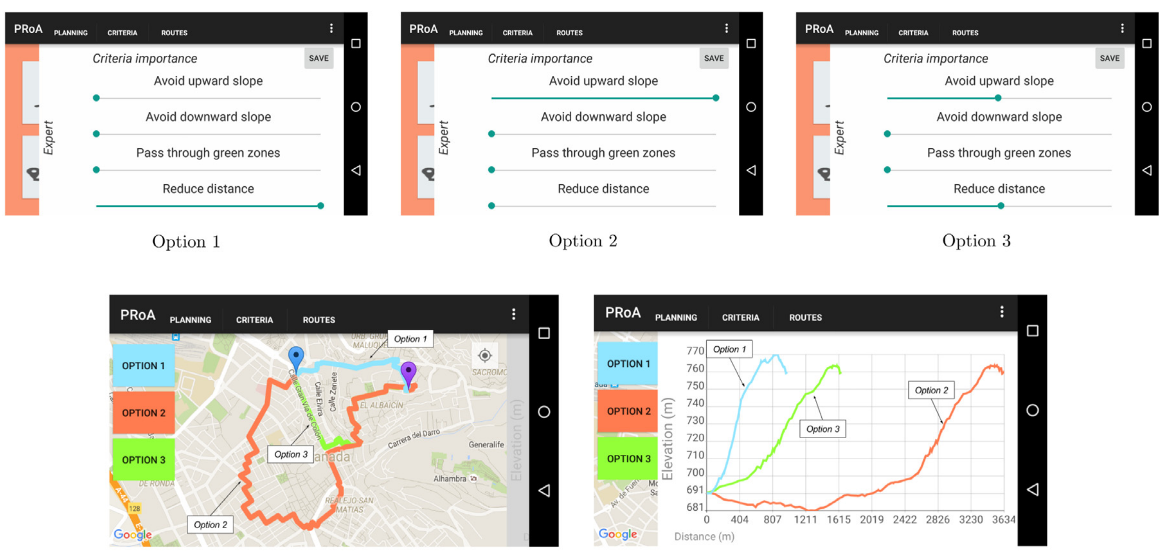
Fig. 4. Main screens of the application. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2
Illustrative examples of shorter distances routes found by PRoA against Google Maps.

Origin		Destination		Distance		
Latitude	Longitude	Latitude	Longitude	PRoA	GoogleMaps	Reduction (%)
37.178013	−3.595665	37.183169	−3.599816	777	816	4.8
37.178346	−3.596976	37.178904	−3.596161	142	230	38.3
37.179344	−3.597739	37.178037	3.593710	517	550	6.0
37.175637	−3.594760	37.178022	−3.593719	444	500	11.2
37.177094	−3.595219	37.179450	−3.594187	348	450	22.7
37.182653	−3.598963	37.183898	−3.597441	281	350	19.7

(a) (b)

Fig. 5. Examples of different routes obtained by PRoA (left) and Google (right). On (a), case number 2 from Table 2 with a reduction of 38.3%. On (b), case number 4 from Table 2 with a reduction of 11.2%.



(a) (b)

Fig. 6. On top, three different user defined profiles are shown. The corresponding routes are plotted below, together with their elevation profiles.

5.1. Route lengths comparison

Table 2 shows the origins and destinations of six routes, together with their distances according to PRoA and Google Maps. All the routes are located in Granada (Spain). PRoA was ran with the "reduce distance" profile for a fair comparison.

As it can be observed, PRoA routes are shorter than those provided by Google Maps. The reduction in length goes from almost 5% to an impressive 38% in the shortest route (142 m vs. 230 m). This is a crucial point from a pedestrian point of view. The routes for two of those examples are shown in Fig. 5.

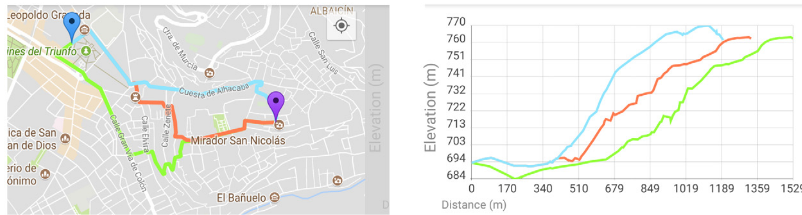
The reason underlying these differences is the OpenStreetMap data completeness for pedestrian routes. This fact was already known (Haklay, 2010), but we confirm it in the case of Granada where a lot of small streets, alleyways and passages are properly mapped in OpenStreetMap, but not in the Google proprietary maps.

