CAPRA: A Contour-based Accessible Path Routing Algorithm

Mohammad Saiedur Rahaman[a,*], Yi Mei[a,b], Margaret Hamilton[a], Flora Dilys Salim[a]

[a] *Computer Science and Information Technology, School of Science, RMIT University, Melbourne, VIC 3000, Australia.*
[b] *School of Engineering and Computer Science, Victoria University of Wellington, Wellington 6012, New Zealand.*

Abstract

Existing journey planners and route recommenders mainly focus on calculating the shortest path with minimum distance or travel time. However, elderly people and those with special needs (i.e. those in wheelchairs or walking with sticks) often prefer a safer and more gentle journey. Given that their route options are affected by accessibility issues such as climbing a steep slope, it is important to design a journey planner that takes in to account the accessibility of the route, as well as the standard metrics, such as travel time and distance. Accessibility has not been explored widely in path finding problems. There are two key challenges for computing accessibility. First, the accessibility of a route is not well-defined. Second, the accessibility of a route varies from user to user. In this paper, a new algorithm is designed to tackle the above two challenges. Two metrics are defined to reflect the accessibility of a route, in terms of the total vertical distance and the maximum slope. Then, a multi-objective A* search algorithm is designed to obtain a set of Pareto-optimal routes in terms of the total distance covered and the two accessibility metrics. The user can then choose from the routes provided by the new algorithm, the most suitable one according to their own preferences. The experimental results show that the proposed algorithm is able to

*Corresponding author. Tel: +61 416800303.
[1]Email addresses: saiedur.rahaman@rmit.edu.au (Mohammad Saiedur Rahaman), margaret.hamilton@rmit.edu.au (Margaret Hamilton), flora.salim@rmit.edu.au (Flora Dilys Salim).
[2]Email addresses: yi.mei@ecs.vuw.ac.nz (Yi Mei).

provide a diverse set of routes with different accessibility options, including the shortest path which does not consider any accessibility metrics. In other words, the new journey planner can satisfy the preferences of a wide range of users including both the healthy and those with special needs.

Keywords: Journey Planning, Accessible Path, Route Ranking, Path Routing, Multi-Objective A* Algorithm.

1. Introduction

The world's population is ageing and people aged 65 and above is increasing at an alarming rate. According to World Bank's report [37], Australia has 15% of their total population aged 65 and over in 2014 whereas in United States, Singapore and Portugal this rate is 14%, 11% and 19% respectively. Globally, within the older population group, older persons aged 80 years or over account for 14% of the total population in 2013. It is projected to reach 19% in 2050 which is equivalent to 392 million persons aged 80 years or over by 2050 [39]. So, it is becoming more and more important to consider the special needs of this increasingly large number of people when developing public services for them. Among the various special needs of elderly people, mobility needs are becoming more important as more people retire from driving, and thus require improved journey planner options which may combine accessible public transport and walking routes to meet their mobility needs. For example, they will require mobility to access health care services, various social activities, shopping, and simply maintain community connections. However, there are many perceived barriers which limit the usual mobility requirements of elderly people and those with special needs. Of these barriers, accessibility issues are considered the most important [45]. A review in [4] shows that public transport has a significant influence on access to various health services for elderly people and those who stop driving their own vehicles. Other research points out that special consideration must be given while constructing or upgrading road and footpath infrastructure [46], for instance, as this can impact on pedestrians who use a cane, guide dog or wheelchair. It is important to meet everyone's mobility needs so that they do not become isolated from society [5]. Although there has been improvement in aspects of public transport and civil engineering to improve accessibility, the following question still remains:

"Which is the most accessible route to take between two points-of-interest (POIs) within walking distance?"

By point-of-interest, we mean a place where a journey starts or finishes (e.g. home, hospital, public transport station, or community place). The proliferation of mobile technologies and navigation services can help to provide solutions to this question. It has now become easier to go from one place to another by using various navigation devices. Route recommendation systems that are available compute choices of routes from a list of recommendations based on various criteria such as shortest route and fastest route. Although these systems are built to help people to be mobile, they cannot always satisfy every type of user. For example, a person with a manual wheel chair, who may be querying a route recommender system to travel between two locations, may not be satisfied with the outcome of their query. He may be directed to a path which is inaccessible or too steep and risky for him. This happens as the recommender only considers paths that are shortest and fastest. But, for this special user, route accessibility is the main key factor that needs to be considered. Moreover, recommendations for a route based on accessibility needs depend on the person's physical capability. For a daily commute, it may be considered less necessary to use a recommender system since the user would be well aware of the environment. However, the situation is different when the user wants to visit a new or unknown place. It is particularly necessary to design an accessible path recommendation for the elderly people and people with special needs to fit their physical abilities.

Route accessibility is very difficult to model as there are many factors that can affect the accessibility of a route. Of them, the most influential factor is the gradient of the route. To be specific, people with a wheelchair can comfortably wheel themselves up a specific gradient but not beyond a slope of one-in-fourteen [41]. How to give a route recommendation that is accessible for a wheelchair is a challenge, since there are so many possibilities that can happen along the path. For example, a very smooth route can be rendered inaccessible by a very sharp rise in gradient over a very small portion of the route. On the other hand, there may have several routes with a gentle rise in gradient in several portions but all of them could be accessible because this rise is below a certain margin. The challenge is to pick the best route from all the latter options. The existing path planning algorithms try to minimize the total travel distance or travel time. However, there is no measure defined for evaluating the accessibility of a path either. In the accessible path routing problem, there are the following challenges: *First*, the current network graph used for journey planning does not take into account the slope of the paths, and thus does not support the accessibility optimization. *Second*, there is no measure for evaluating the accessibility of the path. *Third*, there is no algorithm proposed for the accessible path routing.

In this paper, we propose a Contour-based Accessible Path Routing Algorithm (CAPRA) to address the above challenges. Specifically, the contribution of this paper is listed as follows:

- A Contour-based graph generation and query-based adaptation scheme is proposed to represent the slope of the paths in the graph with the aid of contour lines;
- Two metrics, total vertical distance and maximal slope, are defined for evaluating the accessibility of a path;
- A Multi-Objective A* search algorithm is designed for minimizing the total distance, total vertical distance and maximal slope;
- A diverse set of trade-off paths is provided, including the shortest path. The users can choose the most suitable path according to their own preference.

The rest of the paper is organized as follows. Section 2 introduces the related work on route recommendation. Section 3 describes the proposed route recommendation algorithm (CAPRA), which includes the data preprocessing, accessibility metrics definition, and multi-objective path finding. The experimental studies are carried out in Section 4, comparing the proposed algorithm with Google directions [13] solution, which is the representative state-of-the-art solution. Section 5 discusses and summarizes the results obtained in Section 4. Finally, Section 6 concludes the paper and presents some directions for future work.

2. Related Work

Some researchers focus on identifying different mobility aspects for the elderly and the people with special needs. Another direction of research focuses on different techniques for collecting information on physical accessibility barriers along the path whereas a number of researchers consider different parameters for calculating the score of a path. Also, there is another direction of research where the aim is to develop systems for mobility assistance.

2.1. Mobility Aspects for the Elderly and People with Special Needs

Several surveys have been conducted to identify the mobility aspects and accessibility barriers for the elderly and people with special needs [24, 10, 27, 2, 30]. A spatial analysis of accessibility of train stations and access to their surroundings for elderly passengers is presented in [24], where the authors leveraged data from State Government organizations and conducted a field survey of seven railway stations in Perth, Western Australia. The survey identified the trip purposes and attitudes towards accessibility for the elderly travelers. This research found that accessibility at the train station and surrounding areas is affected by route directness, facility and service quality at station, mixed land use, and intermodal connectivity. The researchers calculated accessibility indices for train stations and surroundings by combining elderly patronage rates and identifying variables that affect accessibility. They classified the data into three types of elderly passengers: those who walk and ride, park and ride and those who take the bus and ride, since the main form of public transport in Perth is bus. However, the research did not consider the route accessibility that can have an impact on the elderly and passengers with special needs and influence their attitudes towards their patronage of public transport in a major way. The research in [10] presented a way to determine the accessibility of public transport and evaluate the service quality by analyzing pervasive mobility data. The authors in [27] conducted a survey to learn about the opportunities and barriers associated with ridesharing from an elderly person's point of view. Some research also focuses on blind passengers' travel needs. An interview with a group of blind and deaf-blind public transport users revealed that they are primarily concerned with independence and safety [2]. In [30], the routing behaviour of pedestrians in an indoor environment is investigated by evaluating responses to active RFID and QR-code based route navigation systems for blind people. Such systems were also evaluated in [1], which recognized that all of these systems must work in an integrated manner to achieve desired accessibility outcomes for the individuals concerned.

