



(51) International Patent Classification:
G05D 1/00 (2006.01)

(21) International Application Number:
PCT/US2020/017923

(22) International Filing Date:
12 February 2020 (12.02.2020)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/804,616 12 February 2019 (12.02.2019) US

(71) Applicant: **ARIZONA BOARD OF REGENTS ON BEHALF OF THE UNIVERSITY OF ARIZONA** [US/US];
The University of Arizona, Tech Launch Arizona, 220 W. 6th Street, 4th Floor, Tucson, Arizona 85701 (US).

(72) Inventor: **GE, Yong**; c/o ARIZONA BOARD OF REGENTS ON BEHALF OF THE UNIVERSITY OF ARIZONA, 220 W. 6th Street, 4th Floor, Tucson, Arizona 85701 (US).

(74) Agent: **FONTES, Timothy** et al.; POLSINELLI PC, 105 West Vandalia Street, Suite 400, Edwardsville, Illinois 62025 (US).

(81) Designated States (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA,

(54) Title: SYSTEMS AND METHODS FOR ROUTE COMPUTING FOR DESTINATION-ORIENTED NAVIGATION

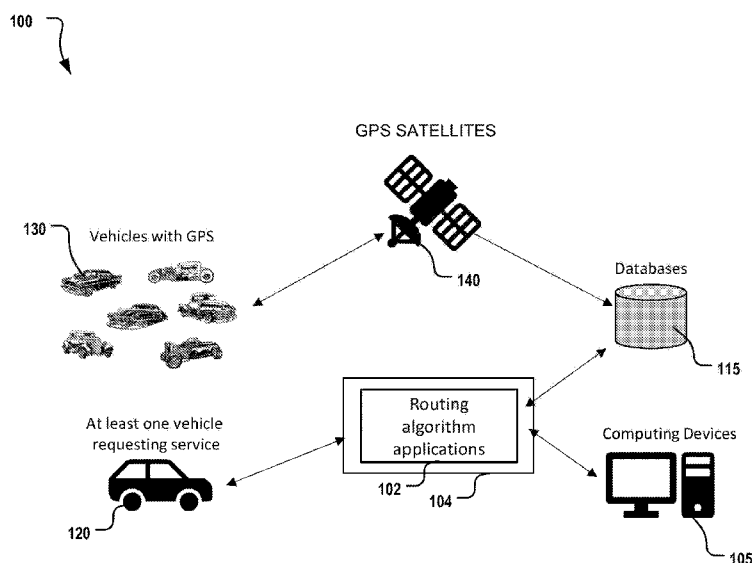


FIG. 8

(57) Abstract: Large amounts of mobility data and the ability to track computing devices associated with moving people or objects is leveraged for the disclosed development of advanced mobile recommendations, which are essentially to recommend a sequence of locations to an individual user on the move. As described herein, data associated with a particular case of mobile recommendations is analyzed, and route recommendations are provided to devices associated with drivers, by utilizing vehicle GPS data. Specifically, a new Route Recommendation is formulated by way of a Relaxed Assumptions (RR-RA) solution, the goal of which is to recommend a sequence of locations to a driver based on a current location. A computer-executable algorithm is implemented to efficiently generate such a route recommendation. Further, a destination-oriented route recommendation (DORR) problem may be solved by implementing a computer-executable algorithm for solving the DORR problem.

SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR,
TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

SYSTEMS AND METHODS FOR ROUTE COMPUTING FOR DESTINATION-ORIENTED NAVIGATION

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This is a non-provisional application that claims benefit to U.S. provisional patent application serial number 62/804,616 filed on February 12, 2019, which is herein incorporated by reference in its entirety.

FIELD

[0002] The present disclosure generally relates to route computing; and more particularly, to route computing for optimal destination-oriented navigation.

BACKGROUND

[0003] For an increasing number of drivers, weather unpredictability, topography, and traffic congestion can make digital navigation planning a daunting, and sometimes impossible task. Individual drivers and businesses may rely upon or otherwise desire access to effective driving instructions to meet timelines, goals, and other needs. Unfortunately, technical challenges with route computing may involve complications at least with (1) planning and accounting for real-time computations and (2) the translation of location-based information.

[0004] It is with these observations in mind, among others, that various aspects of the present disclosure were conceived and developed.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] FIG. 1 is a node graph depicting a sample route.

[0006] FIG. 2 is a sample of a tree structure utilized for route computing as described herein.

[0007] FIG. 3A and FIG. 3B are graphs illustrating pick-up events and direction summarization, respectively, as described herein.

[0008] FIG. 4A and FIG. 4B are graphs depicting illustration examples for destination-oriented route recommendations as described herein.

[0009] FIG. 5 is a sample of a tree structure for route recommendation as described herein.

[0010] FIG. 6 is a process flow diagram illustrating one possible algorithm for destination-oriented route recommendation (DORR), as described herein.

[0011] FIG. 7 is a graph illustrating the visualization of the effectiveness of the recommendations as described.

[0012] FIG. 8 is a system diagram illustrating possible devices and tangible components related to the present novel disclosure.

[0013] FIG. 9 is a process flow for solving the RR-RA problem defined herein.

[0014] FIG. 10 is a process flow for solving the DO-RR problem defined herein.

[0015] FIG. 11 is a simplified block diagram of a computing device that may be configured as a special purpose computer using the algorithm and features described herein.

[0016] Corresponding reference characters indicate corresponding elements among the view of the drawings. The headings used in the figures do not limit the scope of the claims.

DETAILED DESCRIPTION

[0017] The present disclosure relates to computer-implemented technology for generating an advanced mobile recommendation, and the present disclosure is responsive to various technical problems in the mobile recommendation and route computing space. In some embodiments, aspects of the present disclosure may take the form of a computing device configured for executing functionality defined by at least one novel algorithm which configures the computing device to output an advanced mobile recommendation that accommodates an optimal navigational experience. In general, large amounts of mobility data and the tracking of movement and location of devices may be accessed by the computing device and leveraged as inputs or considerations for the algorithm, and the mobile recommendation may take the form of a recommended sequence of locations to a driver based on the driver's current location. The computing device may further measure pick-up location probability and may generate an optimal route to inform the driver of the likelihood of connecting with a passenger device based on current location and route.

[0018] In some embodiments, the novel technology of the present disclosure involves the formulation and implementation/application of a novel Route Recommendation with Relaxed Assumptions (RR-RA) solution. In particular, a (RR-RA) problem may be solved by the solution described herein according to instructions executed via the computing device in the course of generating the advanced mobile recommendation/s, which provide a proactive way for taxi drivers to better search for potential passengers. In some embodiments, the RR-RA solution may consider a general mobile sequential recommendation (MSR) problem with fixed length constraint. Solving the general MSR problem may involve two assumptions. The first assumption is to fix the number of pick-up locations of a candidate route and only search for an optimal route from among these ones. The second assumption is that the travel distance, when a driver cannot pick up any passenger after passing the last pick-up location of candidate route, is assumed to be a large constant and that this constant is the same for all candidate routes. However, these assumptions to the MSR problem are problematic in real world applications. Accordingly, the novel (RR-RA) solution of the present disclosure may involve relaxing both assumptions as described herein.

[0019] More specifically, the (RR-RA) solution may be formulated or otherwise defined, as described herein, and this solution may contribute to the advanced recommendation of a sequence of pick up locations for an empty cab based on potential pick up locations mined from historical taxi GPS data and current location of the empty cab (i.e., no passenger). This recommendation may be in the form of driving routes which makes the problem particularly challenging. Accordingly, a novel approach is developed to solve this problem. The goal of this approach is to generate driving routes that could minimize the travel distance before picking up passenger(s). While the intelligent route recommendation solutions described herein each provide or otherwise define a new way for taxi drivers to improve their business performance, conventional methods for resolving these problems involves a computing process that is very computationally expensive. It is challenging to make recommendations practical for real taxi businesses because the results need to be produced online in a timely fashion to meet the real-time requests of drivers. Therefore, in some embodiments, a pruning-based algorithm executable by a computing device or processor is described for addressing various problem definitions and is described for efficiently searching optimal routes to recommend.

