A dynamic and scalable user-centric route planning algorithm based on Polychromatic Sets theory
Li, Peisong; Wang, Xinheng; Gao, Honghao; Xu, Xiaolong; Iqbal, Muddesar; Dahal, Keshav

Published in:
IEEE Transactions on Intelligent Transportation Systems

DOI:
10.1109/TITS.2021.3085026

Published: 07/06/2021

Document Version
Peer reviewed version

Link to publication on the UWS Academic Portal

Citation for published version (APA):

General rights
Copyright and moral rights for the publications made accessible in the UWS Academic Portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
If you believe that this document breaches copyright please contact pure@uws.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

“© © 2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”
A Dynamic and Scalable User-centric Route Planning Algorithm Based on Polychromatic Sets Theory

Peisong Li, Xinheng Wang, Senior Member, IEEE, Honghao Gao, Senior Member, IEEE, Xiaolong Xu, Muddesar Iqbal, and Keshav Dahal, Senior Member, IEEE

Abstract—Existing navigation services provide route options based on a single metric, such as the shortest path or the shortest time; however, user’s preference is not considered. This results in the planned route not meeting the actual needs of users. In this paper, a personalized route planning algorithm is proposed, which can provide users with a route that meets their requirements. Based on the multiple properties of the road, the Polychromatic Sets (PS) theory is first introduced into route planning. Firstly, we designed a road properties representation scheme based on the PS theory. By this scheme, users’ travel preferences can be quantified into using multiple road properties, and then personalized road property group can be constructed according to these properties. Then, the idea of setting priority for road segments was utilized. Based on user's travel preferences, all the combination schemes of property categories in the property group can be prioritized, so that all the road segments have their own priority under the current travel plan. Finally, based on the priority division of different road segments, an efficient path planning scheme was proposed, in which priority is given to the high-priority road segments in the target direction as a planned route. In addition, the system can constantly obtain real-time road information through mobile terminals, update road properties, and provide other users with more accurate road information and navigation services, so as to avoid crowded road segments without excessively increasing time consumption. Experiment results show that our algorithm can realize personalized route planning services without significantly increasing the travel time and distance.

Index Terms—User-centric, Route planning, Polychromatic Sets Theory.

I. INTRODUCTION

ROUTE planning is one of the most basic functions of navigation systems, which can help travelers determine the best route that meets the requirement [1]. Existing navigation applications such as Google Maps and Waze can provide accurate services with options, for example, multiple routes are provided for selection on Google Maps before departure and accurate arrival time could be assured by Waze owing to the update of real-time traffic conditions. Some other services such as going to the petrol station in the middle of the journey are also considered, which makes the services more personalized.

Normally, these services rely heavily on one or two metrics such as the shortest distance or the shortest travel time. However, users’ demands are diverse. Some drivers might desire to the destination by using the least time, so highways or street roads with less traffic are selected, but they have to drive very fast or make a lot of right or left turns. Others might prefer to drive an easy way, such as driving on a more straightforward road despite of the heavy traffic or not. Upon existing navigation systems, it is obvious that recommending the same path for all the users based on predefined metric(s) cannot meet the individual needs for different users. In this case, the recommended route is usually not selected [2]. However, there are no other choices available in existing navigation applications. In addition, according to the study in [3], the routes that are usually selected by the drivers are significantly longer than their shortest route when the route selection criterion is either ‘travel length’ or ‘travel time’. Therefore, authors declared that “...algorithms based on shortest route to represent routes may not capture real-world route choice decisions.”, which means that routes computed based on a single-criterion shortest route algorithm like Dijkstra may not be desirable for a driver.

Recently, in order to solve these problems, multi-criterion route planning has become a research hot-spot for public travel information services [4]–[6]. These methods consider multiple criteria and weight each criterion to make an optimal new choice. However, in reality, it is still a one-criterion navigation method. It cannot offer multiple choices to the users, particularly to those during the navigation. In addition, these methods don’t have the scalability in terms of metrics. The number of metrics is limited. When a metric needs to be considered into the service, a new algorithm has to be designed to meet the demand.

