2016 17th IEEE International Conference on Mobile Data Management

Dynamic and Personalized Routing in PreGo

Abdeltawab M. Hendawi[1] Aqeel Rustum[2] Amr A. Ahmadain[1]
Dev Oliver[3] David Hazel[2] Ankur Teredesai[2] Mohamed Ali[2]

[1] *University of Virginia, VA, USA,* {hendawi, aaa9aj}@virginia.edu

[2] University of Washington, WA, USA, {binrusas, dhazel, ankurt, mhali}@uw.edu

[3] (ESRI) Environmental Systems Research Institute, CA, USA, doliver@esri.com

***Abstract*—Existing routing services calculate the best route from source to destination over a road network graph. Most commercial routing services offer the best route in terms of either the shortest travel distance or the shortest travel time (with or without considering current traffic conditions). While travel distance and travel time are crucial route preferences for the commuter, other preferences are equally, or even more, important. Examples of other route preferences include fuel consumption, gas emissions, road safety, points of interest along the route, construction activities, open shops and restaurants. While some route preferences are static (e.g., travel distance and points of interests), other route preferences are dynamic and vary according to the time of the day (e.g., traffic-dependent travel time and the number of open shops/restaurants).Volunteered Geographic information (*VGI*) has been proposed as an approach to collect massive amounts of route information and, more specifically, the time varying parameters.**

This demo presents *PreGo*, a time-dependent multi-preference routing engine. During the demo, audience would interact with the *PreGo* routing engine to (1) find the optimal route w.r.t. the user's personal preferences for a given start time, (2) dynamically obtain the best start time for a trip given a set of preferences, (3) feed the system with *VGI* and examine their effect on the chosen route at real time, and (4) examine the correctness and efficiency of the *PreGo* selected routes compared to routes chosen by other commercial systems.

I. Introduction

People have become highly dependent on routing services in their daily commute, especially with the availability of routing services on a variety of platforms, e.g., web mapping, mobile applications, and in-car GPS navigation devices [3]. The *best* route in terms of travel time or travel distance is the typical answer returned by such routing services. However, this answer does not always comply with commuters' preferences when distance and time are not the only attributes under consideration. For example, a truck driver shipping a heavy load is more interested in a flat non-steep route for safety purposes. A tourist is more interested in travelling on roads that are rich in points of interests (probably in the morning) and full of open shops/restaurants (probably in the evening).

Various routing preferences can be considered in addition to the travel distance and the travel time. To name few other preferences, the number of traffic lights, traffic conditions, road safety (measured by the number of accidents/crimes), road construction activities, the number of points of interest along the route and the number of open shops/restaurants are legitimate routing preferences. UPS has saved millions of gallons of fuel by avoiding left turns [12], which demonstrates the effect of considering attributes other than the distance and time. Moreover, some of these attributes are of dynamic nature and tend to vary at different times of the day. For instance, the number of open services, e.g., gyms and stores, is usually larger during the day time compared to the night time. Also, road congestions tend to increase during the rush hours and disappear during the night. As we mentioned earlier, the tourist's interest in a route varies given the time of day. Therefore, our goal is to leverage routing services to answer time-dependent multi-preference route queries.

In this Demo, we present the *PreGo* routing engine. While several efforts have considered the time dependent dimension of route queries [5] or considered the multi-preference dimension of route queries [4], [10], *PreGo* considers both dimensions at the same time. *PreGo* recommends an optimal path(s) that satisfies multiple preferences and, at the same time, considers the start time of the trip. The routing engine of *PreGo* is tunable in the sense that the discovery of the optimal path is driven by a weighted set of user's preferences. The user is given a set of knobs to adjust (or tune) the weights of routing preferences according to their priorities. *PreGo* is also dynamic over the time dimension. *PreGo* encompasses a time dependent road network graph that is augmented with the Attribute Time Aggregated Graph, (*ATAG*) data structure [9]. This structure is constructed from volunteered geographic information (VGI), e.g., crowd-sourced GPS trajectories, open access maps, and crowd reported events. In *ATAG*, each edge has multiple attributes, each of which corresponds to a user's preference. Each attribute is fine grained to multiple intervals. Each interval corresponds to a time slot of the day. The accompanying Time Parameterized Multi-Preference Shortest Path (*TP_SP*) algorithm processes multi-preference routing queries through a single traverse of the *ATAG* structure. In case user's start time is flexible, *PreGo* recommends the best start time for a route that fits the user's preferences. To further reduce the response latency, *PreGo* employs a *bidirectional TP_SP* algorithm that processes the routing query from the two ends at the same time.