[0020] Further, an important destination constraint is identified that taxi drivers may frequently face, and a complimentary destination-oriented route recommendation (DORR) problem is further defined and discussed. The DORR problem may be crucial for any practically successful solution to the driver's route recommendation problem. As discussed herein, in some cases, in solving the DORR problem, the solution for the RR-RA problem cannot be very practical for real application because drivers frequently run into the DORR problem. Although the DORR problem shares similar inputs as the RR-RA problem, there is a destination constraint that has to be considered for route recommendation. Thus, the present disclosure further includes a new recommendation approach solution for solving the DORR problem. It further calls for and defines efficient algorithms to make the online recommendation practical - a dedicated and efficient algorithm is discussed which is implemented to search for an optimal route in this destination-constrained scenario. By solving the DORR problem according to the DORR solution described herein, the RR-RA solution is strengthened which makes the approaches of the present disclosure more practical because the combined solutions for both RR-RA and DORR problems provide a comprehensive way for route recommendations to taxi drivers.

[0021] In addition, the efficiency of the methods/algorithms for addressing the complimentary RR-RA and DORR problems are evaluated using both real-world and synthetic data. The experimental results demonstrate that the proposed methods could significantly save computing time for online recommendation compared with baseline methods. Furthermore, it is important to validate how effective the recommended routes are. While the ideal way for validating this is to conduct a field study (a.k.a., A/B testing) via collaborating with taxi and ride sharing companies, it is very difficult to provide such validations for several reasons. Therefore, the effectiveness of the described recommendations is evaluated offline by using a traditional machine learning paradigm. The empirical results demonstrate that the described novel route recommendations could greatly help drivers search for potential passengers, and reduce driving time without passengers by around 18%.

[0022] Overall, the contributions of the present disclosure may be summarized as follows.

- A technically challenging and practically important RR-RA problem is presented and defined, and an advanced method for solving this problem is developed and described.
- A DORR problem is formulated, and an advanced solution is developed to solve it. Solving both of the RR-RA and DORR problems accommodates the generation of a complete package solution for route recommendations to taxi drivers.
- Empirical tests are conducted to evaluate the performance of the described methods with large-scale real-world taxi GPS data, and simulations are used to demonstrate the practical value of the route recommendations.
- The subject novel approaches may also be useful for solving other transportation problems such as parking lot search and routing for carpool as discussed herein. The techniques developed for RR-RA and DORR problems will open a new range of methods in the recommendation field and the developed algorithms will lay a foundation for studying other mobile recommendation problems.

Introduction and Relevant Technical Problems:

Mobile Recommendations:

[0023] The aim of mobile recommendations is to provide recommendations to users on the move, such as route recommendations to taxi drivers, places of interest (POI) recommendations to tourists, and the like. Mobile recommendations constitute a growing and important sub-area in the field of recommender systems due to the following factors: (a) the existence of a rich and diverse class of applications requiring mobile recommendations; (b) the establishment and maturity of mobile infrastructure, including proliferation and wide adoption of various mobile devices that generate rich and useful data of high quality; and (c) because many mobile recommendation problems are still underexplored, and development of novel methods is required to solve these problems. All of the aforementioned three factors constitute a perfect storm and make the emerging area of mobile recommendations an important and rich topic of research.

[0024] Historically, the area of mobile recommendations started with the work on mobile guides that come in various “forms and shapes,” and provide an extensive categorization of the whole ranges of these guides according to their connectivity to the Internet, being indoor (e.g., museums) or outdoor, etc. Then, the work evolved to use this classification to develop a set of mobile guide design principles that can be used by application developers. The mobile guide systems typically use temporal, location-based and other types of contextual information (such as the current mood and interests of a tourist, weather information, etc.) obtained either from the user or extracted from the environment in order to query or search a certain repository of resources (such as, restaurants, museums and shows) and present the best matching resources (e.g., nearby restaurants that are currently open) to the user.

[0025] One of the early examples of research on mobile tourist guides is the Cyberguide project, which develops several tour guide prototypes for different hand-held platforms. COMPASS uses a similar type of the context-driven querying and searching approach as the Cyberguide system. Other examples of context-aware tourist guide systems include the GUIDE system, the INTRIGUE system, and the MyMap system.

[0026] An alternative approach to mobile recommendations is based on the contextual preference elicitation and discovery and it is being surveyed. As an example of this approach, the UbiquiTO system implementing a mobile tourist guide provides intelligent adaptation of recommendations not only based on the specific location-based, temporal and other contextual information, but it also uses various rule-based and fuzzy set techniques to adapt the application content based on the user preferences and interests. Other examples of contextual preference elicitation and estimation approaches to mobile recommendations are also available. Further, different combinations of these approaches could be achieved via a specially developed integration module that also features a graphical interface for mobile users. Usefulness of such mobile recommendations has been demonstrated in a previous study where 155,000 customers of a mobile portal revealed that the use of personalized recommendations instead of non-personalized ones leads to a significant increase in viewed and sold items in different navigational situations and may positively affect the underlying business.

Analytics of Taxi Trajectory Data:

[0027] The data mining research literature has witnessed a growing interest in taxi trajectory data. The general goal of these works is to understand and discover behaviors and patterns that trajectories generate. For instance, taxi global positioning system (GPS) data has been studied to analyze the impact of information on driver behavior and economic outcome, and find heterogeneity in individual learning behavior and driving decisions having economic impact. Other studies have focused on how a taxi driver gathers and learns information in an uncertain environment through the use of his social network with massive taxi GPS datasets. Other studies analyze the decision process from a social perspective with trajectory data, and the taxi driving fraud behavior (i.e., overcharging passengers by taking unnecessary detours) and propose detection methods for such fraud with taxi GPS data. Although these studies reveal useful information and knowledge about taxi driver behaviors, they fail to provide solutions for navigating taxi drivers to achieve optimal routes that increase efficiency and earn more business.

Analytics of Taxi Trajectory Data:

[0028] Some research has been done which involves proposed methods for recommendations directed to taxi services. However, these methods work based on certain assumptions as described herein. Thus, they are unsuitable or otherwise incapable of being used for solving the novel RR-RA or DORR problems formulated.

Vehicle Routing:

[0029] The present disclosure also considers the vehicle routing problem (VRP). The general goal of the VRP is to search optimal routes for vehicles to traverse in order to reach a set of customers, and the traveling cannot be completed until all customers are reached. The problems addressed by the present novel disclosure (discussion of RR-RA/DORR problems) are different from VRP in that it routes a driver for searching for the next passenger, and it is not needed to reach all potential locations and the travel will stop once the next passenger is reached. Due to such inherent differences between VRP and the disclosed RR-RA/DORR problems, those solutions for the VRP could not be used for solving the RR-RA and DORR problems described herein, and the present novel disclosure therefore describes novel solutions/methods to solve RR-RA and DORR problems. In addition, while the VRP is often resolved offline, the RR-RA and DORR problems

need to be solved online by using the real-time location information of drivers. The efficiency of solution for RR-RA and DORR problems is much more important than that for VRP problem, and the algorithms and methods described herein aim to solve both problems.