We argue the reason behind this is because there lacks of
a systematic mathematical model to describe the conditions or properties of the traffic and then make recommendations based on these conditions. In this paper, we have proposed a new property description method based on Polychromatic Sets (PS) Theory, where unlimited number of metrics can be described and used for forming new routes. This makes the system scalable, and the selection of routes can be based on user’s preferences, rather than one or two physical parameters.

In this new proposed method, firstly, various conditions of the road and traffic, where route property is used to refer to these conditions, can be described comprehensively and combinations of route properties can be established according to different travel plans. Then, according to the specific travel plan, priority is assigned to the properties. In the route planning stage, the high-priority road segment is selected first, and then low-priority road segments are selected next, if no high-priority segment is available. In addition, priority should be given to the road segment in the direction to the destination, which can effectively reduce the search area and ensure that the planned path is close to the shortest path. In this way the execution efficiency of the algorithm could be improved.

The contributions of this paper are summarised in the following three aspects:

(1) A road property description scheme based on PS theory is provided. This scheme is capable of including main properties of road networks in route planning.

(2) A user-centric road priority setting method is proposed. Under different and personalized travel plans, the priority of each road segment can be dynamically adjusted and selected to plan the route.

(3) An efficient route planning process is proposed. When carrying out route planning, the road segments with high priority and in the direction to the destination will be chosen first, which can ensure that the planned route not only meets the user’s preferences and the specific travel plan, but also effectively reduce the search area and further saving time used for route planning.

The remainder of this paper is organized as follows: In Section II, related work is reviewed. It also demonstrates how our work is different from existing research. In Section III, a framework of a property description method and route planning process is proposed. The experimental setup and the performance evaluation results are presented in Section IV to verify the effectiveness of the proposed approach. Finally, this paper is concluded and the future research directions are discussed in Section V.

II. RELATED WORK

Traditional travel route planning provides a route between an origin and a destination according to a certain metric, such as the shortest distance or the shortest travelling time. Dijkstra’s algorithm [7] is the most well-known method for solving the shortest path problem. However, Dijkstra’s algorithm needs to be modified for some real travel applications, due to its inefficiency. To address this problem, Delling et al. [8] proposed a customizable route planning for real-time queries in terms of arbitrary metrics, including avoidance of U-turns and/or left turns.

Traditional travel route planning schemes have other two limitations. First, some factors, such as road safety and traffic jams, are usually latent in travelling and difficult to access; hence, traditional route planning only considers limited factors and cannot make a comprehensive consideration of travel. Second, traditional route planning algorithms are not personalized because they do not consider user’s travel preferences and provide only generic recommendations to the public.

User-centric is an increasingly concerned direction, and the future development will definitely be based on user experience. Several previous studies have investigated user-centric route planning algorithms. Table I shows the details of the studies. In [9], users are allowed to decide the bus route. A geospatial decision support tool for bus travel planning based on design and modeling considerations was developed and implemented in this paper, which is more individual-centric than existing approaches. The distinctive characteristic of the proposed tool is the incorporation of bus riders’ cognition, information processing, and decision-making process during their search for an appropriate bus transit travel plan. Authors in [10] proposed a personalized route planning query method. They developed new preprocessing schemes that allow for real-time personalized route planning in huge road networks while keeping the memory footprint of the preprocessed data and subsequent queries small. Researchers in [11] defined users’ travel behaviors from their historical GPS trajectories and proposed two personalized travel route recommendation methods – collaborative travel route recommendation (CTRR) and an extended version of CTRR (CTRR+). In [12], a probabilistic driving preference model in multi-attribute dynamic road traffic network was proposed, which can be used to solve personalized Skyline path.

Nadi and Delavar [13] proposed a model for the personalized multi-criterion route planning that fused a pair-wise comparison method and quantifier-guided ordered weighted averaging (OWA) aggregation operators to determine an impedance value for each link of the urban transportation network. Sadeghi Niaraki and Kim [14] integrated an Analytic Hierarchical Process (AHP) with a generic ontology-based architecture to identify and define a class of road segment criteria for personalized route planning. In the two mentioned studies, the personalized multi-criterion route planning problem has been simplified to a single objective optimization routing problem. When they consider a certain index, the weight of this index increases, but they ignore other indices and reduce the weight of other indices. As a result, the planned route does not meet the expectation of users and an “abnormal” route is provided.