II. Overview of PreGo

The architecture of the *PreGo* system is given in Figure 1. *PreGo's* routing engine consists of three main components:

1) The *ATAG* graph construction and maintenance module through which the underlying road network is initial-

2375-0324/16 $31.00 © 2016 IEEE
DOI 10.1109/MDM.2016.63

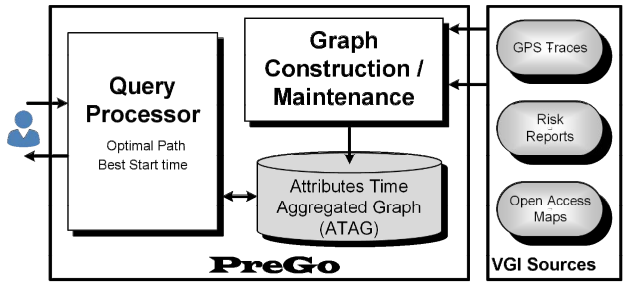


Fig. 1. The System Architecture

ized, built and updated using the collected VGI data (Section II-A).
2) The query processor that is responsible for answering two types of user routing queries (Section II-B).

Two types of route queries are supported within the *PreGo* framework:

1) find the optimal path(s) given a set of preferences for a given start time, and
2) recommend a start time of a trip (for users with flexible start time) such that their preferences are best fulfilled for the resultant route.

PreGo has two interfaces, one interface faces the user's side to accept route queries while the other interface faces the data sources to accept VGI data. These two interfaces can be summarized as follows:

1) The query interface, where a commuter sends a route query, (e.g., source-to-destination route query, a set of preferences and their weights, and an optional trip start time). The system dispatches the *query processing* module to find the optimal path(s) that satisfies the received parameters and returns the resultant path to the user.
2) The data source interface, where *PreGo* calls the *graph construction and maintenance* module upon the arrival of updated VGIs to refresh the edge weights of the *ATAG* structure. Example VGIs that are utilized in the context of *PreGo* are presented in Section II-C.

A. Graph Construction and Maintenance

Basically, each edge in the *ATAG* data structure [9] can have more than one attribute, e.g., travel time, distance, and risk. For each edge, we store multiple weights in different time slots. The graph construction and maintenance modules processes the VGI data collected from various data sources. This component has two major roles. First, it is responsible for building the underlying road network graph *ATAG* via extracting map data such as the graph nodes, edges, and edges' attributes. Second, it initializes and continuously updates the edge weights for each attribute upon the arrival of new VGIs. Edge weight extraction approaches significantly varies according to the attribute's underlying data source. On one side, weight extraction can be as simple as a straightforward interpretation of the input data into an attribute's weight for an edge. On the other side, weight extraction can involve several steps with the employment of different geospatial techniques.

Let's consider few examples over here. Translating the number of car accidents and crimes into a risk weight is computed by normalizing that number of accident/crime reports over the length of the edge. To extract the travel time from a set of GPS tracks, we initially map-match each input GPS track to its corresponding consecutive set of road segments. Several map matching techniques are found in literature. In our work ,we have used the map matching algorithm proposed by Kuien et al. [11]. Then, we apply a sequence of validation rules on each GPS track to filter out anomalies resulted from noisy GPS devices, e.g., maximum speed limit violation. Only valid tracks are considered to compute the travel time weights. Note that since each GPS point in a GPS track is timestamped, travel time weights are time parameterized.