Proposed Technical Solutions – Route Recommendation with Relaxed

Assumptions:

[0030] The present novel disclosure considers the concepts and addresses the various technical problems above. For example, the present novel concept is associated with a particular application of the diverse and multifaceted field of providing mobile recommendations and focuses on route computing and destination-oriented navigation recommendations for drivers of different types of vehicles, including taxis, private cars, and tour buses. Such a transportation application is important because it can be leveraged to: (a) recommend more efficient routes to drivers, especially the less experienced ones, and help them save time, money and possibly increase vehicle occupancy and utilization rates, and (b) help economize fuel consumption which will result in energy savings and produce less traffic and pollution.

[0031] While the present disclosure is not limited to any particular application or end user for the described route recommendations, it is believed that the features described herein may be advantageous for ride sharing and especially taxi dispatch and driving. For example, taxis provide mobility in urban areas and play an important role in today's public transportation. However, current taxi services are still facing several practical challenges. First, although today's online taxi transportation companies (such as Uber, DiDi, Lyft) provide support to connect drivers with passengers, their solutions mainly work as a request-based passive dispatch. These current solutions do not pay much attention to the potential passengers who may need services in near future and could not provide good assistance for empty cabs to proactively search for future passengers. Second, most empty cab drivers plan their routes for searching for passengers based on their own experience when there is no service request available, which could lead to low occupancy rate. Effective and efficient recommendations of driving routes to drivers could increase this rate, and further lead to increased efficiency and effectiveness. To address these problems, conventional methods may involve physical sensing

techniques to sense nearby potential passengers and send the passenger location information to taxi cabs on which a physical receiver is installed. Besides the privacy and cost issues, these methods could not capture passengers who do not use their applications and cannot be detected.

Route Recommendation with Relaxed Assumptions (RR-RA):

[0032] Further detail regarding the RR-RA problem shall now be described. Unlike the problems analyzed in other studies, the RR-RA problem relaxes two important assumptions to make the problem more practical for real applications.

Problem Statement:

[0033] Given an empty cab's location $PoCab$, there are a set of N potential pick-up locations denoted as $C = \{C_1, C_2, \dots, C_N\}$ nearby $PoCab$. For each pick-up location, there is estimated pick-up probability, denoted as P_i . Let us use P to denote the set of probabilities $P = \{P_i\}(i = 1, \dots, N)$, where $P_i(i = 1, \dots, N)$ is assumed to be independent to each other. The objective of the RR-RA problem is to search the optimal route that has minimum expected driving distance (EDD) before picking up passenger(s) and includes less than L pick-up locations. Each candidate route is essentially a sequence of potential pick-up locations. We assume that all potential pick-up locations in each candidate route are different. Let us use $\vec{R}$ to denote one candidate route, and use $D(\vec{R})$ to represent the expected driving distance before a pick-up event while a driver is taking a candidate route $\vec{R}$. Based on these notations, the RR-RA problem is formally stated as follows.

The RR-RA Problem

Given: An empty cab's location $PoCab$, a set of N potential pick-up points denoted as $C = \{C_i\}(i = 1, \dots, N)$ nearby $PoCab$, and a set of pick-up probability at these N locations denoted as $P = \{P_i\}(i = 1, \dots, N)$.

Objective: Recommending an optimal route $\vec{R}$ for the empty cab, which minimizes expected driving distance $D(\vec{R})$ before pick-up event and includes less than L pick-up locations ($L \leq N$).

[0034] Compared with the MSR problem that only searches optimal routes with fixed L pick-up locations, we relax this constraint in the subject RR-RA problem. Such a relaxed assumption brings two consequences. First, we make the

recommendation problem more practical because taxi drivers may want to explore driving routes with different numbers of pick-up locations, rather than a fixed number of ones. The recommendation results will better meet the practical needs of taxi drivers than those of MSR problem. Second, the search space of the RR-RA problem is much larger than that of the MSR problem, which leads to more computational complexity. Other assumptions may also be relaxed, and a dedicated method may be developed to solve RR-RA problem as described herein.

[0035] There are two challenges to solving the RR-RA problem. The first is how to obtain potential pick-up locations and pick-up probability given an empty cab requesting recommendation service. Second, is how to timely search the optimal route from all potential candidates to meet the real-time request of route recommendation. In fact, if the computational cost for one candidate route is $\text{Cox}(\vec{R}) = 1$, the computational complexity of RR-RC problem is $\mathcal{O}(N!)$. In the following two subsections, we introduce possible solutions for both challenges.

[0036] *Generating Pick-up Location and Probability:*

[0037] In this section, we introduce how to generate potential pickup locations (i.e., C) and estimate pick-up probability at each location (i.e., P) when the time and location of a recommendation request by one empty cab is given.

[0038] Generating Pick-up Location. Given the location $PoCab$ and time t (e.g., 2PM on Monday) of one empty cab that requests the recommendation of driving route, we will extract a set of raw pick-up points from historical taxi GPS data. Specifically, as long as one pick-up event happens within the time range $[t - \alpha, t + \alpha]$ on the same week day (e.g., Monday) and the corresponding pick-up point locates within circle range that has center as $PoCab$ and radius as T_1 , we will extract this pick-up point. After getting all these raw pickup points, we will group them into different clusters based on the driving distance calculated via Google Map APIs or otherwise. Finally, the centroids of these clusters are considered as potential pick-up locations that will be used for generating recommendations.

[0039] Estimating Pick-up Probability. For each candidate pick-up location (i.e., the centroid of pick-up location cluster), we will estimate the probability of pick-up event around time t (e.g., 2PM on Monday) based on historical data. The idea is to measure how frequent pick-up event happens when cabs travel across a pick-up location cluster C_i around time t on the same week day. Specifically, we first

count the number of empty cabs that enter the spatial coverage of cluster C_i within the time range $[t - \alpha, t + \alpha]$ on the same week day (e.g., Monday) from historical data, and denote it as $\#T$. Among these $\#T$ empty cabs, we count the number of pick-up events $\#P$ that happen within the spatial coverage of cluster C_i among the same time range. Finally, the probability of pick-up event at C_i around time t is

estimated as $P_i^t = \frac{\#P}{\#T}$.

Method for Computing Optimal Route:

[0040] First let us introduce how we define $D(\vec{R})$ for a candidate route $\vec{R}$. To simplify the introduction, let us consider an example of candidate route $\vec{R} = PoCab \rightarrow C_1 \rightarrow C_2 \rightarrow C_3$ as shown in FIG. 1. When a driver takes this route, he will have probability P_i ($1 \leq i \leq 3$) of picking up passenger(s) at each location C_i ($1 \leq i \leq 3$). The driving distance before picking up passenger could be D_1 with probability P_1 , $D_1 + D_2$ with probability $\bar{P}_1 P_2$, or $D_1 + D_2 + D_3$ with probability $\bar{P}_1 \bar{P}_2 P_3$, where $\bar{P}_i$ is defined as $\bar{P}_i = 1 - P_i$. In addition, the probability of picking up no passenger by taking this route is $\bar{P}_1 \bar{P}_2 \bar{P}_3$. We denote the driving distance beyond the last pick-up location of candidate route $\vec{R}$ as $\tilde{D}_3$. We assume $\tilde{D}_3$ is a very big constant and it is the same at the last pick-up location of any candidate route. Based on this assumption, we can develop/configure one or more efficient algorithms for solving the MSR problem. However, this assumption barely holds in reality. First, as $\tilde{D}_i$ denotes the distance that a taxi driver needs to drive to find passenger from location C_i , it could be different at different pick-up location C_i . Second, the value of $\tilde{D}_i$ may not be big at some locations such as those around crowded POIs (e.g., station, hotel). Therefore, we relax this assumption to make the problem more practical. Specifically, we leverage historical taxi GPS data to estimate $\tilde{D}_i$ for each pick-up location C_i offline. Instead of being a large constant, the estimated value of $\tilde{D}_i$ ($1 \leq i \leq N$) may vary at different locations and around different times. Given the above statement, we derive the expected driving distance (EDD) $D(\vec{R})$ before pick-up event while a driver taking route $R = PoCab \rightarrow C_1 \rightarrow C_2 \rightarrow C_3$ as:

$$D(\vec{R}) = D_1 P_1 + (D_1 + D_2) \tilde{P}_1 P_2 + (D_1 + D_2 + D_3) \tilde{P}_1 \tilde{P}_2 P_3 + \tilde{D}_3 \tilde{P}_1 \tilde{P}_2 \tilde{P}_3. \quad (1)$$