Some researchers are focusing on how to provide route navigation options based on the road information and
driver’s profiles. Marina et al. [15] focused on the personalized routes problem and then presented a specific case of designing accessible and green pedestrian routes. Similarly, a route planning framework was proposed in [16], which utilizes the in-vehicle and smartphone sensors to build a crowdsensed database on segment quality and the driver’s personalized behavioral information in different driving environments. A two-phase approach for personalized trip planning was proposed in [17]. In the route search phase, TRIPPLANNER works interactively with users to generate candidate routes with specified venues; in the route augmentation phase, TRIPPLANNER applies heuristic algorithms to add user’s preferred venues iteratively to the candidate routes, with the objective of maximizing the route score while satisfying both the venue visiting time and total travel time constraints.

Some researchers focus on the impact of real-time road traffic information on route planning. The dynamic and time-dependent route planning problem was studied in [18], which takes both prediction (based on historical data) and live traffic into account. Traffic jams at different times of the day is also considered. A novel fuel-efficient path-planning framework called Green Planner was presented in [19], which contains two phases. In the first phase, a Personalized Fuel Consumption Model (PFCM) for each driver was built, based on the individual driving behaviors and the physical features such as traffic lights, stop signs, and road network topology along the routes. In the second phase, the real-time traffic information was collected via a mobile crowdsensing manner. This method could estimate and compare the fuel cost among different routes for a given driver, and recommend him/her with the most fuel-efficient one.

However, above researchers didn’t consider the multiproperty of the dynamic and complex road networks in reality. Some existing algorithms can analyze users’ travel preferences according to their historical trajectories, but they cannot deal with users’ new travel preferences. At the same time, the existing user-centric route planning algorithm can satisfy the user’s single travel preference, but cannot solve the multi-objective travel requirements. In addition, towards the road property, the existing algorithms only focus on the traffic information, but not consider the road inherent properties sufficiently. Moreover, the calculation process of the algorithm is complex, so it is difficult to widely implement and deploy. Therefore, this paper aims to solve the problems that the planned route is not consistent with the user’s current travel preference, so as to make the planned route more personalized. The route planning algorithm based on PS theory proposed in this paper fully considers the multiple properties of the road, and the travel plan can include multiple independent criteria, such as time, type of vehicle, road choice, and traffic conditions. This not only meets the user’s multiple needs for the travel route, but also ensures that the planned route is as close as possible to the shortest route between two points.

### III. POLychromatic Sets Theory and Algorithm Design

A pioneering work in applying PS theory in computer networks was introduced in [20]. In that paper, the theory of the polychromatic sets and the way how the network properties were described by the theory was fully explained. Following the same principle of applying PS theory in computer networks, the properties of road networks can be described by this theory. An example of the road properties is listed in Table II.

#### A. Road property description based on PS

There are three important definitions in the PS based road property description: Property, Property group and Property categories.

1) Property: Each road in the network has a variety of properties, and each property contains one or more property categories. Examples of road properties are shown in Table II:

<table>
<thead>
<tr>
<th>Property</th>
<th>Property group</th>
<th>Property category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Road Type</td>
<td>Road Type Group</td>
<td>Road Type Category</td>
</tr>
<tr>
<td>Traffic Flow</td>
<td>Traffic Flow Group</td>
<td>Traffic Flow Category</td>
</tr>
<tr>
<td>Road Surface</td>
<td>Road Surface Group</td>
<td>Road Surface Category</td>
</tr>
</tbody>
</table>

Set $A$ is defined as a set of all the road segments in the road network:

$$A = \{a_1, a_2, \ldots, a_i, \ldots, a_p\}$$

where $a_i$ represents the $i^{th}$ road segment, $p$ is the total number of segments. If the property is defined as $f$, then all the properties in a road network can be defined as a set $F$, as shown in Eq. (2), where $q$ represents the total number of
properties. Then the network’s properties can be calculated by a Cartesian product of $A$ and $F$, shown in Eq. (3).

$$F = \{f_1, f_2, \cdots, f_j, \cdots, f_q\}$$

$$c_{ij} = [A \times F] = \begin{bmatrix} a_1 \cdots a_q \end{bmatrix} = \begin{bmatrix} c_{11} & \cdots & c_{1j} & \cdots & c_{1q} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ c_{i1} & \cdots & c_{ij} & \cdots & c_{iq} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ c_{p1} & \cdots & c_{pj} & \cdots & c_{pq} \end{bmatrix} a_p$$

$[A \times F]$ is a two-dimensional matrix, the size of it is $[p \times q]$. $f_j$ represents the $j$th property of $q$ properties. If one road segment $a_i$ is coloured as $f_j$, which means that $a_i$ has the property $f_j$, then the element $c_{ij}$ in the above matrix is set as 1, otherwise, it is 0.