B. Query Processing

The query processing module is in charge of processing users' multi-preference route queries. The *query processor* handles a given query expressed as source-destination pair of nodes, a set of user's preferences and relative preference weights. There are two modes of operations:

Mode 1: Predefined trip start time: If the user provides the trip start time, the *query processing* module directs the search over the *ATAG* for the optimal obtainable path that fulfills the user's routing preferences for that given start time. The *TP_SP* routing algorithm considers the weights of an edge along the route at the time the traveller is expected to hit the beginning of this edge.

Mode 2: Flexible trip start time: If the user has a flexible start time, *PreGo* recommends the best start time at which the optimal possible path(s) that fits all the submitted preferences and weights is guaranteed.

To work under the above modes of operation while preserving efficiency and scalability, *PreGo* is equipped with the Time Parameterized Multi-Preference Shortest Path (*TP_SP*) algorithm that calculates the best route from the *ATAG* structure. Following the Dijkstra's approach [13], via a single traverse of the *ATAG* structure, the *TP_SP* algorithm discovers the optimal path(s) from a source to a destination w.r.t. a subset or all attributes. *TP_SP* utilizes a *prune and wait* approach. $Prune$ stops the traversal of the graph for all branches except the branches that can be optimal for at least one attribute. Hence, the $prune$ strategy ensures the system's efficiency by not draining the computation resources in non-required graph expansion. The $wait$ prevents the *TP_SP* algorithm from declaring the in-hand path (that reached the destination node) as the optimal path until all other branches are either pruned or examined. Hence, $wait$ ensures the optimality of the chosen path and it will not be beaten by any other path in the graph.

In order to further decrease the response time, we introduce the *bidirectional* version of our *TP_SP* algorithm. Bidirectional shortest path algorithms [7] traverse the road network graph from the two ends (source and destination) at the same time to speed up the shortest path discovery. We also introduce

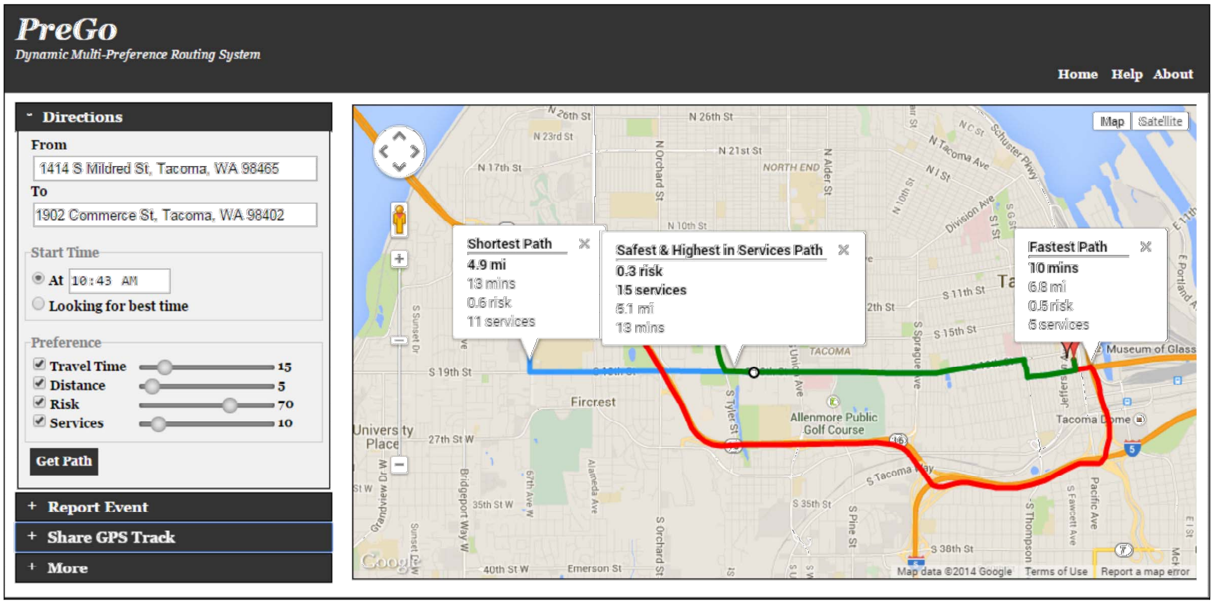


Fig. 2. Obtaining The Multi-preference Routing Results

the best start TP_SP algorithm to address the second type of queries where the trip start time is flexible. For more technical depth of the proposed algorithm, we refer the reader to [8].