[0041] To search for an optimal route for a taxi driver, a straight-forward method needs to compute $D(\vec{R})$ for each possible candidate route with less than L pick-up locations, and then pick the best one $\vec{R}$ that has minimum $D(\vec{R})$. This will involve a lot of computation. In the following, we introduce important monotone property of $D(\vec{R})$, which allows us to prune many candidate routes without computing $D(\vec{R})$. Such pruning can be done offline before the real-time request of route recommendation occurs, and thus it can save a lot of time for generating a real-time recommendation.

[0042] To introduce the monotone property, let us consider two general candidate $\vec{R}^1 = PoCab \rightarrow C_1 \dots \rightarrow C_i$ and $\vec{R}^2 = PoCab \rightarrow C_1 \dots \rightarrow C_i \rightarrow C_{i+1}$. The only difference of $\vec{R}^2$ than $\vec{R}^1$ is that one more pick-up location C_{i+1} is appended. $D(\vec{R}^1)$ and $D(\vec{R}^2)$ can be derived as follows.

$$D(\vec{R}^1) = D_1 P_1 + (D_1 + D_2) \tilde{P}_1 P_2 + \dots + \sum_{n=1}^i D_n P_n \prod_{m=1}^{n-1} \tilde{P}_m + \tilde{D}_i \prod_{n=1}^i \tilde{P}_n. \quad (2)$$

$$D(\vec{R}^2) = D_1 P_1 + (D_1 + D_2) \tilde{P}_1 P_2 + \dots + \sum_{n=1}^i D_n P_n \prod_{m=1}^{n-1} \tilde{P}_m + \sum_{n=1}^{i+1} D_n P_{n+1} \prod_{m=1}^n \tilde{P}_m + \tilde{D}_{i+1} \prod_{n=1}^{i+1} \tilde{P}_n. \quad (3)$$

We then derive the difference between $D(\vec{R}^2)$ and $D(\vec{R}^1)$ as follows.

$$\begin{aligned} \Delta D &= D(\vec{R}^2) - D(\vec{R}^1) \\ &= \prod_{n=1}^i \tilde{P}_n \left\{ P_{i+1} \sum_{n=1}^{i+1} D_n + \tilde{P}_{i+1} \tilde{D}_{i+1} - \tilde{D}_i \right\} \\ &= \prod_{n=1}^i \tilde{P}_n \left\{ P_{i+1} \left(\sum_{n=1}^{i+1} D_n - \tilde{D}_{i+1} \right) + (\tilde{D}_{i+1} - \tilde{D}_i) \right\} \end{aligned} \quad (4)$$

[0043] From Equation 4, we could conclude the following two monotone properties: (a) if $\sum_{ii=1}^{i+1} D_{ii} > \tilde{D}_{i+1}$ and $\tilde{D}_{i+1} > \tilde{D}_i$, ΔD should be bigger than 0, i.e., $D(\overline{R^2}) > D(\overline{R^1})$; and (b) if $\sum_{ii=1}^{i+1} D_{ii} < \tilde{D}_{i+1}$ and $\tilde{D}_{i+1} < \tilde{D}_i$, ΔD should be less than 0, i.e., $D(\overline{R^2}) < D(\overline{R^1})$. For the convenience of presentation, let us denote these two properties as $PR(a)$ and $PR(b)$. In other words, when we append one more pick-up location to a candidate route, we may conclude that the EDD value will increase if the condition of $PR(a)$ is satisfied, or will decrease if the condition of $PR(b)$ is satisfied.

[0044] Next, we introduce how we use these two properties ($PR(a)$ and $PR(b)$) to prune candidate routes in advance without computing EDD function. To make the introduction easy to follow, let us consider a set of 4 pick-up locations and specify $L = 3$. There are totally 24 possible candidate routes with less than 3 pick-up locations. In FIG.2, we show partial possible candidate routes with a tree structure. Let us consider route $PoCab \rightarrow C_1$. As there is only one pick-up location, there is no need to apply property $PR(a)$ and $PR(b)$. Next let us consider all routes that include one more appended pickup location to route $PoCab \rightarrow C_1$, which include $PoCab \rightarrow C_1 \rightarrow C_2$, $PoCab \rightarrow C_1 \rightarrow C_3$, and $PoCab \rightarrow C_1 \rightarrow C_4$. We could check the conditions of property $PR(a)$ and $PR(b)$ for each of these three. Take route $PoCab \rightarrow C_1 \rightarrow C_2$ as an example, if $D_1 + D_2 > \tilde{D}_2$ and $\tilde{D}_2 > \tilde{D}_1$, we can conclude EDD of route $PoCab \rightarrow C_1 \rightarrow C_2$ will increase compared with that of route $PoCab \rightarrow C_1$. Let us use I to tag this conclusion at pick-up location C_2 . This tag indicates that route $PoCab \rightarrow C_1 \rightarrow C_2$ cannot be the optimal route because route $PoCab \rightarrow C_1$ is better than it, thus we do not need to compute EDD for the route $PoCab \rightarrow C_1 \rightarrow C_2$. Similarly, if the condition of $PR(b)$ (i.e., $D_1 + D_2 < \tilde{D}_2$ and $\tilde{D}_2 < \tilde{D}_1$) is satisfied, we use D to tag the conclusion at C_2 , i.e., EDD of route $PoCab \rightarrow C_1 \rightarrow C_2$ will decrease compared with that of route $PoCab \rightarrow C_1$. The tag D indicates that route $PoCab \rightarrow C_1$ cannot be optimal because route $PoCab \rightarrow C_1 \rightarrow C_2$ is better than it, thus we do not need to compute EDD for route $PoCab \rightarrow C_1$.

[0045] Next, given *PoCab*, N potential pick-up locations and L , we will first generate all possible candidate routes and store them with a tree structure as shown in FIG. 2. We will then conduct the same process over all candidate routes (from ones with 1 pick-up locations to those with L ones) on the tree structure. Finally we will be able to tag partial pick-up locations with either I or D . Please note that many pick-up locations may not have any tag if conditions of both $PR(a)$ and $PR(b)$ are neither satisfied, and that we do not need to tag any pick-up location which appears as the first one of candidate route. In FIG. 2, we show some hypothetic tags for partial pick-up locations of shown candidate routes. Based on these tags, we can conduct the following two pruning strategies: (a) if I is tagged to one pick-up location C_i , the candidate route ending with C_i can be pruned, i.e., removing from consideration for an optimal route; (b) if D is tagged to one pick-up location C_i , the candidate route ending with the parent node (i.e., pick-up location) of C_i can be pruned. As many candidate routes can be pruned based on both strategies and pruning takes less time, we will be able to save a lot of computation of EDD for them.

[0046] In fact, we can build the tree structure as shown in FIG. 2 offline before we know the position of an empty cab requesting a recommendation service. We could obtain N potential pick-up locations at different times in a region offline based on historical data. We then build the tree structure (excluding *PoCab*) with these locations and user-specified L . Based on the tree structure, we could conduct the same pruning strategies as described above. Once *PoCab* of one empty cab is given, we can search the optimal one from remaining candidate routes. Since many candidate routes have been pruned offline, we will only need to compute EDD for the remaining candidate routes and pick the optimal one with minimum EDD. Thus, we will save a lot of computation for the online recommendation.