2) Property group: According to various travel plans, road properties can be divided into different groups. Different groups represent different property combination schemes. The classification examples are as follows:

(1) When the planned travel is “The shortest time”, road properties in this group include speed level and congestion level:

$$G(T_1) = \{f_2, f_3, \cdots\}$$

where $G(T_1)$ represents the group of properties that need to be considered under the travel plan of The shortest time $T_1$. Therefore when planning the route, only these two properties need to be considered.

(2) When the planned travel is “Avoid crowded urban streets”, road properties in this group include road type and congestion level:

$$G(T_2) = \{f_1, f_3, \cdots\}$$

(3) When the planned travel is “Avoid the toll”, the road properties in this group include the toll status:

$$G(T_3) = \{f_5, \cdots\}$$

(4) When the planned travel is “The least transfer”, the road properties in this group include road functions:

$$G(T_4) = \{f_4, \cdots\}$$

In conclusion, according to different travel plans, only the corresponding road properties need to be considered when the routes are being planned, the other properties may be ignored.

In addition, the setting of the property group should take into account not only the travel plan selected by the user, but also the user’s choice of vehicle. For example, when the user chooses to travel on foot, the corresponding property group should include the property of road function.

3) Property categories: As shown in Table II, each segment has one or more property categories. The Table II can be represented by the matrix in Eq. (8):

$$P(a_i) = \begin{bmatrix} p_{1-1} & p_{1-2} & \cdots & p_{1-k} \\ p_{2-1} & p_{2-2} & \cdots & p_{2-k} \\ \vdots & \cdots & \cdots & \vdots \\ p_{j-1} & p_{j-2} & \cdots & p_{j-k} \end{bmatrix} f_j$$

where $P(a_i)$ represents the category matrix of one road segment $a_i$, $p_{j-k}$ represents the $k$th category of property $f_j$. The category that the road segment has is set as 1, otherwise, it is 0.

For example, the category matrix of one road segment $a_m$ is:

$$P(a_m) = \begin{bmatrix} 1 & 0 & 0 & 0 & f_1 \\ 0 & 1 & 0 & 0 & f_2 \\ 1 & 0 & 0 & 0 & f_3 \\ 1 & 0 & 0 & 0 & f_4 \\ 1 & 0 & 0 & 0 & f_5 \end{bmatrix}$$

According to Table II, this matrix means that the road segment $a_m$ has the following property categories: National road, Expressway, Smooth, Only small cars allowed, and Toll on the road.

What’s more, the combination of categories in different properties can form a variety of combination schemes. For example, as shown in Table II, there are 4 property categories in the property $f_1$ (Road type) and 3 property categories in the property $f_2$ (Speed level):

$$f_1 = \{p_{1-1}, p_{1-2}, p_{1-3}, p_{1-4}\}$$

$$f_2 = \{p_{2-1}, p_{2-2}, p_{2-3}\}$$

As shown in Eq. (10) and (11), property $f_1$ (Road type) has 4 property categories: National road ($p_{1-1}$), Provincial road ($p_{1-2}$), Country road ($p_{1-3}$) and Street ($p_{1-4}$); property $f_2$ (Speed level) has 3 property categories: Highway ($p_{2-1}$), Expressway ($p_{2-2}$) and Ordinary ($p_{2-3}$); Both can be combined to form 12 different combination schemes of property categories, and each road segment satisfies one of the combination schemes, where $C_i$ is the $i$th combination scheme.
Meanwhile, scalability is an important feature of the proposed route planning system. With the change of real-time road information, the road segments’ properties, as shown in Table II, can be expanded to provide more property group. In this way we can construct more property category combination schemes and satisfy more personalized multi-criterion travel preferences for users. Since the route planning can be based on combination of properties, new routing metrics can be always created and added to the system to provide services. This makes the system dynamic as well.