2.2. Crowdsourcing as a Tool for Data Collection and Route Recommendation

Several studies have collected data on accessibility barriers along a path through crowdsourcing [28, 19, 3, 34, 35, 21, 38, 18, 9]. Crowdsourcing has been widely used for accessibility data collection in general as well as for pedestrian navigation. An accessibility information sharing platform for people with disabilities was explained in [28], which aimed to provide disabled people with a suitable path to their destination. The authors in [19] proposed and onlined crowdsourcing techniques with the Google Street View application to identify the bus-stop landmark locations and improve the accessibility of blind riders. Crowdsourcing was also used to collect information on stop identification landmarks in [2]. A platform for collaborative accessibility map generation was proposed in [3]. The system allowed users to add photos of the side walk accessibility barriers and comment on them. The authors in [34] designed a system which they call mPASS to collect indoor and outdoor accessibility data as well as analyse many outdoor accessibility requirements. The system crowdsourced notifications about a possible accessibility barrier (such as stairs for example) to alert other users of the system to be aware but it may also need to be confirmed. By considering the user preferences and specific needs, the system aimed to provide personalized paths for users. It stores the user profiles based on their needs and preferences, which are then updated by allowing the users to select their choices (neutral, like, dislike and avoid) against a specific accessibility barrier. A route recommendation system based on crowdsourced data was presented in [35], where the authors quantified the human perceptions of quietness; happiness and beauty to recommend paths. Crowdsourcing was also used to select a small set of paths from a large set of recommendations. A crowd-driven turn-by-turn path selection technique was proposed in [21], where the authors collect live traffic information through crowdsourcing and then ruled out the less important paths. Crowd perceptions about routing directions were collected through a series of routing questions. The research also proposed a strategy to select the most important set of routing questions. A route recommender system based on crowd-voting data from social media was introduced in [38]. The aim was to suggest the most pleasurable route for urban walking rather than recommending a route based on time/distance. A crowd-aided mobile platform for user safety perception management was presented in [18]. The authors also extended their work by finding the safest route between two locations in [9], leveraging the data collected through their mobile crowdsourced platform. Though crowdsourcing is an effective tool for data acquisition, it can suffer from various issues such as trust, missing data, incorrect data, etc.

2.3. Measuring Route Scores

Several authors have defined a walkability score for a pedestrian route or a specific location [47, 42, 23, 15, 36, 41, 43, 11]. A model for measuring walking accessibility towards public transport terminals was presented in [47] by introducing the concept of equivalent walking distance. The equivalent walking distance is the sum of the actual walking distance plus other factors along the route (crossing, ascending steps, conflict points), the values of which are measured by calculating the trade-off of that factor with respect to the actual walking distance. "Walkscore" is a publicly available system which provides a score for the walk and a transit score for a specific address [42]. It uses the distance of local amenities and transit facilities from an address to assign the score. A map route ranking method that considers environmental factors is presented in [23]. The direction and elevation services are used to select and rank the routes recommended by the Google Maps application [15]. However, this approach did not consider a context aware route search and the routes generated from Google Maps are based on shortest distance or minimum time and no on accessibility issues. Another model for recommending a walking route was proposed in [36]. Routes were generated by combining the A* algorithm and genetic algorithms and were evaluated against safety, amenity and walkability criteria. In the system, the user was required to enter the weights for each of these criteria to define the objective functions for each route. However, the safety was a qualitative measure and users might find it difficult to assign weights for different parameters. RouteCheckr [41] is a Dijkstra-based client/server architecture which aims to provide personalized routing to mobility impaired users. The system is based on multimodal annotation of geo-data. Users can rank their choices and then, based on the multi-criteria cost associated with each route, the best route is presented. The problem with the weighted sum approach is that all the parameters are required to be converted to a common scale. A traffic aware real-time route recommendation system was proposed in [43]. A combination of Dijkstra and A* algorithms was used to recommend the best route based on shortest time. The technique employed the real-time and historical taxi data. A bi-criteria optimization algorithm for urban navigation was proposed by [11]. The aim was to provide a set of paths that shows trade-off between distance and safety.

2.4. Mobility Assistance

A significant amount of mobility assistance can be made available to aid different groups of users. Considering the concept that blind travelers navigate through a place based on some landmarks, a braille-based application was developed by [2] that provides information on bus and bus-stop landmarks. It can become a problem if the landmark is not available due to any construction work. A train station navigation application for blind passengers was presented in [16] where descriptions of the station were stored at different levels of the tree structure: overview, floors, platforms and places of interest (POIs). The system starts with a basic overview of the station, i.e., how many floors the station has and how they are numbered with respect to the ground. The user can travel floor by floor and can have various descriptions about the POIs. M3I is an interactive platform for pedestrian navigation in both indoor and outdoor environments [44]. The platform incorporates speech and gesture recognition for navigation support. A rich overview of mobility assistance systems for elderly or mobility impaired persons was presented in [22], where the authors also explain the current status and usability of such systems.

Current literature does not consider topographical information which is one of the most influential factors in accessibility based path modelling. Also these researchers aim to achieve a single objective such as minimising distance, optimising safety, or increasing accessibility of the path. To the best of our knowledge, our work is the first that combines contour information from topographical map data with road network data to model path accessibility and aim for optimising two objectives: the distance and the accessibility of the path.

3. Contour-based Accessible Path Routing

In this section we model path accessibility in terms of path elevation and optimise two objectives of the paths: the distance and the accessibility. In summary our contributions are as follows:

- We try to optimise the accessibility along with the distance for the *first* time. No existing literature has addressed this issue before.

- For solving the problem, we develop a new multi-objective A* search algorithm known as Contour-based Accessible Path Routing Algorithm

(CAPRA), more particularly the admissible heuristic functions for all the objectives, so that we can guarantee to obtain *all* the Pareto-optimal solutions in query time.

- We propose a new graph model that contains both the distance information and the elevation information for the A* search.

The proposed Contour-based Accessible Path Routing Algorithm (CAPRA) mainly consists of the following three modules:

1. Data preprocessing: contour-based graph generation and query-based adaptation;

2. Accessibility evaluation of paths;

3. Path routing based on distance and accessibility.

First, in the data preprocessing phase, the contour line is adopted to generate a new contour-based graph so that the elevation difference of each road segment can be evaluated more precisely. We have developed two new accessibility metrics: the vertical distance and maximal slope based on the contour graph to evaluate the accessibility of a path. Finally, we have designed a multi-objective A* search algorithm to find the best trade-off paths in terms of both distance and accessibility.

3.1. Data Preprocessing: Contour-based Graph Generation and Query-based Adaptation

Contour-based Graph Generation. Many of the papers discussed in Section 2 consider the road network as a representation of a graph where the nodes are the intersections where the roads cross each other. A *road segment* is referred to an edge between two nodes in the road network. It is different from a road or a street. For example, in Figure 1 which shows an area in Melbourne City, Australia, the Queen Street is divided into several road segments, e.g., the one between Lonsdale Street and Little Bourke Street, and the one between Little Bourke Street and Bourke Street.

Finding the shortest path between two locations in the road network is a commonly encountered problem in journey planning and tourist trip design. Here, a path is a sequence of nodes in the road network connecting with road segments. Dijkstra [7] and A* [20] search algorithms, and their

Figure 1: Google Map of an area in Melbourne City, Australia.

variants [40, 12, 6, 8, 26] are mostly used to find the shortest path in terms of distance or travel time. There is no doubt about the effectiveness of such algorithms. However, this approach to representing the road network has a major drawback because the accessibility of one route segment might not always give a true reflection of the accessibility if the route segment is only the connection between two road crossings. It could happen that a route segment with a good accessibility rating may contain a very small portion which is wheelchair inaccessible due to a steep slope or steps. In this paper, we consider this issue to be very important. That is, a road segment can have a number of different slopes in between corners and intersections. For example, in Figure 2, the road segment AB has two different slopes, one upward from A (elevation of 60m) to C (elevation of 75m), and the other downward from C to B (elevation of 60m). The road network is a planar graph, only considering the latitude and longitude values. Therefore, the accessibility of AB given by the road network (the elevation difference between A and B) will be much different from its actual accessibility (the elevation difference between C and A, and C and B).

In practice, it is challenging to identify the exact geographical locations of the turning points between the slopes in a road segment (the exact location of the point C in Figure 2 for example). Therefore, in this paper, a contour-based graph generation is developed to approximate the locations of such turning points. Specifically, a new graph is generated by including the contour lines on the road network, and adding new nodes at the cross-sections

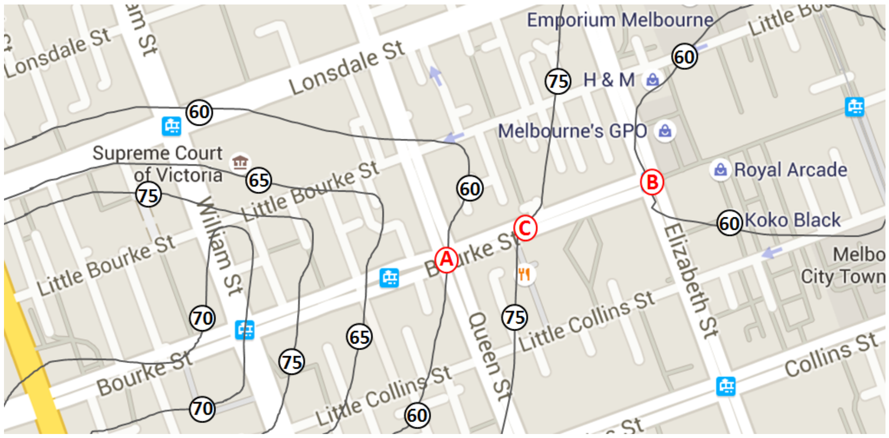


Figure 2: A contour map showing road segment AB with two different slopes.

between the contour lines and the road segments. A contour line is the line of geographical surface points on the map connecting points of the same elevation. Contour lines can be drawn for any elevation value on the earth's surface. Figure 2 can be considered as an illustration of a partial contour map of Melbourne, Australia, in which the grey curve lines on the map are contour lines. This contour map is an example of 5 meter contour interval which can be generated using the Open Street Map application, Srtm2Osm [33] for any location on the earth. The Srtm2Osm is a module which can generate the contour lines from the digital elevation model provided by the Shuttle Radar Topography Mission (SRTM) [29]. It can be seen that there are many crossing points between the contour lines and the road segments on the map. We include these crossing points as nodes with our road network graph.