5.2. Beyond distance minimization

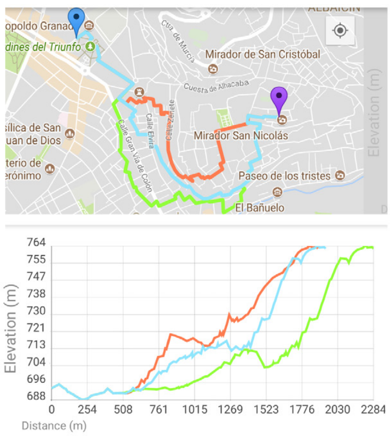
In this part, we emphasize the role of "personalization" showing how different routes can be obtained for different user profiles.

Example 1 (*Different User Profiles*). The first example is shown in Fig. 6. Three different user defined profiles appear on top, each corresponding with the following sets of weights (preferences) W: for Option 1 $\{w_d = 1\}$, Option 2 $\{w_{up} = 1\}$ and Option 3 $\{w_d = 0.5, w_{up} = 0.5\}$. Fig. 6(a) shows the three routes obtained according to the profiles and Fig. 6(b) allows the user to analyze the elevation profiles. It can be observed how the route corresponding to option 2 allows to reach the destination using a larger but "softer" (in terms of elevation) path.

Example 2 (*Different Constraints*). In this second example, shown in Fig. 7, we illustrate the differences on the routes obtained when

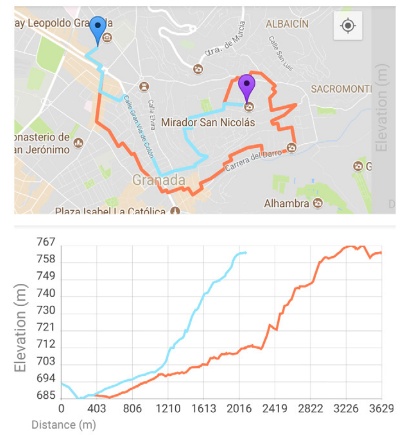


Case 1



Case 2

Case 3



Case 4

Fig. 7. Use of limitations. Case 1: "reduce distance" profile and no limitations. Case 2: "pass through green zones" profile with a limitation of 30% on maximum slopes and stairs allowed; Case 3: "pass through green zones" profile with a limitation of 30% on maximum slopes and stairs not allowed; and Case 4 "avoid slopes" profile with a limitation of 25% on maximum slopes and stairs not allowed. The corresponding elevation profiles are also shown. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

setting slope limitations and not allowing stairs. The routes go from (37.184332, −3.602442) to (37.181191, −3.592776). First, routes on Case 1 are obtained under the "reduce distance" profile with no other constraints. Cases 2 and 3 are both obtained with the "pass through green zones" profiles. Case 2 has a constraint on maximum slope of 30% allowing stairs while routes from Case 3 has the same slope constraint but does not allow stairs. Finally, Case 4 uses the "avoid slopes" profile with a constraint on maximum slope of 25%, also not allowing stairs.

All of these examples clearly demonstrate that PRoA properly deal with and solve the personalized route problem oriented to pedestrians. A wide variety of routes are obtained as the use of different user profiles and requirements are defined.

5.3. Performance analysis

We made a brief performance analysis to provide information regarding the running time of the application. We define eight routes in Granada, that lead to different underlying graphs ranging from 4000 to 42 000 nodes. We run PRoA with the "reduce distance" profile, measuring the computational time of every computational stage.

We consider two situations for the analysis: when the map information (the database) is not available in the mobile phone and when it is. Fig. 8(a) shows the former situation while (b), the latter.

It is clear that the information request from Web Services is the most time consuming part, which is also completely out of our management. The times displayed should be considered just as an illustrative examples because a high variability should be expected due to different response time from OpenStreetMap servers throughout the day.