In addition, the real-time traffic information is collected via the mobile devices during the navigation. Then the current status of the related segment in map database can be adjusted via Message Queuing Telemetry Transport service (MQTT). MQTT has become the de facto protocol for transferring data between mobile devices and the cloud database.

B. User-centric road priority setting

Existing route planning algorithms only consider how to make the planned route shortest in the process of route planning, but ignore the user’s personalized requirements for travel. Therefore, we should consider how to make the planned route satisfy the user’s preferences. So, we propose to set priority for road segments according to users’ different travel requirements.

As shown in Fig. 1, each combination scheme represents a different priority. When planning the route, the high-priority segments are selected first and if there are no high-priority segments, then the low-priority segments will be selected. By default, the first and second priority segments are regarded as high-priority, and the rest are low-priority segments. The details of each priority are explained below:

(1) The segments that fully meeting the requirements of travel plan (one of the property-category combination schemes) can be called as first-priority segments. As the user’s travel preference can be quantified as multiple criteria based on the PS theory, the property-category combination scheme with all these criteria is set as the first-priority combination. In this way the segment that with this combination is set as first-priority segment.

(2) In general, the actual traffic conditions are more complicated, and, sometimes, it is difficult to find segments that fully meet all the criteria. Considering the connectivity of the navigation road and the integrity of the route, the property-category combination scheme with most of these criteria is set as the second-priority combination.

(3) Similarly, if there is no second-priority segment, then the requirement is lowered again and the segment that meets the criteria at this stage is called third-priority segment.

For example, when the travel plan is the shortest time, the road property group corresponding to this plan includes speed level and congestion condition. The road segments that have property categories of highway and smooth are set as the first priority. Those having property categories of highway and mild congestion are set as the second priority, and those having property categories of highway and moderate congestion are set as third priority.

In conclusion, under different travel plans, before route planning, the system will form corresponding priority groups. A priority represents a combination scheme of property categories. After the user selects a travel plan, the system divides the priority of each property category combination scheme in this travel plan and establishes the corresponding priority classification standard. According to the priority, all road segments in the road network are divided into different priority sets:

\[
G(C_1) = \{a_1, \ldots, a_i\} \\
G(C_2) = \{a_{i+1}, \ldots, a_{i+j}\} \\
\ldots \\
G(C_{12}) = \{a_{i+j+\ldots+1}, \ldots, a_p\}
\]

where \(G\) represents the group of segments.

C. Route planning process

In the process of route planning, the widely implemented algorithms, such as Dijkstra, traverse all the segments connected with the current vertex. Although this approach can guarantee to find the shortest path, it requires a large amount of computation and is less efficient. Therefore, this paper proposed an optimized progressive route planning process.

Based on the PS-based road property description and the setting of priority in section III-A and III-B, we proposed an efficient route planning process. This process includes 8 steps, mainly comprised of building the map database, segment selection and starting navigation stages.

(1) Building map database
Step1: Pre-built map database, comprised of two parts:

One is the road properties database: it comprises of the property information of each road segment, such as congestion status (smooth, mild congestion, moderate congestion, severe congestion), road types (highway, expressway, ordinary road), length of the road segment, road speed limit, etc. For a segment, the property it owns is marked as 1; otherwise, it is marked as 0.

Another is the priority database of road segment selection. For different navigation choices, road segment has different priority classification criteria. Each segment is assigned with different priority according to the corresponding division criteria. First-priority is the first, followed by the second-priority, third-priority, and so on.

(2) Segment selection
Step2: A navigation path will include multiple segments. First, an empty set of segments will be established. The optimal path is formed by connecting segments end-to-end.

Step3: Based on the connection between starting point and destination, we traverse all the road segments connected to the starting point in a 90° area, which is shown in Fig. 2(a). By default, the first and second priority segments are high-priority segments, and the rest are low-priority segments.

Step4: When more than one road segment is connected with the starting point, according to the travel plan set by the user, the road segment meeting the first priority will be selected. If there is no road segment meeting the first priority, then the road segment meeting the second priority will be selected.