In Figure 2, we can see that there is only one contour line that intersects road segment AB at point C. Therefore, we add C to the road segment AB. After combining the road network and the contour lines, the nodes in the graph are defined as the union set of the intersections of the roads with the crossing points of other roads and other contour lines. As a result, there are more nodes and edges in the newly generated graph than the original one. For example, the original road segment AB is divided into two smaller segments AC and BC. The operations (i.e., intersection and union) required for contour-based graph generation can be seen to be similar to the *ll_intersects* and *pp_plus* operators respectively as described by Güting et al. in [17]. The *pp_plus* operator outputs the union of two point objects. It scans and merges the point sequences from two point objects into a new points object. Given two line objects L_1 and L_2, the *ll_intersects* operator outputs whether they intersect or not. The output is true if both objects have no segments in common but at least one common point which is an intersection point but not a meeting point. Note that the elevation interval is an important parameter, since it determines both the accuracy of the turning point approximation and the number of new points added, and thus the size of the newly generated graph.

Each node in the contour-based graph, has a latitude and a longitude value given by the road network. In addition, the elevation value can be obtained by the Google Elevation API [14]. Therefore, the contour-based graph can be seen as a 3-D graph, where each node can be featured with the 3-dimensional vector (latitude, longitude, elevation). With the contour-based graph, one can calculate the elevation differences in different segments of a path much more accurately than by using only the pure road network.

In the proposed contour-based accessible path routing system, the contour-based graph is generated in the data preprocessing phase, and stored in an XML file. The details of the data preprocessing are described in Algorithm 1 below, where the road network is extracted from Open Street Map (OSM), and JOSM is a cross-platform OSM editor that can merge the contour lines and OSM road network and identify the crossing points of the contour lines and the roads.

Algorithm 1: Data preprocessing: contour-based graph generation

```
Extract the contour lines using Srtm2Osm [33];
Extract the road network using Open Street Map [32];
Combine the road network and the contour lines using JOSM [31];
Identify the crossing points between the contour lines and streets;
Generate the contour-based graph by adding the new crossing points and
  edges;
Generate the XML file for the contour-based graph using JOSM;
```

Query-Based Adaptation. The generated network graph only consists of the intersection points between the road segments and between the road segments and the contour lines. On the other hand, the query points (starting and ending points of the journey) can be anywhere on the map, and thus are highly likely to be outside the network graph. Therefore, it is necessary to include the query points into the graph in real time. Intuitively, a journey must start and end somewhere in the middle of a street. Therefore, the following scheme is proposed:

Step 1. Identify the existing edge on the graph that is closest to the query point;

Step 2. Remove the edge, and add an edge from the query point to each of the two end-nodes of the edge.

Figure 3 shows an example of including a query point into the network graph. Given the query point O, the closest edge AB is first identified and removed. Then, the two edges AO and BO are added into the network graph. The above procedure is applied to both the starting and ending point of the journey.

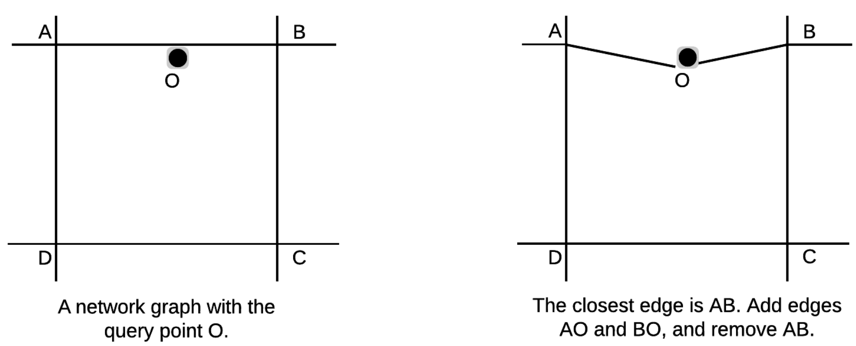


Figure 3: An example of including a query point into the network graph.

3.2. Accessibility Evaluation of Paths

The accessibility metrics of a path are derived from classical physics. In particular, assuming that the user keeps the same velocity while travelling along the path, the following two factors are closely relevant to the accessibility of a path: (1) the total energy consumed and (2) the maximal force needed to climb up the slopes along the path.

To facilitate this description, we take an example of a slope from point A to B in Figure 4, where a wheelchair user with gravity G is climbing up the slope whose steepness is α with a constant velocity v.

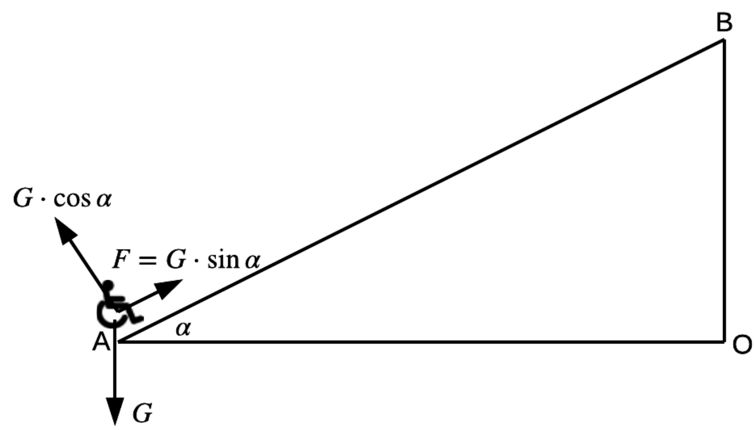


Figure 4: An example of moving up a slope of incline α from A to B.

According to the relationship between work and mechanical energy, when moving up a slope from point A to B, we have

$$W_{AB} = TME_B - TME_A, \tag{1}$$

where W is the energy consumed (work done) by the user for climbing from A to B (the elevation of B is higher than that of A), and TME_A and TME_B are the total mechanical energy of the user at points A and B, respectively. It is known that the total mechanical energy is the sum of the kinetic energy KE and the potential energy PE. Then,

$$TME_A = KE_A + PE_A = \frac{1}{2}mv^2 + Gz_A, \tag{2}$$

$$TME_B = KE_B + PE_B = \frac{1}{2}mv^2 + Gz_B, \tag{3}$$

$$W_{AB} = TME_B - TME_A = G(z_B - z_A), \tag{4}$$

where m is the mass of the user, v is the velocity of the user, which stays the same during the climbing, G is the gravity of the user, and z_A and z_B are the respective elevation of points A and B so that $|z_B - z_A|$ is the vertical distance between point A and B.

On the other hand, the driving force needed for climbing up the slope from A to B while maintaining the velocity is as follows:

$$F_{AB} = G \cdot \sin\alpha = G \cdot \frac{|BO|}{|AB|}, \tag{5}$$

where α is the steepness of the slope.

Similarly, when moving down from a higher point C to a lower point D, the two objectives are

$$W_{CD} = G(z_C - z_D), \tag{6}$$

$$F_{CD} = G \cdot \frac{|CO|}{|CD|}. \tag{7}$$

Note that $|AB|$ and $|CD|$ are not straightforward in practice. Therefore, they are replaced by $|AO|$ and $|OD|$, respectively, and $\sin\alpha$ is replaced by $\tan\alpha$ accordingly. Since α is always less than 90 degrees, minimizing $\sin\alpha$ is equivalent to minimizing $\tan\alpha$.

Then, given a path represented by a sequence of nodes $P = (v_0, v_1, \ldots, v_n)$, the total energy consumed $W(P)$ and the maximal force $F(P)$ needed to climb up and moving down all the slopes along the path are calculated as follows:

$$W(P) = \sum_{i=1}^{n} W_{v_{i-1}v_i}, \tag{8}$$

$$F(P) = \max_{i\in\{1,\ldots,n\}} \{F_{v_{i-1}v_i}\}, \tag{9}$$

where $W_{v_{i-1}v_i} = G \cdot |z_{v_i} - z_{v_{i-1}}|$, and $F_{v_{i-1}v_i} = G \cdot \frac{|z_{v_i} - z_{v_{i-1}}|}{d(v_{i-1},v_i)}$ in which $d(v_{i-1}, v_i)$ is the horizontal distance of path segment between v_{i-1} and v_i and $|z_{v_i} - z_{v_{i-1}}|$ is the vertical distance between nodes v_i and v_{i-1}.