C. *Examples of Volunteered Geographic Information*

In this section, we highlight three example Volunteered Geographic Information (VGI) data sources that are utilized within *PreGo*:

(1) *Public GPS Traces.* Our main source of volunteered data is the GPS tracks given by the crowd. Using these GPS tracks, actual travel times on various road segments for a given time of the day can be extracted. Open Street Map (OSM) allows for the download of volunteered GPS traces filtered by areas of interest.

(2) *Open Access Maps.* To obtain the base for the *ATAG* structure, we rely on the available free accessible map resources such as shape files [6], and the *Tareeg* web-service [2]. Using these resources, we are able to extract the road network graph as a set of nodes and edges, and compute the basic weights, e.g., distance. In addition, they give us the ability to extract some indicators about the near by services and points of interests around each edge, e.g., lakes, parks, commercial buildings, schools etc.

(3) *Risk Reports.* The final data source contains data about car accidents and crime linked to their locations and times during the day, e.g., NHTSA [1]. The number of recorded crimes and accidents events around an edge gives an indicator of how risky this edge is.

III. Demo Scenarios

In this section, we briefly describe how the attendees of the conference will be able to interact with the system from different perspectives. The audience interacts with the system and examines its features and capabilities through its front end user interface (Figure 2). The user interface leverages the Google Maps APIs and the D3.js visualization packages. The demo is based on real data for road network graphs obtained from OpenStreeMap with the assistance of the *Tareeg* framework [2]. The remainder of this section describes the propsoed demo scenarios.

A. *Scenario 1: Find Multi-Preference Optimal Path*

As a fundamental scenario of using the *PreGo* system, users can submit multi-preference route queries to obtain the optimal route from a source location to a destination given a set of preference attributes and a specific start time. From the left expandable panel in Figure 2 under ”Directions”, the user provides the source location, which is initially set to user’s current location, and the destination either in a standard address format or in a latitude/longitude format. The user also specifies the trip start time in this scenario. Using multiple knobs, the user specifies his preferences through selecting the attributes to be considered in the routing plan. The attributes are presented with checkboxes through which he is able to specify more than one attribute. Then, each attribute’s weight is decided using a sliding bar. Finally, the user clicks the ”Get Path” button and as a result the system will show the optimal path(s) on the map. An example query result is given in Figure 2, where multiple routes are returned back in response to the query parameters on the left pane. The total cost of each route is reported across multiple preferences (e.g., travel time, travel distance, risk, and services along the route). The users will be able to compare the *PreGo* selected route to routes selected by other commercial routing systems on the same map. In Figure 2, *PreGo* was able to achieve user’s risk preference but was not able to find one route satisfying all preferences. So multiple routes are returned.

B. Scenario 2: Find Best Start Time

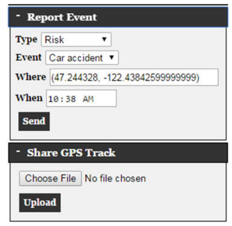
For the same route query (as in the previous scenario), the user chooses to get recommendation from *PreGo* about the best time to start his trip in order to maximize the values of his preference attributes. This feature is beneficial for users with flexible start time. To explore this feature, the user simply checks the ”Looking for best start time” radio button and then clicks the ”Get Path” button. The best-start *TP_SP* algorithm will be fired to compute the best start time for the user's preferences combination. In the best case, one starting time along with its relevant optimal route will be recommended. Nevertheless, it is possible to be recommended with more than one start time, each of which is associated with a different route. This case happens when there is no single optimal path according to the user specifications can be achieved.