When the map is already available in the device, the time needed to manage the stored information is clearly reduced. In the biggest case (a graph with 41563 nodes), the map processing stage took less than a minute.

Regarding the computational times of the graph evaluation and the routing stages, we can observe a linear behavior of the algorithms with a very small slope, indicating good scalability and speed.

6. Conclusions and future work

Personalization of pedestrian routes becomes a necessity due to the wide variety of user profiles that may differ on preferences or requirements to choose a route. Several software applications offer routes usually based on single criterion like distance or time; however, these criteria do not often fit the pedestrian needs.

In this context we modeled and solved the Personalized Routes Problem, focusing in the case of pedestrian routes design. We presented PRoA: an intelligent personalized route assistant for walking routes, implemented as an Android application. The personalization process is simple and easy to use when considering the predefined profiles or the user defined profile.

From the analysis developed in the city of Granada, Spain, several conclusions can be drawn. In first place, PRoA routes can be shorter than those provided by Google Maps. This is due to the higher quality of OpenStreetMaps from which PRoA extracts the map information. In second place, PRoA is able to offer routes considering characteristics that other routing applications do not even consider. We provide a set of examples showing how the use of slopes limitation, green zones and stairs avoidance effectively led to different routes, which is of utmost importance for users with different preferences. Finally, and in terms of performance, we confirmed that the running time increased in a linear way (with a very small constant) as the number of nodes in the underlying graph increased.

As future research we plan to consider different evaluation/aggregation schemes and explore personalization in other terms, like safer routes or cycling routes which requires different sets of criteria.

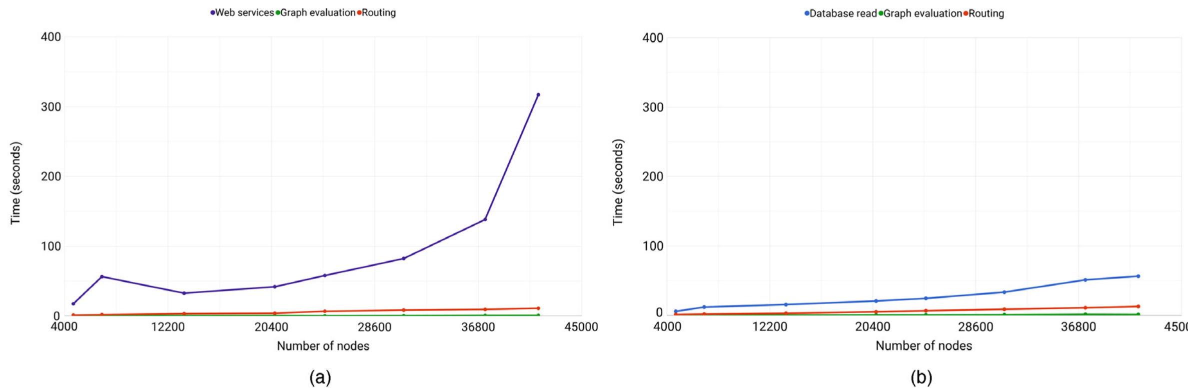


Fig. 8. Running times versus map size (number of nodes on the graph) (a) requiring the download of the database and (b) when the database is already available.

Acknowledgments

Research supported in part by projects TIN2014-55024-P (Spanish Ministry of Economy and Competitiveness) and P11-TIC-8001 (Consejería de Economía, Innovación y Ciencia, Junta de Andalucía). Both projects include FEDER funds from the European Union.

M. Torres enjoys a Ph.D. research training staff grant associated with the project TIN2014-55024-P and co-funded by the European Social Fund.

References

Bader, R., Dees, J., Geisberger, R., Sanders, P., 2011. Alternative route graphs in road networks. In: Marchetti-Spaccamela, A., Segal, M. (Eds.), Theory and Practice of Algorithms in (Computer) Systems. Springer Berlin Heidelberg, ISBN: 978-3-642-19754-3, pp. 21–32.

Balstrøm, T., 2002. On identifying the most time-saving walking route in a trackless mountainous terrain. Danish J. Geogr. 102 (October), 51–58. http://dx.doi.org/10.1080/00167223.2002.10649465.