Given that the gravity G of the user is a constant, and $-\pi/2 \leq \alpha \leq \pi/2$, Eqs. (8) and (9) can be simplified to

$$W(P) = \sum_{i=1}^{n} |z_{v_i} - z_{v_{i-1}}|, \tag{10}$$

$$F(P) = \max_{i \in \{1,\ldots,n\}} \left\{ \frac{|z_{v_i} - z_{v_{i-1}}|}{d(v_{i-1}, v_i)} \right\}. \tag{11}$$

Therefore, Eq. (10) illustrates the relationship between energy consumption and vertical distance travelled either up or down between successive points along the path. Note that the total energy consumption for traveling along a path is related to the sum of the vertical distances along the path. So, minimizing the total vertical distance during accessible path planning will reduce the total energy consumption. Eq. (11) shows the greatest force required to move up or down the biggest elevation difference. Here, the required maximal driving force for moving up or down a slope is related to the vertical distance and the length of the slope. The elevation difference between start and end point of a slope is crucial. A longer path segment requires less travelling force compared to a shorter path segment with similar vertical distance. Also people may want to choose a path which is the shortest of all.

3.3. Path Routing Based on Distance and Accessibility

When an elderly user or person with special needs is planning to travel along a path from a source to a destination on the map, both the distance and accessibility are critical factors to consider. To be specific, we assume the user would prefer the path with shorter distance and higher accessibility. However, in practice, the distance and accessibility may be in conflict with each other. In this case, one should provide a set of trade-off paths, which are termed the *Pareto-optimal* paths, instead of one single global optimal path.

The three objectives to be minimized in the accessible path routing can be described as follows:

$$\min_P f_1(P) = \sum_{i=1}^{n} d(v_{i-1}, v_i), \tag{12}$$

$$\min_P f_2(P) = W(P) = \sum_{i=1}^{n} |z_{v_i} - z_{v_{i-1}}|, \tag{13}$$

$$\min_P f_3(P) = F(P) = \max_{i \in \{1,\ldots,n\}} \left\{ \frac{|z_{v_i} - z_{v_{i-1}}|}{d(v_{i-1}, v_i)} \right\}, \tag{14}$$

where, $f_1(P)$ is the total horizontal distance of P, $f_2(P)$ is the total vertical distance of P, and $f_3(P)$ is the maximal slope of P, and $W(P)$ and $F(P)$ are defined in Eqs. (10) and (11) respectively.

Note that $f_2(P)$ is consistent with the energy consumed for moving up and moving down all the slopes. $f_3(P)$ is standing for the maximal force needed.

Given two paths P_1 and P_2, P_1 is said to *dominate* P_2 if and only if all the objective values of P_1 are no worse than those of P_2, and there is at least one objective for which P_1 has a better value than P_2. We denote $P_1 \prec P_2$ for P_1 dominating P_2. A path P^* is said to be *Pareto-optimal*, if and only if there is no other path that dominates P^*. The goal of this problem is to find all the possible Pareto-optimal paths.

In this paper, the multi-objective A* search algorithm is employed to find the Pareto-optimal paths. Specifically, the framework of the multi-objective A* search algorithm proposed in [25] is adopted here. The framework is described in Algorithm 2.

Two sets of labels $OPEN$ and $GOAL$ are defined where $OPEN$ is initialized with the source nodes and the algorithm steps through all nodes identifying non-dominated nodes which are stored in $GOAL$.

Once the target or destination node is reached, the elements in $GOAL$ and $OPEN$ are updated by removing the elements that are dominated by the new label. The search process stops when $OPEN$ becomes empty, and all the paths have been obtained by the backtracking procedure `Backtrack`$(GOAL)$. Further details of the multi-objective A* search algorithm can be found in [25].

For using the multi-objective A* search algorithm, the following two requirements must be satisfied:

1. the costs $\vec{c}(u, v)$ of all the edges $(u, v) \in E$ must be nonnegative;

2. the heuristic function is admissible, i.e., it never overestimates the actual minimal cost of reaching the goal.

Therefore, to design a multi-objective A* search algorithm for minimizing the objectives shown in Eqs. (12)–(14), we must design the cost functions $\vec{c}(u, v)$ and the heuristic functions `Heuristic`(v, t, G) that satisfy the above two requirements.

From Eqs. (12), (13), and (14), we set $\vec{c}(u, v)$ and `Heuristic`(v, t, G) as follows:

$$c_1(u, v) = d(u, v), \quad h_1(v) = d(v, t),$$

$$c_2(u, v) = |z_v - z_u|, \quad h_2(v) = |z_t - z_v|,$$

Algorithm 2: The framework of multi-objective A* search algorithm.

Input: The graph G, source node s and target node t

Output: A set of trade-off paths $\mathbf{P} = \{P_1, \dots, P_m\}$

// *Initialization*

foreach $v \in G$ **do** $\vec{g}_{cl}(v) \leftarrow \emptyset$, $\vec{g}_{op}(v) \leftarrow \emptyset$;

$GOAL \leftarrow \emptyset$, $OPEN \leftarrow \emptyset$;

$OPEN \leftarrow OPEN \cup (s, \emptyset, \vec{0}, \vec{h}(s))$, $\vec{g}_{op}(s) \leftarrow \vec{g}_{op}(s) \cup \vec{0}$;

// *Search*

while $OPEN$ *is not empty* **do**

  $L(u) := (u, \mathrm{pred}(u), \vec{g}(u), \vec{h}(u)) \leftarrow$ `Extract`$(OPEN)$;

  $OPEN \leftarrow OPEN \setminus L(u)$;

  $\vec{g}_{op}(u) \leftarrow \vec{g}_{op}(u) \setminus \vec{g}(u)$, $\vec{g}_{cl}(u) \leftarrow \vec{g}_{cl}(u) \cup \vec{g}(u)$;

  if $u = t$ **then**

    Add $L(u)$ into $GOAL$, and remove from $GOAL$ the elements with dominated $\vec{g}(\cdot)$;

    Remove from $OPEN$ the elements whose $\vec{f}(\cdot) := \vec{g}(\cdot) + \vec{h}(\cdot)$ are dominated by $\vec{g}(u)$;

  else

    foreach $v \in \mathcal{N}(u)$ **do**

      if *Adding* (u, v) *forms a cycle* **then continue**;

      $\vec{g}(v) \leftarrow \vec{g}(u) + \vec{c}(u, v)$; // *update* $\vec{g}(v)$

      $\vec{h}(v) \leftarrow$ `Heuristic`(v, t, G); // *calculate* $\vec{h}(v)$

      $L(v) := (v, L(u), \vec{g}(v), \vec{h}(v))$;

      if v *is a new node* **then**

        $OPEN \leftarrow OPEN \cup L(v)$, $\vec{g}_{op}(v) \leftarrow \vec{g}_{op}(v) \cup \vec{g}(v)$;

      else

        if $\vec{g}(v)$ *is non-dominated by any* $\vec{g} \in \vec{g}_{op}(v) \cup \vec{g}_{cl}(v)$ **then**

          Remove from $\vec{g}_{cl}(v)$ and $\vec{g}_{op}(v)$ the elements whose $\vec{g}(\cdot)$ are dominated by $\vec{g}(v)$;

          $OPEN \leftarrow OPEN \cup L(v)$, $\vec{g}_{op}(v) \leftarrow \vec{g}_{op}(v) \cup \vec{g}(v)$;

        end

      end

    end

  end

  return $\mathbf{P} \leftarrow$ `Backtrack`$(GOAL)$;

end

$$c_3(u,v) = \max\left\{\frac{|z_v - z_u|}{d(u,v)} - g_3(u), 0\right\}, \quad h_3(v) = 0,$$

$$\vec{c}(u,v) \leftarrow (c_1(u,v), c_2(u,v), c_3(u,v)),$$

$$\texttt{Heuristic}(v,t,G) \leftarrow (h_1(v), h_2(v), h_3(v)).$$

First, we note that $\forall (u,v) \in E$, $c_i(u,v) \geq 0$, $i = 1,2,3$. Then, for the total distance f_1, the heuristic $h_1(v)$ is admissible under the assumption of triangular inequality. For the total vertical distance f_2, for any other point $v' \neq v$ and $v' \neq t$, we have

$$|z_t - z_v| \leq |z_{v'} - z_v| + |z_t - z_{v'}|.$$

That is, $h_2(v) \leq c_2(v,v') + h_2(v')$. Therefore, $h_2(v)$ is admissible.