Destination-Oriented Route Recommendations (DORR):

[0047] In this section, we introduce a destination-oriented route recommendation (DORR) problem. Compared to the RR-RA problem, there is a destination-driven direction constraint in the DORR problem, which makes the methods for RR-RA problem not work.

Problem Statement:

[0048] In the real-world taxi business, a driver of an empty cab may want to quickly go to an area from a far-away location and he may also expect the opportunity to pick up a customer on the way back. A taxi driver may have such a preference under several scenarios. For instance, a driver in New York City (NYC) may typically operate in the downtown area; after delivering a passenger to a hotel in the uptown area, he wants to quickly drive back to the downtown area and expects to earn some business on the way back. In fact, in big urban areas, many taxi drivers usually have their typical or familiar operation areas and they tend to drive back after they happen to travel to a far-away location. As another scenario, many taxi drivers in metropolitan areas live in places which are often far away from their operation areas, and they want to pick up customers on the way to their operation area when they leave home and start their work daily. Since a particular driver usually has a destination constraint during his driving, we name this practical challenge as a destination-orient route recommendation (DORR) problem.

[0049] We observe two main constraints imbedded in DORR problem. First, we have to take into account the specific destination where a driver wants to go back for addressing this DORR problem. Second, in the destination-constrained scenario, drivers have more concern about the driving direction which may be detoured after picking up a customer, because they want to go back to their operation areas within some reasonable time. The request-based passive dispatch solution provided by today's online taxi transportation companies (e.g., Uber and DiDi) could not work in this scenario because they have not considered both constraints at all. To meet this specific need, we developed a dedicated method for addressing this DORR problem. In the following, let us first state this DORR problem in a formal way.

[0050] Let us denote one location as $\mathcal{E}$ which represents an area where the driver D_i usually operates, and another location as $\mathcal{S}$ which is a place that is far away from $\mathcal{E}$, and the driver D_i wants to go back to $\mathcal{E}$ from $\mathcal{S}$. From historical data, we could obtain a set of N potential pick-up points, $\mathcal{C} = \{C_1, C_2, \dots, C_N\}$, along the way from $\mathcal{S}$ to $\mathcal{E}$, and the estimated pick-up probability P associated with each pickup point $\mathbb{P} = \{P_1, P_2, \dots, P_N\}$ to denote the probability set. In addition to the probability, we could also get the direction information associated with each pick-up point. For each

pick-up point $\mathcal{C}_i$, we first gather all historical pickup events $T^{\mathcal{C}_i}$ at this pick-up point. We denote all pick-up events associated with the pick-up point set $\mathcal{C}$ as $T^{\mathcal{C}}$. Each pick-up event is actually a location trace, which indicates that a taxi driver delivers passenger(s) from this pick-up point to another location. We represent each pick-up event as a directed arrow starting from this pick-up point as shown in FIG. 3A. This arrow reflects the direction toward a particular location where the driver drops off passenger(s). For each pick-up point $\mathcal{C}_i$, we summarize the potential direction of pick-up events happening at $\mathcal{C}_i$ into 8 bins as shown in FIG. 3B.

[0051] Comparing with the direct travel from $\mathcal{S}$ to $\mathcal{E}$, it will certainly increase the travel distance for driver D_i to intentionally pass several pick-up points on the way from $\mathcal{S}$ to $\mathcal{E}$. We can calculate this certainly increased travel distance based on the passed pick-up points. We denote the certainly increased travel distance as *certain cost*. Furthermore, when driver D_i picks up passenger(s) at a pick-up point on his way from $\mathcal{S}$ to $\mathcal{E}$, this pick-up event may cause the driver to travel a greater distance. We denote such potentially increased travel distance as *uncertain cost*. Given all pick-up events associated with one pick-up point, we are able to estimate this *uncertain cost*. We first compute the exactly-increased travel distance for each historical pick-up event, such as a pick-up event at A in FIG. 4(a). Then the average of exactly-increased travel distance over all pick-up events at one pick-up point can serve as an estimation of such *uncertain cost*. Thus, there are such two aspects of cost associated with each pick-up point. Of course, the driver has some chance to earn business by passing pick-up points. Given all pick-up points and their associated information, we can generate many candidate routes consisting of different pick-up points from $\mathcal{S}$ to $\mathcal{E}$. For example, in FIG. 4(a), we can have sequences like $\mathcal{S} \rightarrow A \rightarrow B \rightarrow \mathcal{E}$ or $\mathcal{S} \rightarrow A \rightarrow C \rightarrow D \rightarrow \mathcal{E}$. We represent the set of all possible routes as $\mathbb{R}$, each of which leads to different *uncertain* and *certain* cost, and *business success*.

[0052] Our goal is to search an optimal sequence of pick-up points for a taxi driver in order to maximize his business success with constraints on certain and uncertain cost. Let us use a function $\mathcal{G}$ to assess the two aspects of cost for each candidate route. For example, for a particular sequence $R_i (R_i \in \mathbb{R})$, we denote the

function as $\mathcal{G}(R_i, S, E, P_{R_i}, T^{R_i})$, where T^{R_i} and P_{R_i} represent the associated pick-up events and probability information of R_i . In other words, for a specific sequence, the function $\mathcal{G}$ depends on source and end nodes, pick-up points along the sequence, and all associated information of these pick-up points. The business success of the taxi driver in this destination-constrained scenario may be measured in different ways. In the following paragraphs, we will introduce two ways for measuring the business success and the specific formula of function $\mathcal{G}$. Given the above statement, we can formally define the DORR problem as follows.

The DORR Problem

Given: An original location $\mathcal{S}$ and a destination location $\mathcal{E}$, a pick-up point set $\mathbb{C}$ and associated probability set $\mathbb{P}$, the associated pick-up event set $\mathbb{T}_{\mathbb{C}}$, a taxi driver who locates at $\mathcal{S}$ and wants to go back to $\mathcal{E}$ quickly, a cost constraint TC that is essentially to limit the potentially travel distance.

Objective: Recommending an optimal sequence of pick-up points for the taxi driver, which could maximize the business success under the given cost constraint.

[0053] Similar to the RR-RA problem, we assume that a driver would not visit a pick up location more than once in the DORR problem. In other words, there is no duplicate pick-up location in each candidate route.

[0054] To solve this problem, we have two challenges. The first one is how to obtain reliable pick-up locations and associated information from the historical taxi GPS data. The second challenge is how to efficiently search optimal driving routes to meet the real-time recommendation requests of drivers. Suppose the computational cost for one candidate route is $\text{Cost}(\mathcal{G}) = 1$. The computational complexity of this DORR problem is $\mathcal{O}(N!)$. In the following sections, we introduce the proposed method for solving these two challenges.

[0055] Generating Pick-up Location and Probability. Given the original location $\mathcal{S}$ and destination location $\mathcal{E}$ for one empty cab that requests recommendation, we will obtain the potential pick-up points and pick-up probabilities in an online fashion.