Borst, H.C., de Vries, S.I., Graham, J.M., van Dongen, J.E., Bakker, I., Miedema, H.M., 2009. Influence of environmental street characteristics on walking route choice of elderly people. J. Environ. Psychol. (ISSN: 02724944) 29 (4), 477–484. http://dx.doi.org/10.1016/j.jenvp.2009.08.002.

Cecchini, I.B.A., Congiu, T., Fancello, G., Trunfio, G.A., 2014. Walkability explorer: An evaluation and design support tool for walkability. In: Computational Science and Its Applications–ICCSA 2014. In: Lecture Notes in Computer Science, vol. 8582, Springer International Publishing, ISBN: 978-3-319-09146-4, pp. 511–521. http://dx.doi.org/10.1007/978-3-319-09147-1.

Craig, C.L., Brownson, R.C., Cragg, S.E., Dunn, A.L., 2002. Exploring the effect of the environment on physical activity. Amer. J. Prevent. Med. (ISSN: 07493797) 23 (2), 36–43. http://dx.doi.org/10.1016/S0749-3797(02)00472-5.

Haklay, M., 2010. How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets. Environ. Plan. B: Plan. Des. URL http://kfrichter.org/crowdsourcing-material/day1/haklay10.pdf.

Hochmair, H.H., 2008. Grouping of optimized pedestrian routes for multi-modal route planning: A comparison of two cities. In: The European Information Society. In: Lecture Notes in Geoinformation and Cartography, Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN: 978-3-540-78945-1, pp. 339–358. http://dx.doi.org/10.1007/978-3-540-78946-8.

Leslie, E., Coffee, N., Frank, L., Owen, N., Bauman, A., Hugo, G., 2007. Walkability of local communities: using geographic information systems to objectively assess relevant environmental attributes. Health Place (ISSN: 1353-8292) 13 (1), 111–122. http://dx.doi.org/10.1016/j.healthplace.2005.11.001.

López-Ornelas, E., Zepeda-Hernández, S., Abascal-Mena, R., 2014. A geo-collaborative recommendation tool to help urban mobility. Hum.-Comput. Interact. Appl. Serv. 8512, 466–472. http://dx.doi.org/10.1007/978-3-319-07227-2.

Naranjo, J.C., Bayo, J.L., 2008. VADEO, URL http://www.vadeo.es/.

OpenStreetMap, 2004. Openstreetmap wiki. URL https://wiki.openstreetmap.org/wiki/Main_Page.

Owen, N., Humpel, N., Leslie, E., Bauman, A., Sallis, J.F., 2004. Understanding environmental influences on walking; Review and research agenda. Am. J. Prev. Med. 27 (1), 67–76. http://dx.doi.org/10.1016/j.amepre.2004.03.006. ISSN 0749-3797.

Pate, R.R., 1995. Physical activity and public health. JAMA (ISSN: 0098-7484) 273 (5), 402. http://dx.doi.org/10.1001/jama.1995.03520290054029.

Quercia, D., Schifanella, R., Aiello, L.M., 2014. The shortest path to happiness: Recommending beautiful, quiet, and happy routes in the city. In: Proceedings of the 25th ACM Conference on Hypertext and Social Media. ACM, pp. 116–125.

Saelens, B.E., Sallis, J.F., Frank, L.D., 2003. Environmental correlates of walking and cycling: Findings from the transportation, urban design, and planning literatures. Ann. Behav. Med. 25 (2), 80–91. http://dx.doi.org/10.1207/S15324796ABM2502_03. ISSN 0883-6612.

Torres, M., 2016. Proa: A personalized route assistant application. Technical report, University of Granada, Spain. URL https://play.google.com/store/apps/details?id=com.application.proa.

Tranter, M., Robineau, D., Goodman, G., 2015. National travel survey: 2014. Technical report, Department for Transport, UK. URL https://www.gov.uk/government/statistics/national-travel-survey-2014.

Walkonomics, 2011. Find a Walkable Route. URL http://www.walkonomics.com/about.

Zielstra, D., Hochmair, H., 2012. Using free and proprietary data to compare shortest-path lengths for effective pedestrian routing in street networks. Transp. Res. Rec.: J. Transp. Res. Board (ISSN: 0361-1981) 2299, 41–47. http://dx.doi.org/10.3141/2299-05.