Finally, since $c_3(u,v) \geq 0$, $h_3(v) = 0$ is clearly admissible. In fact, since it is difficult to predict the maximal slope from any point to the target, we set $h_3(v) = 0$ to reduce the A* search in terms of f_3 to the Dijkstra algorithm. The function $g_3(\cdot)$ is naturally defined by A*. That is, $g_3(s) = 0$, where s is the source node, and for any edge (u,v), $g_3(v) = g_3(u) + c_3(u,v)$

In addition, since the function $\texttt{Extract}(OPEN)$ can return any elements with non-dominated $\vec{f}(\cdot)$, we choose the one with the shortest estimated distance $f_1(\cdot)$ so as to reach the target node as soon as possible and reduce the search space.

4. Experimental Studies

For the experimental studies, case studies are conducted for various hilly cities in the world, including San Francisco (USA), Lisbon (Portugal) and Singapore. These cities are good examples for the experimental studies as they are built on slopes which means that moving up and down hills usually occurs in these cities. In addition, different city layouts are taken into account and we selected four random journeys for our experiment. We selected San Francisco, because the streets are normally laid out as a grid system. However, for historical reasons, such rectangular city blocks are not common in many European and Asian cities. Therefore, we selected Lisbon and Singapore as the representative examples of the cities with more complex city layouts which are also hilly.

Note that there is no existing algorithm which takes the elevation into account when computing a path between two points. In addition, since we have designed CAPRA to employ the multi-objective A* search to find the paths, it is guaranteed to find the shortest path. In other words, CAPRA must include the optimal path in terms of distance and there is no need to compare with other shortest path finding algorithms. Here, we only compare CAPRA with the path produced by Google Directions API [13] to show its reasonableness in reality.

In the preprocessing phase, the contour interval is set to 5m. The reason behind the selection of such a small contour interval is that it allows us to obtain even small changes in elevation. Once the contour interval is selected, the corresponding contour-based road network graph is generated and stored in the memory. For each test scenario, both CAPRA and Google Directions API are applied and the paths obtained by them are compared in terms of the three accessibility measures, i.e., horizontal distance, vertical distance and maximal slope defined in Eqs. (12), (13), and (14) respectively . To evaluate the efficacy of our contour based graph generation technique, the accessibility measure values for both CAPRA and Google Directions API paths are also calculated without considering the contours and compared with the obtained accessibility measure values of CAPRA from contour-based graph built with the contour interval of 5m. Specifically, to calculate the accessibility measure values of a path in the later case, the path is first divided into 10m-long small segments. Then, for each segment, the vertical distance and slope are calculated. Finally, the total vertical distance of the path is obtained by summing up all the vertical distances and the maximal slope of the path is obtained by selecting the maximal segmental slope. Although these are still not true values, they are good approximations by choosing a sufficiently small segment length.

4.1. Case Study-1 in San Francisco, USA

Figure 5 gives an example from 817 Lombard St (point A) to 1132 Union St (point B), San Francisco, USA. We selected this path because the path mainly consists of upward slopes and the elevation of point B is higher than point A.

There are four paths from A to B shown in the figure. The solid path is the shortest path found by the Google Directions. The three dashed paths are the trade-off paths obtained by our new algorithm CAPRA. One can see that the first path CAPRA1 (brown dashed) obtained by CAPRA is the same as the one obtained by Google Directions. In addition, CAPRA has

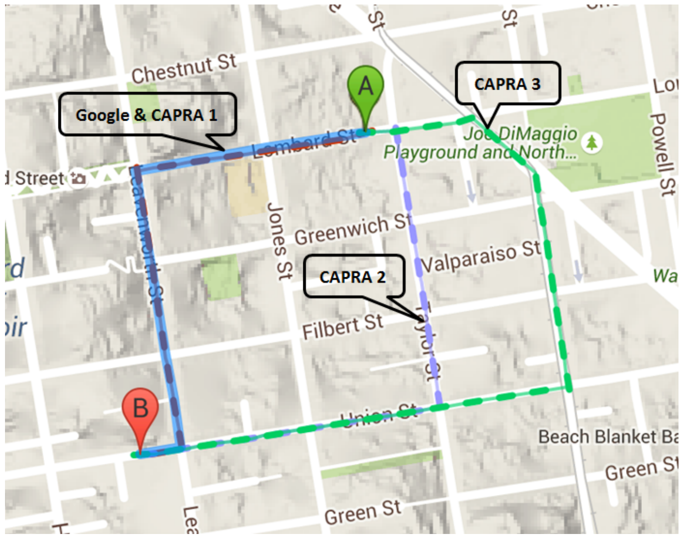


Figure 5: San Francisco, USA: The paths from 817 Lombard St to 1132 Union St. The solid path is obtained by Google Directions, and the dashed paths are obtained by CAPRA.

provided two other paths CAPRA2 and CAPRA3 (purple and green dashed respectively).

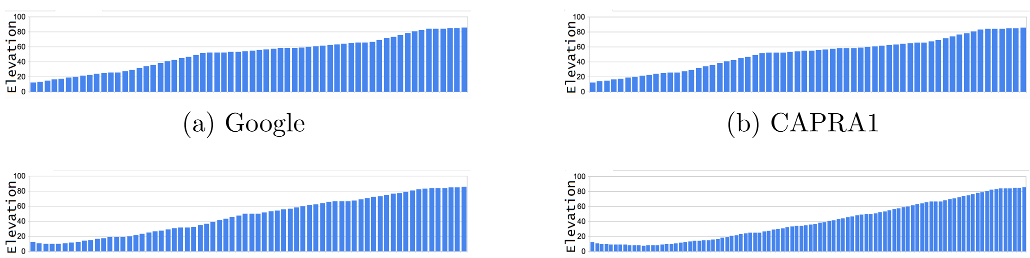
A comparison summary of *cost-benefit* between distance and accessibility measure values of the paths obtained from 5m-interval contour based network graph and 10m-long segment based network graph are given in Table 1. We can see that the CAPRA1 does not provide the best accessibility score in terms of slope. On the other hand, the CAPRA2 path has better slope score, but longer distance and vertical distance compared to CAPRA1. The CAPRA2 also provides shorter distance compared to the CAPRA3 but pays more in terms of slope. Therefore, the paths are non-dominated to each other. A user can choose the best path based of his/her distance and accessibility requirement.

We also can see that the accessibility measure values of the paths obtained by CAPRA from the contour based network graph is very close to the corresponding values from 10m-long segment based network graph. This implies that a contour interval of 5m is sufficient to build an accurate contour-based network graph. In addition, while increasing the length of the path, the maximal slope decreases from 0.23 to 0.15.

Figure 6 shows the elevation changes along the paths given in Fig. 5. It can be seen that the CAPRA3 path has many more segments than the other

Table 1: The accessibility measure values of the paths obtained by Google Directions and CAPRA in the scenario are shown in Fig. 5. “Distance”, “Vertical” and “Slope” stand for the total horizontal distance, total vertical distance $W(P)$, and maximal slope $F(P)$, respectively. There is no accessibility measure value for Google from the 5m contour interval network graph, since the path is obtained by the Google API.

Path	Considering 5m Contour interval			Considering 10m-long road segment		
	Distance(m)	Vertical(m)	Slope	Distance(m)	Vertical(m)	Slope
Google	-	-	-	623	72.7	0.23
CAPRA1	623	73.0	0.23	623	72.7	0.23
CAPRA2	688	78.6	0.21	688	78.3	0.22
CAPRA3	943	82.2	0.14	943	82.2	0.15



(a) Google

(b) CAPRA1

(c) CAPRA2

(d) CAPRA3

Figure 6: The elevation (in meters) changes along the paths given in Fig. 5.

paths due to the much larger horizontal distance. However, it achieved a much smoother slope (evidenced by the maximal slope of 0.15) by choosing the longer distance to travel.

4.2. Case Study-2 in San Francisco, USA

Fig. 7 shows another scenario from 1260 Green St (point A) to 1398 Lombard St (point B), San Francisco, USA, but with mainly downward slopes and the elevation of point B is much lower than point A. In this scenario, CAPRA obtained four different paths. CAPRA3 (green dashed) is the same as that obtained by Google Directions. It should be noted that CAPRA managed to obtain two shorter paths CAPRA 1 and CAPRA 2 (red and purple dashed) than Google Directions, but with larger vertical distance and maximal slope.

Table 2 shows a comparison summary of *cost-benefit* between distance and accessibility measure values of the paths obtained by Google Directions and CAPRA in the second scenario shown in Fig. 7. It can be seen that

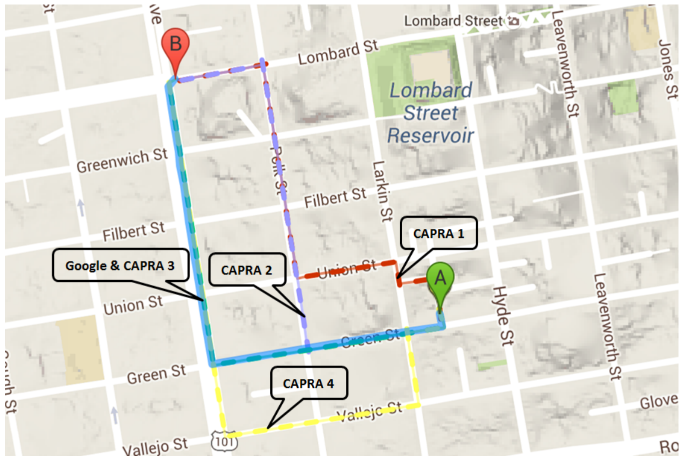


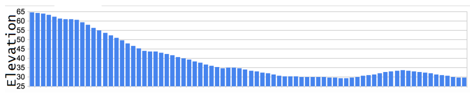
Figure 7: San Francisco, USA: The paths from 1260 Green St to 1398 Lombard St. The solid path is obtained by Google Directions, and the dashed paths are obtained by CAPRA.

when the length of the path increases, the vertical distance and maximal slope tend to decrease. This way, the users can choose the most suitable path based on their own preferences in terms of distance and accessibility.