[0056] Generating Pick-up Location. Given the $\mathcal{S}$ and $\mathcal{E}$, we will extract a set of raw pick-up points from historical taxi GPS data. We apply two criteria for selecting the raw pick-up points. First, we locate the time t when an empty cab requests recommendation, and only keep raw pick-up points that happen within the time range $[t - \alpha, t + \alpha]$ on the same week day. Second, we denote the latitude and longitude of $\mathcal{S}$ and $\mathcal{E}$ as $(lat_{\mathcal{S}}, lng_{\mathcal{S}})$ and $(lat_{\mathcal{E}}, lng_{\mathcal{E}})$, respectively, and only keep raw pick-up points that locate within the circle that has the center as $\left(\frac{(lat_{\mathcal{S}} + lat_{\mathcal{E}})}{2}, \frac{(lng_{\mathcal{S}} + lng_{\mathcal{E}})}{2}\right)$ and the radius as T_2 . After selecting these raw pick-up points, we will group them into different clusters based on the driving distance calculated via Google Map APIs. Centroids of these clusters are considered as potential pick-up points that will be used for generating recommendations.

[0057] Estimating Pick-up Probability. For each candidate pick-up point generated on the above, we estimate the pickup probability at time t in the same way as described above with relation to the RR-RA problem. Specifically, we first obtain the spatial coverage of each cluster. We then count the number of empty cabs which pass the spatial area of a cluster C_i within the time range $[t - \alpha, t + \alpha]$ on the same week day, and denote it as $\#T$. Among these $\#T$ empty cabs, we count the number of pickup events $\#P$ that happen within the same time range. Finally, the

probability of pick-up event at C_i around time t is estimated as $P_i = \frac{\#P}{\#T}$.

Estimating Cost Function and Business Success

[0058] In this section, we introduce the formulation of function $\mathcal{G}$ and the measurements of business success.

[0059] Suppose there are totally K historical pick up events happening at C_i . Let us denote each of them as $T_k^{C_i} (1 \leq k \leq K)$. Also let us denote the corresponding drop off location of each pick up event as $Dr_k^{C_i} (1 \leq k \leq K)$. If a driver only visits one pick up location C_i while traveling from $\mathcal{S}$ to $\mathcal{E}$, he may pick up a customer at C_i and deliver this passenger to different locations. Our idea is to leverage the average of historical K pick-up events to estimate the travel distance.

Thus driver's driving route could be $\mathcal{S} \rightarrow C_i \rightarrow Dr_k^{C_i} \rightarrow \mathcal{E} (1 \leq k \leq K)$. We represent the

driving distance of each route as $D_k(C_i)$ ($1 \leq k \leq K$). The average of K routes is $D(C_i) = \sum_{k=1}^K D_k(C_i)/K$, which is used to estimate the driver's potential driving distance if he picks up a customer at C_i . In fact, as the travel distance $D_{S \rightarrow C_i}$ from S to C_i remains the same for each pick up event $T_k^{C_i}$ ($1 \leq k \leq K$) at C_i , we may rewrite $D(C_i)$ as $D(C_i) = D_{S \rightarrow C_i} + \bar{D}_{C_i \rightarrow E}$ where $\bar{D}_{C_i \rightarrow E}$ denotes the average driving distance from C_i to E when the driver picks up a customer at C_i . It can be estimated with the historical pick up events at C_i as well. Let us use $D_{C_i \rightarrow Dr_k^{C_i} \rightarrow E}^{C_i}$ ($1 \leq k \leq K$) to denote the distance of each pick even k . Then $\bar{D}_{C_i \rightarrow E}^*$ could be computed

$$\bar{D}_{C_i \rightarrow E} = \sum_{k=1}^K D_{C_i \rightarrow Dr_k^{C_i} \rightarrow E}^{C_i} / K$$

as

[0060] To simplify the further introduction about cost function $\mathcal{G}$, let us

look at an example of driving route: $S \rightarrow C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow E$. When a driver takes this sample route, he may pick up a customer at C_1 , C_2 and C_3 with probability as P_1 , $(1 - P_1)P_2$, and $(1 - P_1)(1 - P_2)P_3$, respectively. We get the cost function $\mathcal{G}$ for estimating the potential driving distance as:

$$\mathcal{G} = P_1 D(C_1) + (1 - P_1)P_2 D(C_2) + (1 - P_1)(1 - P_2)P_3 D(C_3) + (1 - P_1)(1 - P_2)(1 - P_3)D,$$

where D represents the driving distance if a driver does not pick up a customer at any of three locations. In other words, D is the actual driving distance along the route $S \rightarrow C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow E$. $D(C_2)$ and $D(C_3)$ are the estimated driving distances when the driver picks up a customer at C_2 and C_3 , respectively.

[0061] When a driver follows a route $S \rightarrow C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow E$, the probability of picking up passenger(s) on the route is $1 - (1 - P_1)(1 - P_2)(1 - P_3)$, which is the measurement of business success used herein. Alternatively, the potential earning could also be one measurement of business success. For instance, if there are totally K historical pick up events at C_i , let us denote the earning of each historical

pick up event as $E_k^{C_i}$. The average of earnings at C_i is $E(C_i) = \sum_{k=1}^K E_k^{C_i} / K$.

Consequently, for the example route $S \rightarrow C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow E$, the potential earning can be estimated as $P_1 E(C_1) + (1 - P_1)P_2 E(C_2) + (1 - P_1)(1 - P_2)P_3 E(C_3)$.

Computing Algorithm

[0062] An efficient computing algorithm for searching the optimal route of DORR problem shall now be described.

[0063] If a driver takes a route $S \rightarrow C_1 \rightarrow E$ as shown in FIG. 4B, the potential driving distance D_1 for this route can be denoted as $D_1 = P_1(D_{S \rightarrow C_1} + \bar{D}_{C_1 \rightarrow E}) + (1 - P_1)(D_{S \rightarrow C_1} + D_{C_1 \rightarrow E})$ based on the defined function G , where $D_{S \rightarrow C_1}$ (or $\bar{D}_{C_1 \rightarrow E}$) is driving distance from S to C_1 (or from C_1 to E). If the driver takes another route $S \rightarrow C_1 \rightarrow C_2 \rightarrow E$, which is obtained by simply adding a pick up location after C_1 , the potential driving distance D_{12} for this route can be similarly represented as

$$D_{12} = P_1(D_{S \rightarrow C_1} + \bar{D}_{C_1 \rightarrow E}) + (1 - P_1)P_2(D_{S \rightarrow C_1} + D_{C_1 \rightarrow C_2} + \bar{D}_{C_2 \rightarrow E}) + (1 - P_1)(1 - P_2)(D_{S \rightarrow C_1} + D_{C_1 \rightarrow C_2} + D_{C_2 \rightarrow E})$$

[0064] By carefully deriving, we may find that D_{12} cannot be smaller than D_1 . In fact, we may obtain the following general monotone property.

Lemma 1. Given a candidate route $R_1 = S \rightarrow C_1 \rightarrow \dots \rightarrow C_i \rightarrow E$, and another candidate route with an added pick up location following C_i as $R_2 = S \rightarrow C_1 \rightarrow \dots \rightarrow C_i \rightarrow C_{i+1} \rightarrow E$, the potential driving distance for taking R_1 and R_2 is denoted as $D_{1, \dots, i}$ and $D_{1, \dots, i+1}$ respectively. Then we can infer $D_{1, \dots, i} \leq D_{1, \dots, i+1}$. In other words, the potential driving distance cannot decrease when one more pick up location is appended to a candidate route.