Step5: If the road segment meeting the requirements (user’s preferences) cannot be found within the 90° area, then the search range is expanded to 180° and Step 4 is performed, where the highest priority road segment is selected in this area, which is shown in Fig. 2(b). Also, if the road segment meeting the requirement is still not found, the search range is further expanded to 270° and repeat Step 4, which is shown in Fig. 2(c). If still no road segment can’t be found to meet the requirements, it will narrow down the search range to the original 90° area and lower the priority again until a road segment is selected in this area. Then the selected segment is added to the segment set in this navigation. If the road segment meeting the requirements is not found at last, the user is prompted to have no planned route, please re-select.

Step6: Take the end point of the selected segment as the new starting point, perform Steps 3-5 again, and start to select the next segment until the destination set by the user is reached, which is shown in Fig. 2(d).

Step7: At the same time, after each selection of a road segment, a loop avoidance mechanism will be ignited to avoid falling into an endless loop.

As shown in Fig. 3a, when the segment d is connected to the route, e is the best segment to be connected. Therefore, segment e in the segment set. Then the segment f and segment g are chosen into the segment set. There forms a loop as d → e → f → g → d. In the loop avoiding mechanism, when a road segment is selected, it should check the route segment set to find whether this segment is already in the set or not. If this segment is already in the set, it means that a loop has been created. So, the next segment is labelled as a loop segment and the route information on this segment is to be deleted. Then another segment is to be found. As shown in Fig. 3b, when a route d → e → f → g → d has formed a loop, segment e is labelled and the route d → e is deleted. Then perform Step 6 to choose another segment, segment h is to be found. At last a final route is discovered and shown in Fig. 3c.

Fig. 2. Traverse all the road segments

Step7: A navigation path will include multiple segments. First, an empty set of segments will be established. The optimal path is formed by connecting segments end-to-end.

Step5: If the road segment meeting the requirements (user’s preferences) cannot be found within the 90° area, then the search range is expanded to 180° and Step 4 is performed, where the highest priority road segment is selected in this area, which is shown in Fig. 2(b). Also, if the road segment meeting the requirement is still not found, the search range is further expanded to 270° and repeat Step 4, which is shown in Fig. 2(c). If still no road segment can’t be found to meet the requirements, it will narrow down the search range to the original 90° area and lower the priority again until a road segment is selected in this area. Then the selected segment is added to the segment set in this navigation. If the road segment meeting the requirements is not found at last, the user is prompted to have no planned route, please re-select.

Step6: Take the end point of the selected segment as the new starting point, perform Steps 3-5 again, and start to select the next segment until the destination set by the user is reached, which is shown in Fig. 2(d).

As shown in Fig. 3a, when the segment d is connected to the route, e is the best segment to be connected. Therefore, segment e in the segment set. Then the segment f and segment g are chosen into the segment set. There forms a loop as d → e → f → g → d. In the loop avoiding mechanism, when a road segment is selected, it should check the route segment set to find whether this segment is already in the set or not. If this segment is already in the set, it means that a loop has been created. So, the next segment is labelled as a loop segment and the route information on this segment is to be deleted. Then another segment is to be found. As shown in Fig. 3b, when a route d → e → f → g → d has formed a loop, segment e is labelled and the route d → e is deleted. Then perform Step 6 to choose another segment, segment h is to be found. At last a final route is discovered and shown in Fig. 3c.

Fig. 3. Illustration of loop avoidance
navigation services for users on their mobile terminal. In the navigation process, the system can obtain the real-time traffic information via the mobile crowdsensing manner anytime. The collected information could be uploaded to the background database via Message Queue service (MQ) and the priority of segments in database could be adjusted.

The pseudocode of user-centric route planning algorithm based on PS theory is shown in Algorithm 1:

Algorithm 1 PS based route planning algorithm
1: \( s \): Current segment during the route planning process
2: \( \text{origin} \): The origin of the planned route
3: \( \text{destination} \): The destination of the planned route
4: User decide the travel preference \( P \)
5: Construct the road property group \( G(f_1, f_2, \cdots) \)
6: Set the priority for each segment
7: Establish an empty segment set \( S \)
8: \( s \leftarrow \text{origin} \)
9: while \( s \) doesn’t reach to \( \text{destination} \) do
10: if have required segment \( i \) in 90° area then
11: \( s \leftarrow i \)
12: Add \( s \) into \( S \)
13: else
14: if have required segment \( i \) in 180° area then
15: \( s \leftarrow i \)
16: Add \( s \) into \( S \)
17: else
18: if have segment \( i \) in 270° area then
19: \( s \leftarrow i \)
20: Add \( s \) into \( S \)
21: else
22: Lower the requirement and traverse required segment \( i \) in 90° area
23: \( s \leftarrow i \)
24: Add \( s \) into \( S \)
25: end if
26: end if
27: end if
28: end while
29: Start navigation based on the \( S \)

As shown in algorithm 1, the time complexity of Function \( while \) is \( O(1) \), so the overall time complexity is:

\[
T(n) = O(n)
\]  

(13)

The space complexity is:

\[
S(n) = O(1)
\]  

(14)

IV. ALGORITHM IMPLEMENTATION AND EVALUATION

A. Experimental Setup

1) Evaluation environment and target: In order to model and implement the proposed algorithm, experiments were conducted by using MATLAB R2016a on a PC with a 64-bit Windows 10 operating system, a 16 GB of RAM, and a 1.19-GHz-Core(TM) i5-based processor.

In order to verify the effectiveness of our proposed user-centric and multi-criterion route planning algorithm, we have compared with the following two widely used route planning algorithms, Greedy Forwarding geographical Routing (GRF) and the improved Dijkstra algorithm on Graphhopper.

(1) GRF is carried out by default in a greedy way: The current node always forwards the message to the neighbor node closest to the target [21]. A simple and effective iterated greedy algorithm is presented in [22].

(2) Graphhopper is an open source routing library and server written in Java and provides a web interface called GraphHopper Maps, which has been widely used [23]. Graphhopper can be configured to use different algorithms, such as Dijkstra, A*, and its bidirectional versions. In order to make the routing fast enough for long paths (continental size) and avoid heuristic approaches, Graphhopper uses contraction hierarchies by default.

The Graphhopper routing service based on Dijkstra algorithm was used for the comparison experiments in this paper. [24] proposed a multi-criterion decision making of optimal route based on Dijkstra algorithm.

2) Data Preparation: In our experiments, the proposed algorithm is tested and validated through a case study of a real driving scenario in London, the UK. We take the road network in an urban area in London as an example, where the actual road network is shown in Fig.4:

![Fig. 4. A regional road network](image)

In order to compare the performance of proposed algorithm with the planned route on Graphhopper, we simplify and show the main roads of actual city road network using Matlab, which is shown in Fig.5:
There are totally 216 road segments in this area. The map database in this experiment is the 216 segments with their properties.

3) Evaluation Metrics: Degree of Demand Satisfaction (DDS) is the major metric used to evaluate the proposed algorithm. In addition, Time and Distance are also two important evaluation metrics to evaluate the performance.

(1) DDS is one of the evaluation indices used to evaluate that, to what extend, the planned route meets the travel plan set by the user, which is defined as:

\[
DDS = \frac{S_f + S_c}{S_T}
\]  

where \(S_f\) represents the first-priority segments in the planned route, \(S_c\) represents the second-priority segments in the planned route, both are combined as the high-priority segments which highly meeting the travel plan. \(S_T\) represents the total distance of the planned route.

(2) The shortest time and the shortest distance are always the users’ default travel preferences. In this case, we need to provide users with a planned route as short as possible in terms of time and distance while ensuring users’ DDS.

B. Experimental steps and computation

1) Experimental steps: First, the starting point, destination point and user’s travel plan are set. These parameters are the input of the experiment.

Second, the system constructs the segment property group and the corresponding priority setting according to the travel plan.

Finally, plan the route from the starting point based on the proposed algorithm, Dijkstra and GFR methods, respectively.

2) The computation of metrics: First of all, we need to calculate the DDS. The calculation method is shown in Eq. (15).

For the calculation of travel distance, the distance of each road section in the database is consistent with the real-world actual distance.

In order to calculate the travel time conveniently, we set the speed of the road segment reasonably according to different road type and different congestion level. Each type of road, such as city road or street, has its usual speed limit. The speed of different segments is set in a hierarchical way. Specific calculation method is shown in Eq. (16):

\[
T = \frac{d_1}{s_1} + \frac{d_2}{s_2} + \ldots + \frac{d_n}{s_n}
\]

where \(T\) represents the travel time, \(d_n\) represents the distance of the \(n\)th segment and \(s_n\) represents the speed limit of the \(n\)th segment.