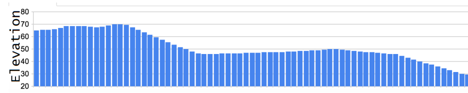
Fig. 8 gives the elevation changes for the paths shown in Fig. 7. It can be seen that for the path obtained by Google Directions and the first three paths obtained by CAPRA, the downward slopes are concentrated in the first half of the path (and the end of the path for CAPRA2). In contrast, the slopes are more uniformly distributed throughout the path for CAPRA4, which leads to a much smoother path overall.

Table 2: The accessibility measure values of the paths obtained by Google Directions and CAPRA in the scenario shown in Fig. 7. "Distance", "Vertical" and "Slope" stand for the total horizontal distance, total vertical distance $W(P)$, and maximal slope $F(P)$, respectively. There is no accessibility measure value for Google from 5m contour interval network graph, since the path is obtained by Google API.

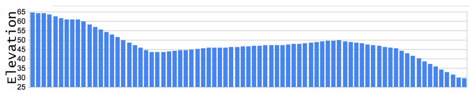
Path	Considering 5m Contour interval			Considering 10m-long road segment		
	Distance(m)	Vertical(m)	Slope	Distance(m)	Vertical(m)	Slope
Google	-	-	-	787	43.9	0.15
CAPRA1	730	55.6	0.19	730	55.6	0.21
CAPRA2	772	47.4	0.14	772	47.6	0.15
CAPRA3	787	43.6	0.14	787	43.9	0.15
CAPRA4	997	43.6	0.09	997	43.9	0.09



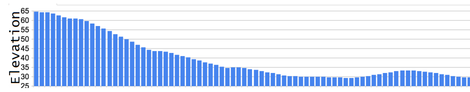
(a) Google



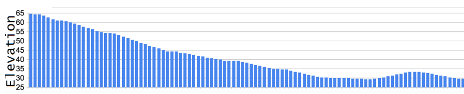
(b) CAPRA1



(c) CAPRA2



(d) CAPRA3



(e) CAPRA4

Figure 8: The downhill elevation (in meters) changes along the paths given in Fig. 7.

4.3. Case Study in Lisbon, Portugal

Fig. 9 shows a scenario from Rua São Boaventura 182 (point A) to Travessa Horta 21 (point B), Lisbon, Portugal. We selected this area of Lisbon because it is no longer a simple grid-like road network and is therefore more complex than that in San Francisco. The road network partly consists of some parallel streets (e.g., R. Vinha) which increases multiple routing possibilities. It can be seen that CAPRA obtained seven different paths in this scenario, none of which was the same as the Google path. A shortcut path CAPRA1 (brown dashed) was found, and the second path CAPRA2 (green dashed) was very similar to the Google path (turn right at a parallel street). In order to reduce the slope, two longer paths CAPRA6 and CAPRA7 (light and deep blue dashed) were also obtained, which have much reduced maximal slope. In this case, the irregular roads were employed as well.

Table 3 summarizes the *cost-benefit* between distance and accessibility measure values of the paths obtained by Google Directions and CAPRA in the scenario shown in Fig. 9. One can see that the CAPRA2 path has very similar distance and accessibility measure values to the Google path, due to the similar structure. For the paths obtained by CAPRA, although the value of the maximal slope for the paths from 5m contour interval is slightly higher than the 10m-long road segment one, the partial order is still consistent (i.e., a larger estimated value still leads to a larger real value). Therefore, one can

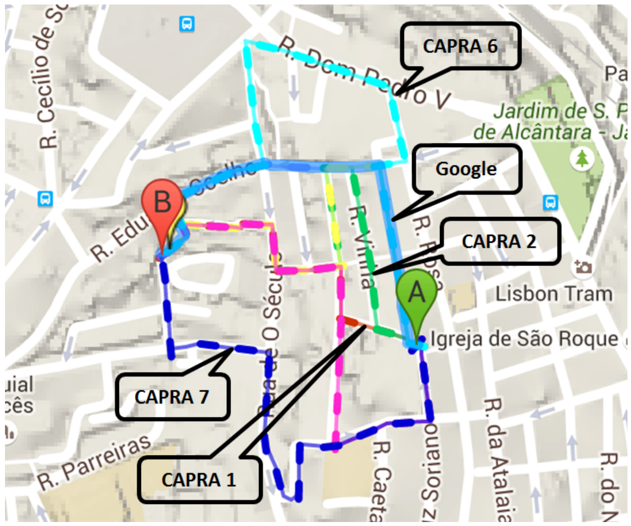


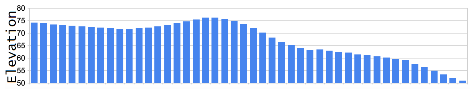
Figure 9: Lisbon, Portugal: The paths from Rua São Boaventura 182 to Travessa Horta 21. The solid path is obtained by Google Directions, and the dashed paths are obtained by CAPRA.

Table 3: The accessibility measure values of the paths obtained by Google Directions and CAPRA in the scenario shown in Fig. 9. "Distance", "Vertical" and "Slope" stand for the total horizontal distance, total vertical distance $W(P)$, and maximal slope $F(P)$, respectively. There is no accessibility measure value for Google from 5m contour interval network graph, since the path is obtained by Google API.

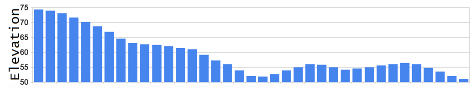
Path	Considering 5m Contour interval			Considering 10m-long road segment		
	Distance(m)	Vertical(m)	Slope	Distance(m)	Vertical(m)	Slope
Google	-	-	-	464	31.6	0.19
CAPRA1	376	36.8	0.18	376	35.4	0.23
CAPRA2	463	32.1	0.18	463	31.5	0.19
CAPRA3	568	51.7	0.17	568	50.4	0.20
CAPRA4	575	45.7	0.17	575	45.3	0.20
CAPRA5	601	37.7	0.17	601	36.6	0.19
CAPRA6	613	44.7	0.13	613	43.1	0.14
CAPRA7	720	36.8	0.13	720	36.2	0.14

still find the correct relative position of the paths on the Pareto front which is the set of Pareto optimal outcomes. It means that a CAPRA user is still able to choose a Pareto-optimal path which suits him/her best.

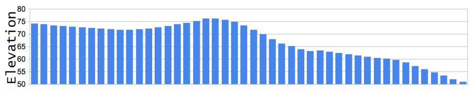
Fig. 10 gives the elevation changes over the paths given in Fig. 9. It can be seen that the vertical motions of the paths can be quite different from each other. For example, the first half of the Google path is relatively flat



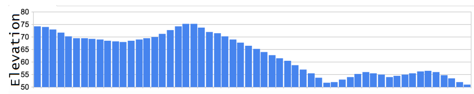
(a) Google



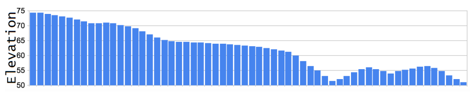
(b) CAPRA1



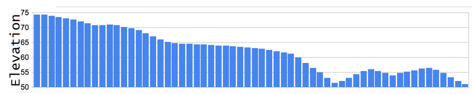
(c) CAPRA2



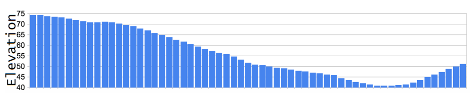
(d) CAPRA3



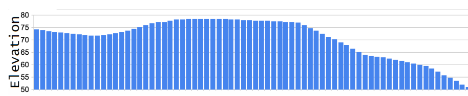
(e) CAPRA4



(f) CAPRA5



(g) CAPRA6



(h) CAPRA7

Figure 10: The elevation (in meters) change along the paths given in Fig. 9.

(slightly upward), while the CAPRA6 path keeps falling down until the last 15% of the path, and then goes up to reach the destination. They are trade-off paths and thus it is hard to tell which elevation change is better unless we look at elevation changes of each segment separately.

4.4. Bukit Timah, Singapore

Fig. 11 shows a scenario from 23 Victoria Park Rd (point A) to 21 Duke's Rd (point B), Singapore. We selected this place because the roads in Singapore are very hilly and do not follow a grid. In this case, only two paths were obtained by CAPRA. The first path CAPRA 1 (brown dashed) is same as the Google path.