[0065] We provide detailed proof for this lemma as follows:

PROOFS FOR LEMMA 1

The potential driving distance $D_{1,...,j}$ for R_1 can be denoted as:

$$\begin{aligned} D_{1,...,j} = & P(C_1)\{D_{S \rightarrow C_1} + \overline{D}_{C_1 \rightarrow E}\} \\ & + (1 - P(C_1))P(C_2)\{D_{S \rightarrow C_1} + D_{C_1 \rightarrow C_2} + \overline{D}_{C_2 \rightarrow E}\} \\ & + \dots + (1 - P(C_1)) \dots (1 - P(C_{n-1}))P(C_n)\{D_{S \rightarrow C_1} + D_{C_1 \rightarrow C_2} + \dots + D_{C_{n-1} \rightarrow C_n} + \overline{D}_{C_n \rightarrow E}\} \\ & + \dots + (1 - P(C_1)) \dots P(C_j)\{D_{S \rightarrow C_1} + D_{C_1 \rightarrow C_2} + \dots + D_{C_{j-1} \rightarrow C_j} + \overline{D}_{C_j \rightarrow E}\} \\ & + (1 - P(C_1)) \dots (1 - P(C_j))\{D_{S \rightarrow C_1} + D_{C_1 \rightarrow C_2} + \dots + D_{C_{j-1} \rightarrow C_j} + D_{C_j \rightarrow E}\}. \end{aligned} \quad (1)$$

The potential driving distance $D_{1,...,j+1}$ for R_2 can be denoted as:

$$\begin{aligned} D_{1,...,j+1} = & P(C_1)\{D_{S \rightarrow C_1} + \overline{D}_{C_1 \rightarrow E}\} \\ & + (1 - P(C_1))P(C_2)\{D_{S \rightarrow C_1} + D_{C_1 \rightarrow C_2} + \overline{D}_{C_2 \rightarrow E}\} \\ & + \dots + (1 - P(C_1)) \dots (1 - P(C_{n-1}))P(C_n)\{D_{S \rightarrow C_1} + D_{C_1 \rightarrow C_2} + \dots + D_{C_{n-1} \rightarrow C_n} + \overline{D}_{C_n \rightarrow E}\} \\ & + \dots + (1 - P(C_1)) \dots P(C_j)\{D_{S \rightarrow C_1} + D_{C_1 \rightarrow C_2} + \dots + D_{C_{j-1} \rightarrow C_j} + \overline{D}_{C_j \rightarrow E}\} \\ & + (1 - P(C_1)) \dots (1 - P(C_j))P(C_{j+1})\{D_{S \rightarrow C_1} + D_{C_1 \rightarrow C_2} + \dots + D_{C_{j-1} \rightarrow C_j} + D_{C_j \rightarrow C_{j+1}} + \overline{D}_{C_{j+1} \rightarrow E}\} \\ & + (1 - P(C_1)) \dots (1 - P(C_j))(1 - P(C_{j+1}))\{D_{S \rightarrow C_1} + D_{C_1 \rightarrow C_2} + \dots + D_{C_{j-1} \rightarrow C_j} + D_{C_j \rightarrow C_{j+1}} + D_{C_{j+1} \rightarrow E}\}. \end{aligned} \quad (2)$$

Then we can derive the $\Delta D = D_{1,...,j+1} - D_{1,...,j}$ as:

$$\begin{aligned} \Delta D = & (1 - P(C_1)) \dots (1 - P(C_j))P(C_{j+1})\{D_{S \rightarrow C_1} + D_{C_1 \rightarrow C_2} + \dots + D_{C_{j-1} \rightarrow C_j} + D_{C_j \rightarrow C_{j+1}} + \overline{D}_{C_{j+1} \rightarrow E}\} \\ & + (1 - P(C_1)) \dots (1 - P(C_j))(1 - P(C_{j+1}))\{D_{S \rightarrow C_1} + D_{C_1 \rightarrow C_2} + \dots + D_{C_{j-1} \rightarrow C_j} + D_{C_j \rightarrow C_{j+1}} + D_{C_{j+1} \rightarrow E}\} \\ & - (1 - P(C_1)) \dots (1 - P(C_j))\{D_{S \rightarrow C_1} + D_{C_1 \rightarrow C_2} + \dots + D_{C_{j-1} \rightarrow C_j} + D_{C_j \rightarrow E}\}. \end{aligned} \quad (3)$$

Let us use Q and B to denote $Q = (1 - P(C_1)) \dots (1 - P(C_j))$ and $B = D_{S \rightarrow C_1} + D_{C_1 \rightarrow C_2} + \dots + D_{C_{j-1} \rightarrow C_j}$, respectively. Now we can rewrite ΔD as:

$$\begin{aligned} \Delta D = & QP(C_{j+1})(B + D_{C_j \rightarrow C_{j+1}} + \overline{D}_{C_{j+1} \rightarrow E}) \\ & + Q(1 - P(C_{j+1}))(B + D_{C_j \rightarrow C_{j+1}} + D_{C_{j+1} \rightarrow E}) - QD_{C_j \rightarrow E} \\ = & QP(C_{j+1})B + QP(C_{j+1})D_{C_j \rightarrow C_{j+1}} + QP(C_{j+1})\overline{D}_{C_{j+1} \rightarrow E} \\ & + QB + QD_{C_j \rightarrow C_{j+1}} + QD_{C_{j+1} \rightarrow E} - QP(C_{j+1})B - QP(C_{j+1})D_{C_j \rightarrow C_{j+1}} \\ & - QP(C_{j+1})D_{C_{j+1} \rightarrow E} - QD_{C_j \rightarrow E} \\ = & QP(C_{j+1})\overline{D}_{C_{j+1} \rightarrow E} - QP(C_{j+1})D_{C_{j+1} \rightarrow E} + \\ & QD_{C_j \rightarrow C_{j+1}} + QD_{C_{j+1} \rightarrow E} - QD_{C_j \rightarrow E} + QB. \end{aligned} \quad (4)$$

As we can see that $QP(C_{j+1})\overline{D}_{C_{j+1} \rightarrow E}$ is equal to or bigger than $QP(C_{j+1})D_{C_{j+1} \rightarrow E}$, and that $QD_{C_j \rightarrow C_{j+1}} + QD_{C_{j+1} \rightarrow E}$ is equal to or bigger than $QD_{C_j \rightarrow E}$. Then we can infer that ΔD is equal to or bigger than 0 and that $D_{1,...,j+1}$ is equal to or bigger than $D_{1,...,j}$. Finally, we can conclude Lemma 1 holds.

[0066]

This monotone property could greatly help us save the computational cost when we search the optimal sequence of pick up locations for a taxi driver in real time, because after we conclude that the potential driving distance of one candidate route R is over the threshold of cost (i.e., TC), we do not need to consider those extended candidate routes which include the candidate route R in the beginning.

[0067]

Furthermore, we identify another monotone property that is also useful for computing potential driving distance. Let us still take the route

$\mathcal{S} \rightarrow C_1 \rightarrow C_2 \rightarrow \mathcal{E}$ in FIG. 4 (b) as an example, where we have

$$D_{12} = P_1(D_{\mathcal{S} \rightarrow C_1} + \bar{D}_{C_1 \rightarrow \mathcal{E}}) + (1 - P_1)P_2(D_{\mathcal{S} \rightarrow C_1} + D_{C_1 \rightarrow C_2} + \bar{D}_{C_2 \rightarrow \mathcal{E}}) + (1 - P_1)(1 - P_2)(D_{\mathcal{S} \rightarrow C_1} + D_{C_1 \rightarrow C_2} + D_{C_2 \rightarrow \mathcal{E}})$$

[0068] We identify that D_{12} will monotonely increase as $D_{C_1 \rightarrow C_2}$, $\bar{D}_{C_2 \rightarrow \mathcal{E}}$ or P_2 increases. In general, we have the following monotone property.