C. Performance Evaluation

1) Comparison of travel plan "Smooth highway first": First of all, when the user’s travel plan is “smooth highway first”, the road properties that need to be considered are road congestion and road type. The road property diagram is shown in Fig.6, \(S\) represents the starting point and \(D\) represents the destination.

Road priorities are set in Table III.

The route planned by Graphhopper using Dijkstra algorithm is shown in Fig.7.

The planned routes based on PS theory, GFR algorithm, and Dijkstra algorithm are shown in Fig.8, which are demonstrated by bold lines. Among them, as shown in Fig.8, the planned route based on Dijkstra algorithm is the simplified.
TABLE III
SEGMENT PRIORITY SETTING (PLAN 1)

<table>
<thead>
<tr>
<th>Smooth</th>
<th>Mild congestion</th>
<th>Moderate congestion</th>
<th>Severe congestion</th>
<th>Highway</th>
<th>Expressway</th>
<th>Ordinary road</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>12</td>
</tr>
</tbody>
</table>

Fig. 8. Planned routes based on PS, GFR, and Dijkstra algorithm (plan 1)

Fig. 9. Comparison of three algorithms (plan 1)

The proportion of different types of segments included in the three planned routes lead to the difference in user’s satisfaction. As can be seen from Fig.8, the planned route based on the PS algorithm proposed in this paper is mainly composed of smooth highway. Only when there is no highway at the starting point, the system will choose the smooth expressway. While, the planned route based on the GFR algorithm and Dijkstra algorithm does not consider the requirements of users, so the planned travel route includes ordinary roads and congested segments.

2) Comparison of travel plan “Avoid the speed limited road”: Then, we test the second plan. When the user’s travel plan is “avoid speed limited road”, we need to consider whether the road segment is speed limit or not. The road property diagram is shown in Fig.10, where $S$ represents the starting point and the $D$ represents the destination.
Road priorities are set as Table IV:

<table>
<thead>
<tr>
<th>Speed limit road</th>
<th>No speed limit road</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

The route planned by Graphhopper using Dijkstra algorithm is shown in the Fig.11:

The planned routes based on PS theory, GFR algorithm, and Dijkstra algorithm are shown in the Fig.12, which are demonstrated by dotted lines for the route designed by PS, solid lines for GFR algorithm and dashed lines for Dijkstra, respectively. The planned route based on Dijkstra algorithm is the simplified result of the route planned by Graphhopper.

The performance of the planned routes obtained by the three algorithms is shown in Fig. 13. As can be seen from Fig.13, the planned route based on PS algorithm does not include the speed limit segment, so the degree of user’s demand satisfaction has reached 100%. At the same time, although the consideration of road speed limit is added in the route planning, the planned route based on PS algorithm does not significantly increase the road distance, and the travel time is shorter than the other two routes.

The type of segments included in the three planned routes leads to the difference in user’s satisfaction. As can be seen from Fig.12, considering the requirement of “avoid the speed limit road”, the route planned based on PS algorithm can avoid the speed limit segment very well. However, the other two algorithms do not consider the road property of ”speed limit or not” in the route planning, so the planned route contains the speed-limited segments, which will bring a negative impact on user’s travel experience.

V. CONCLUSION

In this paper, a user-centric route planning algorithm is proposed. Based on the PS theory, this algorithm focuses
on considering the multi-property information of the road when carrying out route planning, and provides different shortest travel routes for users according to the personalized differences on travel demand. At the same time, the system can update the road property database in real-time according to the feedback of users, and then provide users with more accurate route planning services. Experimental results show that the planned route based on PS algorithm not only meets the requirements of users, but also is very close to the shortest route between two points.

This algorithm can improve the single-criterion route recommendation pattern which only considers time or distance and provide users with diversified routes. Meanwhile, periodic updating of road properties can improve the real-time and accuracy of route planning.

In the future, we will consider more properties to design and optimize the routing algorithm, so as to make better travel planning under the modern complex road network.

REFERENCES