Table 4 shows the *cost-benefit* between distance and accessibility measure values of the paths obtained by Google Directions and CAPRA in the scenario shown in Fig. 11. As in the other scenarios, CAPRA managed to reach a smoother slope at the cost of a longer distance.

Fig. 12 gives the elevation change through the paths given in Fig. 11. In this case, the elevation change of the three paths are similar to each other. This is because their directions are roughly the same, and a major portion of the paths are parallel to each other.

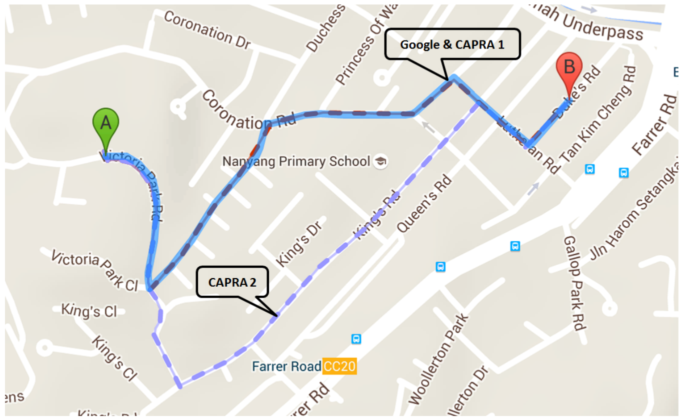
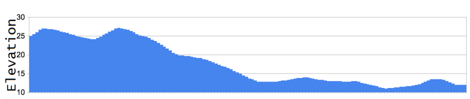


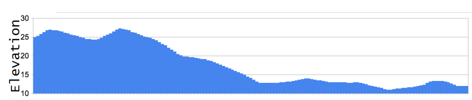
Figure 11: Singapore: The paths from 23 Victoria Park Rd to 21 Duke's Rd. The solid path is obtained by Google Directions, and the dashed paths are obtained by CAPRA.

Table 4: The accessibility measure values of the paths obtained by Google Directions and CAPRA in the scenario shown in Fig. 11. "Distance", "Vertical" and "Slope" stand for the total horizontal distance, total vertical distance $W(P)$, and maximal slope $F(P)$, respectively. There is no accessibility measure value for Google from 5m contour interval network graph, since the path is obtained by Google API.

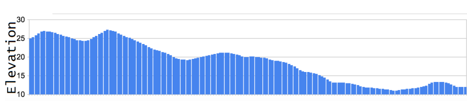
Path	Considering 5m Contour interval			Considering 10m-long road segment		
	Distance(m)	Vertical(m)	Slope	Distance(m)	Vertical(m)	Slope
Google	-	-	-	1444	30.7	0.06
CAPRA1	1444	28.8	0.06	1444	30.7	0.06
CAPRA2	1595	29.4	0.05	1595	31.7	0.05



(a) Google



(b) CAPRA1



(c) CAPRA2

Figure 12: The elevation (in meters) change along the paths given in Fig. 11.

5. Discussion

Overall, the results for all the above case studies show that CAPRA is able to provide a wide range of reasonably good paths in terms of both vertical distance $W(P)$ and slope $F(P)$, including the optimal path in terms of distance. In most of the cases, CAPRA can obtain the Google path, or the paths with the same measure values as the Google path. In addition, the trade-off paths with larger horizontal distances but smoother slopes are obtained as well. The estimated values of the CAPRA paths obtained from 5m contour intervals are close to their equivalent values obtained from 10m-long small road segments, which verifies the accuracy of the contour-based graph generation.

From the summary of *cost-benefit* analysis in Table 1-4, it can be seen that CAPRA can achieve a good trade-off between path length and accessibility (i.e., vertical distance and maximal slope). This way, the CAPRA users can choose the most suitable path based on their own preferences and accessibility requirements.

We note that there may be other physical accessibility barriers (i.e., stairs, ramps, traffic and road conditions) that can influence the accessibility of a walking path. For example, a path with an accessible elevation score may have a segment with stairs that cannot be traversed by people with wheelchairs. In this paper, we assume that those physical accessibility barriers are handled with care.

We further note the *computation complexity* for calculating the Pareto-optimal trade-off paths. The worst-case time complexity of the adopted MOA* framework is $O(d^b)$, where d is the length of the longest non-dominated path, and b is the branching factor, i.e. the number of neighbours of each node in the graph. This computation complexity is no more than the traditional MOA* presented in [25]. The search space is an issue for the multi-objective framework since the function $\texttt{Extract}(OPEN)$ can return any elements with non-dominated $\vec{f}(\cdot)$. Therefore, we choose the element with the shortest estimated distance $f_1(\cdot)$ so as to reach the target node as soon as possible. In this way our adopted multi-objective framework is able to reduce the search space. The experiments showed that our system can provide results in *query time* ($<$ 1 seconds) on normal machine (4GB RAM, Windows 7 OS, Intel Core-i7 CPU with 3.40 GHz clock speed) for all the test scenarios. Table 5 summarizes the four scenarios used in this research to illustrate the corresponding number of nodes, edges and trade-off paths.

The main contribution of our paper is to address the walk accessibility problem considering the elevation of the path. However, we believe that there might be several ways to speed up the algorithm. For example, to reduce

Table 5: Summary of the Four Scenarios.

Case Study	Total Nodes	Total Edges	Trade-off Paths
San Francisco-1	33,122	2,963	3
San Francisco-2	33,122	2,963	4
Lisbon	10,411	2,515	7
Bukit Timah	16,177	1,870	2

the search space and speed up the algorithm, we could return the shortest distance element until a complete path to the destination is found. From then on, we could return the element with the smallest vertical distance f_2 until a path to the destination with smaller f_2 is found. Then we could switch to returning the element with smallest f_3 until a path with smaller f_3 is found, then switch back to f_1 and so on. This way, we would make sure to successively decrease the limits for f_1, f_2, and f_3.

6. Conclusion and Future Works

In order to serve the elderly and disabled people and those with special needs, a new contour-based path planning system called CAPRA is proposed in this paper. This new algorithm considers the accessibility of the path as well as the horizontal distance. The paths constructed may well serve healthy commuters while travelling, bike-riding, roller-skating as alternative routes with more gentle slopes.

The contributions of the paper include a new contour-based graph generation for path planning. We have developed new accessibility measures for routes and have designed a multi-objective A* routing algorithm.

We have demonstrated the use of CAPRA in four different hilly environments where the path elevation could be very steep and problematic for a person in a wheelchair. The experimental studies on several representative hilly cities in the world shows that the proposed CAPRA can provide not only the standard shortest path which is the same as that provided by Google Directions or an A* algorithm, but also other alternatives which may be longer but have smoother slopes.

Our new algorithm can give the users a wider range of options to choose from. The users may not necessarily be elderly people or disabled but could instead, be bike riders or roller bladers or people pushing prams. In fact, anyone who might prefer to know about alternate routes to their required destination for a variety of preferences. In this paper, we have explained our three object preferences, but other preferences could also be implemented in the future, depending on the user needs.

In this paper, we address and define walking path accessibility considering the elevation of the path. However, there are other physical accessibility barriers present that may need to be taken into account. For example, the walk accessibility of a path may be affected by the stairs, high curbs and busy intersections. In future, a matrix such as the SAW criteria [36] and walkability score [42] can be incorporated with our approach to help the disabled and elderly people to check whether the route is affected by any physical accessibility barrier. The integration of such data can also be achieved through crowdsourcing, as some of these hazards are not permanent, but temporarily constructed for road maintenance or building construction, for example. For the purpose of real-time data collection, the crowdsourcing platform described in [18] could be used. The R-Q based method proposed in [21] is able to provide an answer to the routing queries related to traffic conditions. Also, it can be adapted with our model to provide live updates about the busy-ness of a road. In this regard, the urban data from pedestrian sensors could be utilized with the crowdsourcing platform. Also, user profiles could be incorporated to satisfy individual requirements.

Acknowledgment

This research was supported under Australian Research Council's Linkage Projects funding scheme (project number LP120200305). Also the authors acknowledge the contribution of Bo Wang who worked as a summer student on this project.

References

[1] S. Alghamdi, R. van Schyndel, M. Hamilton, Blind User Response to a Navigational System to Assist Blind People Using Active RFID and QR-Code, in: Proceedings of the 8th International Conference on Pervasive Computing Technologies for Healthcare, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014, pp. 313–316.

[2] S. Azenkot, S. Prasain, A. Borning, E. Fortuna, R. Ladner, J. Wobbrock, Enhancing Independence and Safety for Blind and Deaf-blind Public

Transit Riders, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2011, pp. 3247–3256.

[3] C. Cardonha, D. Gallo, P. Avegliano, R. Herrmann, F. Koch, S. Borger, A Crowdsourcing Platform for The Construction of Accessibility Maps, in: Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility, ACM, 2013.

[4] K. Corcoran, J. McNab, S. Girgis, R. Colagiuri, Is Transport a Barrier To Healthcare for Older People With Chronic Diseases?, Asia Pacific Journal of Health Management 7 (1) (2012) 49–56.