C. Scenario 3: Volunteering / Reporting

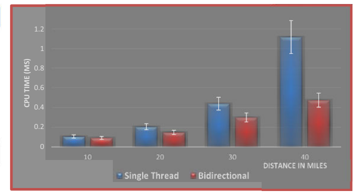
In this scenario, users will be able to contribute to the system's real time data feed by sharing their GPS tracks and other VGIs. The user's collected information updates the edge weights of the *ATAG*. Moreover, users can report events associated with their location and time. Example of such events includes risky events (crimes or accidents), services report (liked restaurant), and disturbances (road construction). The audience will be able to see the *PreGo* selected route in response to the recently reported VGIs, Figure 3(a).

D. Scenario 4: Visualization of System's Internals

In this scenario, attendees learn more about how the *PreGo* system works under the cover. Audience can visually inspect the internal behavior of the system when answering a routing query through a simple animated demonstration. In general, audience can see how the *TP_SP* algorithm is executed on a road network graph augmented by the *ATAG* structure. Users visually see how the *ATAG* data structure is updated in response to an incoming piece of VGI. Users also explore how the dominance relationship among a set of candidate nodes is computed for each graph expansion at each iteration. Moreover, audience will view the animated demonstration of the bidirectional *TP_SP* version. We show how the forward and backward threads can collaboratively work to find an optimal path. This visualization of the system's internals can be reached from the ”More” panel at the bottom of Figure 2. Finally, users will be able to test the overall performance of the *PreGo* system. This is done by generating charts to compare the cost for running the single thread versus the bidirectional versions of the *TP_SP* algorithm, Figure 3(b). Users will have the ability to play with many parameters, e.g., distance between source and destination, time slots of the day, number of preferences, and see the effect on the performance.



(a) VGI Reporting



(b) Performance Charts

Fig. 3. Data Update and Performance Test In PreGo

References

[1] National Highway Traffic Safety Administration. Fatality Analysis Reporting System (FARS). ftp://ftp.nhtsa.dot.gov/FARS/, January 2013.

[2] L. Alarabi, A. Eldawy, R. Alghamdi, and M. F. Mokbel. TAREEQ: A MapReduce-Based Web Service for Extracting Spatial Data from OpenStreetMap. In *SIGMOD*, Utah, USA, June 2014.

[3] AllWebsiteStats. Statisitcs For Websites Usage. http://allwebsitestats.com/, June 2014.

[4] Balteanu Adrian and Jossé Gregor and Schubert Matthias. Mining driving preferences in multi-cost networks. In *SSTD*, pages 74–91, Munich, Germany, August 2013.

[5] George Betsy, Sangho Kim, and Shashi Shekhar. Spatio-temporal network databases and routing algorithms: A summary of results. *SSTD*, 4605:460–477, 2007.

[6] US Census Bureau. TIGER/Line and Shapefiles. http://www.census.gov/geo/maps-data/data/tiger-line.html, October 2013.

[7] Nannicini Giacomo, Delling Daniel, Liberti Leo, and Schultes Dominik. Bidirectional A* search for time-dependent fast paths. *Experimental Algorithms*, pages 334–346, 2008.

[8] Abdeltawab M. Hendawi, Aqeel Rustum, Dev Oliver, David Hazel, Ankur Teredesai, and Mohamed Ali. Multi-preference Time Dependent Routing. In *Technical Report UWT-CDS-TR-2015-03-01, Center for Data Science, Institute of Technology, University of Washington, Tacoma*, Washington, USA, March 2015.

[9] Abdeltawab M. Hendawi, Eugene Sturm, Dev Oliver, and Shashi Shekhar. CrowdPath: a framework for next generation routing services using volunteered geographic information. In *SSTD*, pages 456–461, Munich, Germany, August 2013.

[10] Hans-Peter Kriegel, Matthias Renz, and Matthias Schubert. Route skyline queries: A multi-preference path planning approach. In *ICDE*, pages 261–272, California, USA, March 2010.

[11] Kuien Liu, Yaguang Li, Fengcheng He, Jiajie Xu, and Zhiming Ding. Effective map-matching on the most simplified road network. In *GIS*, pages 609–612, Redondo Beach, CA, USA, November 2012.

[12] JOEL Lovell. Left-hand-turn elimination. http://goo.gl/3bkPb, December 2007.

[13] Dijkstra Edsger W. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.