[5] J. Davey, Older People and Transport: Coping Without A Car, Ageing and Society 27 (1) (2007) 49–65.

[6] D. Delling, P. Sanders, D. Schultes, D. Wagner, Engineering Route Planning Algorithms, in: Algorithmics of Large and Complex Networks, Springer, 2009, pp. 117–139.

[7] E. Dijkstra, A Note on Two Problems in Connexion with Graphs, Numerische mathematik 1 (1) (1959) 269–271.

[8] Y. Disser, M. Müller-Hannemann, M. Schnee, Multi-criteria Shortest Paths in Time-dependent Train Networks, in: Experimental Algorithms, Springer, 2008, pp. 347–361.

[9] S. Elsmore, I. Subastian, F. Salim, M. Hamilton, VDIM: Vector-based Diffusion and Interpolation Matrix for Computing Region-based Crowdsourced Ratings: Towards Safe Route Selection for Human Navigation, in: Proceedings of the 13th International Conference on Mobile and Ubiquitous Multimedia, ACM, 2014, pp. 212–215.

[10] L. Ferrari, M. Berlingerio, F. Calabrese, B. Curtis-Davidson, Measuring public transport accessibility using pervasive mobility data, IEEE Pervasive Computing (2013) 26–33.

[11] E. Galbrun, K. Pelechrinis, E. Terzi, Safe Navigation in Urban Environments, in: The 3rd International Workshop on Urban Computing (UrbComp 2014), 2014.

[12] R. Geisberger, P. Sanders, D. Schultes, D. Delling, Contraction hierarchies: Faster and Simpler Hierarchical Routing in Road Networks, in: Experimental Algorithms, Springer, 2008, pp. 319–333.

[13] Google, Google Directions API, `https://developers.google.com/maps/documentation/directions/`, Last Accessed: 3-July-2015.

[14] Google, Google Elevation API, `https://developers.google.com/maps/documentation/elevation/`, Last Accessed: 26-Feb-2015.

[15] Google, Google Maps, `http://www.maps.google.com`, Last Accessed: 10-Jan-2014.

[16] M. Guentert, Improving Public Transit Accessibility for Blind Riders: A Train Station Navigation Assistant, in: The Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility, ACM, 2011, pp. 317–318.

[17] R. Güting, T. d. Ridder, M. Schneider, Implementation of the ROSE Algebra: Efficient Algorithms for Realm-Based Spatial Data Types., in: Proc. of the 4th Intl. Symposium on Large Spatial Databases, 1995, pp. 216–239.

[18] M. Hamilton, F. Salim, E. Cheng, S. Choy, Transafe: A Crowdsourced Mobile Platform for Crime and Safety Perception Management, in: IEEE International Symposium on Technology and Society 2011, IEEE, 2011, pp. 32–37.

[19] K. Hara, S. Azenkot, M. Campbell, C. Bennett, V. Le, S. Pannella, R. Moore, K. Minckler, R. Ng, J. Froehlich, Improving Public Transit Accessibility for Blind Riders by Crowdsourcing Bus Stop Landmark Locations with Google Street View, in: Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility, ACM, 2013.

[20] P. Hart, N. Nilsson, B. Raphael, A Formal Basis For The Heuristic Determination of Minimum Cost Paths, IEEE Transactions on Systems Science and Cybernetics 4 (2) (1968) 100–107.

[21] C. Jason, Z. Yongxin, T. Lei, Where to: Crowd-aided path selection, in: 40th International Conference on Very Large Data Bases (VLDB'2014), ACM, 2014, pp. 2005–2016.

[22] B. Krieg-Brückner, C. Mandel, C. Budelmann, B. Gersdorf, A. Martínez, Indoor and Outdoor Mobility Assistance, in: Ambient Assisted Living, Springer, 2015, pp. 33–52.

[23] J. Li, Map Route Ranking with Weighted Distance using Environmental Factors, arXiv preprint arXiv:1404.0934.

[24] T. Lin, J. Xia, T. Robinson, K. Goulias, R. Church, D. Olaru, J. Tapin, R. Han, Spatial Analysis of Access to And Accessibility Surrounding Train Stations: A Case Study of Accessibility for The Elderly in Perth, Western Australia, Journal of Transport Geography 39 (2014) 111–120.

[25] L. Mandow, J. De La Cruz, A New Approach to Multiobjective A* Search., in: Proceedings of the 19th international joint conference on Artificial intelligence (IJCAI '05), Citeseer, 2005, pp. 218–223.

[26] L. Mandow, J. De La Cruz, Multiobjective A* Search with Consistent Heuristics, Journal of the ACM (JACM) 57 (5).

[27] J. Meurer, M. Stein, D. Randall, M. Rohde, V. Wulf, Social Dependency and Mobile Autonomy: Supporting Older Adults' Mobility with Ridesharing ICT, in: Proceedings of The 32nd Annual ACM Conference on Human Factors in Computing Systems, ACM, 2014, pp. 1923–1932.

[28] T. Miura, K. Yabu, M. Sakajiri, M. Ueda, J. Suzuki, A. Hiyama, M. Hirose, T. Ifukube, Social Platform for Sharing Accessibility Information Among People with Disabilities: Evaluation of a Field Assessment, in: Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility, ACM, 2013.

[29] NASA, Shuttle Radar Topography Mission (SRTM), `http://srtm.usgs.gov/`, Last Accessed: 26-Feb-2015.

[30] M. Nasir, C. Lim, S. Nahavandi, D. Creighton, Prediction of Pedestrians Routes Within a Built Environment in Normal Conditions, Expert Systems with Applications 41 (10) (2014) 4975–4988.

[31] OpenStreetMap, JOSM: OpenStreetMap (OSM) Editor in JAVA, https://josm.openstreetmap.de/, Last Accessed: 26-Feb-2015.

[32] OpenStreetMap, OpenStreetMap (OSM), http://www.openstreetmap.org/, Last Accessed: 26-Feb-2015.

[33] OpenStreetMap, OpenStreetMap, Srtm2Osm, http://wiki.openstreetmap.org/wiki/Srtm2Osm, Last Accessed: 26-Feb-2015.

[34] C. Prandi, P. Salomoni, S. Mirri, mPASS: Integrating People Sensing and Crowdsourcing to Map Urban Accessibility, in: Proceedings of the IEEE International Conference on Consumer Communications and Networking Conference, 2014, pp. 10–13.

[35] D. Quercia, R. Schifanella, L. Aiello, 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, 2014, pp. 116–125.

[36] W. Sasaki, Y. Takama, Walking Route Recommender System Considering SAW Criteria, in: Technologies and Applications of Artificial Intelligence (TAAI), 2013 Conference on, IEEE, 2013, pp. 246–251.

[37] The World Bank, World Bank Group Annual Report 2014,Population Ages 65 And Above (% of Total), http://data.worldbank.org/indicator/SP.POP.65UP.TO.ZS, Last Accessed: 27-Jul-2015.

[38] M. Traunmueller, A. Fatah gen Schieck, Introducing the Space Recommender System: How Crowd-sourced Voting Data Can Enrich Urban Exploration in The Digital Era, in: Proceedings of the 6th International Conference on Communities and Technologies, ACM, 2013, pp. 149–156.

[39] United Nations, United Nations, World Population Ageing 2013, http://www.un.org/en/development/desa/population/publications/pdf/ageing/WorldPopulationAgeing2013.pdf, Last Accessed: 28-Jul-2015.

[40] M. Valtorta, A Result on The Computational Complexity of Heuristic Estimates for the A* Algorithm, Information Sciences 34 (1) (1984) 47–59.

[41] T. Völkel, G. Weber, RouteCheckr: Personalized Multicriteria Routing for Mobility Impaired Pedestrians, in: Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility, ACM, 2008, pp. 185–192.

[42] Walkscore, Walkscore.com, `http://www.walkscore.com`, Last Accessed: 10-Jan-2014.

[43] H. Wang, G. Li, H. Hu, S. Chen, B. Shen, H. Wu, W. Li, K. Tan, R3: A Real-Time Route Recommendation System, Proceedings of the VLDB Endowment 7 (13) (2014) 1549–1552.

[44] R. Wasinger, C. Stahl, A. Krüger, M3I in a Pedestrian Navigation & Exploration System, in: Human-Computer Interaction with Mobile Devices and Services, Springer, 2003, pp. 481–485.

[45] H. Wennberg, A. Ståhl, C. Hydén, Older Pedestrians' Perceptions of The Outdoor Environment in A Year-round Perspective, European Journal of Ageing 6 (4) (2009) 277–290.

[46] M. Whelan, J. Langford, J. Oxley, S. Koppel, J. Charlton, The Elderly and Mobility: A Review of The Literature, Monash University Accident Research Centre Australia, 2006.

[47] S. Wibowo, P. Olszewski, Modeling Walking Accessibility to Public Transport Terminals: Case Study of Singapore Mass Rapid Transit, Journal of the Eastern Asia Society for Transportation Studies 6 (2005) 147–156.