Lemma 2. Given a candidate route $R = \mathcal{S} \rightarrow C_1 \rightarrow \dots \rightarrow C_{i-1} \rightarrow C_i \rightarrow \mathcal{E}$, the potential driving distance for taking route R is denoted as $D_{1, \dots, i}$. We can infer that $D_{1, \dots, i}$ monotonely increases as $D_{C_{i-1} \rightarrow C_i}$, $\bar{D}_{C_i \rightarrow \mathcal{E}}$ or P_i increases

[0069] We provide detailed proof for this lemma as follows:

PROOFS FOR LEMMA 2

The potential driving distance $D_{1, \dots, i}$ for R_i can be denoted as:

$$\begin{aligned} D_{1, \dots, i} = & P(C_1)\{D_{\mathcal{S} \rightarrow C_1} + \bar{D}_{C_1 \rightarrow \mathcal{E}}\} \\ & + (1 - P(C_1))P(C_2)\{D_{\mathcal{S} \rightarrow C_1} + D_{C_1 \rightarrow C_2} + \bar{D}_{C_2 \rightarrow \mathcal{E}}\} \\ & + \dots + (1 - P(C_1)) \dots (1 - P(C_{i-1}))P(C_i)\{D_{\mathcal{S} \rightarrow C_1} + D_{C_1 \rightarrow C_2} + \dots + D_{C_{i-1} \rightarrow C_i} + \bar{D}_{C_i \rightarrow \mathcal{E}}\} \\ & + \dots + (1 - P(C_1)) \dots P(C_i)\{D_{\mathcal{S} \rightarrow C_1} + D_{C_1 \rightarrow C_2} + \dots + D_{C_{i-1} \rightarrow C_i} + \bar{D}_{C_i \rightarrow \mathcal{E}}\} \\ & + (1 - P(C_1)) \dots (1 - P(C_i))\{D_{\mathcal{S} \rightarrow C_1} + D_{C_1 \rightarrow C_2} + \dots + D_{C_{i-1} \rightarrow C_i} + D_{C_i \rightarrow \mathcal{E}}\}. \end{aligned} \quad (5)$$

The coefficients α_1 , α_2 and α_3 for $D_{C_{i-1} \rightarrow C_i}$, $\bar{D}_{C_i \rightarrow \mathcal{E}}$ and $P(C_i)$ can be respectively derived as:

$$\alpha_1 = (1 - P(C_1)) \dots P(C_i) + (1 - P(C_1)) \dots (1 - P(C_i)). \quad (6)$$

$$\alpha_2 = (1 - P(C_1)) \dots P(C_i). \quad (7)$$

$$\begin{aligned} \alpha_3 = & (1 - P(C_1)) \dots (1 - P(C_{i-1}))\{D_{\mathcal{S} \rightarrow C_1} + D_{C_1 \rightarrow C_2} + \dots + D_{C_{i-1} \rightarrow C_i} + \bar{D}_{C_i \rightarrow \mathcal{E}}\} \\ & - (1 - P(C_1)) \dots (1 - P(C_{i-1}))\{D_{\mathcal{S} \rightarrow C_1} + D_{C_1 \rightarrow C_2} + \dots + D_{C_{i-1} \rightarrow C_i} + D_{C_i \rightarrow \mathcal{E}}\} \\ = & (1 - P(C_1)) \dots (1 - P(C_{i-1}))\{\bar{D}_{C_i \rightarrow \mathcal{E}} - D_{C_i \rightarrow \mathcal{E}}\}. \end{aligned} \quad (8)$$

Based on above equations, it is easy to conclude that α_1 , α_2 and α_3 are all bigger than 0. Therefore, $D_{1, \dots, i}$ will increase if $D_{C_{i-1} \rightarrow C_i}$, $\bar{D}_{C_i \rightarrow \mathcal{E}}$ and $P(C_i)$ increase, i.e., Lemma 2 holds.

[0070] With this monotone property, we may be able to prune one candidate route by first checking these three values associated with the last pick up location. For instance, given two candidate routes $R_1 = \mathcal{S} \rightarrow C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow C_4 \rightarrow \mathcal{E}$ and $R_2 = \mathcal{S} \rightarrow C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow C_5 \rightarrow \mathcal{E}$, we can derive that the potential driving distance of R_2 is

equal to or bigger than that of R_1 if we have $D_{C_3 \rightarrow C_4} \leq D_{C_3 \rightarrow C_5}$, $\bar{D}_{C_4 \rightarrow E} \leq \bar{D}_{C_5 \rightarrow E}$ and $P_4 \leq P_5$. Thus we do not need to compute function G for R_2 if we already concluded the potential driving distance of R_1 is over the threshold of cost.

[0071] Based on the two monotone properties, we will first enumerate and check candidate routes with 1 pick-up location. Then we extend previously-considered routes by adding one more pick up location and check each of the extended routes until we get $|\mathbb{C}|$ pick up locations in each route. When we extend routes with l ($1 \leq l \leq |\mathbb{C}| - 1$) pick-up locations, we will apply Lemma 1 to avoid extending those routes whose travel distance is already over the cost threshold; When we check routes with $l + 1$ ($1 \leq l \leq |\mathbb{C}| - 1$) pickup locations, we will apply Lemma 2 to prune possible candidates without computing function G . Basic functionality of an algorithm for DORR is illustrated in FIG. 6, and details are provided below.

Algorithm for DORR:

Input:

S : the original location
 E : the destination location
 $\mathbb{C}$: a set of pick-up locations
 P : a set of probabilities associated with $\mathbb{C}$
 T : a set of pick-up events associated with $\mathbb{C}$
 TC : a cost threshold

Output:

The optimal route from S to E for a driver.

1. **for** $l = 1 : |\mathbb{C}| - 1$
2. Enumerate all routes with l pick-up locations by adding one pick-up location to remaining candidates with $l - 1$ pick-up locations; Avoid considering some routes with l pick-up locations by applying lemma 1
3. Prune routes with l pick-up locations by applying Lemma 2
4. **end for**
5. Compute the function G for all remaining routes and prune those ones with travel distance over TC
6. Compute the business success for all remaining routes
7. Sort them and pick the optimal one with the biggest business success

[0072] A tree structure is used to illustrate the strategy of the above algorithm as shown in FIG. 5, where it is assumed that there are totally six pick-up locations. The root of the tree denotes the empty set of candidate, which is essentially a set of candidates with zero pick-up location. If a route at the level l (i.e., containing 1 pick-up locations) has a potential driving distance over the cost threshold, we do not need to consider all of its extended routes. For instance, if the

route $\mathcal{S} \rightsquigarrow B \rightsquigarrow \mathcal{E}$ on the level 1 is already pruned as its driving distance is over the threshold, there is no need to extend this route (i.e., adding a child to B). Also we may prune some candidate routes on the same level using Lemma 2 and their extended ones as well. For instance, if we conclude the travel distance of route $\mathcal{S} \rightsquigarrow E \rightsquigarrow A \rightsquigarrow \mathcal{E}$ is over the cost threshold and those three values of node B mentioned in Lemma 2 are bigger than those of node A respectively, we can directly prune the route $\mathcal{S} \rightsquigarrow E \rightsquigarrow B \rightsquigarrow \mathcal{E}$ without computing its function $\mathcal{G}$.

Further discussions about RR-RA and DORR problems

[0073] In this section, we discuss the relationship between both problems and other applications.

The Relationship between RR-RA and DORR

[0074] Here we explain the relationship between RR-RA and DORR problems. Both RR-RA and DORR problems are formulated and solved to recommend optimal driving routes to taxi drivers, and they share similar inputs and objectives. With the package of solutions for both RR-RA and DORR problems, we are able to provide a complete solution for route recommendations to taxi drivers: when taxi drivers encounter the destination-oriented scenario, they can obtain route recommendations via the solutions for the DORR problem; otherwise, they could get route recommendations with the solutions for RR-RA problem. On the other hand, there is a difference between them. There is direction and destination constraint in DORR problem, which results in the solution for the RR-RA problem not working for the DORR problem.

Other Applications of RR-RA and DORR

[0075] In this section, we discuss about other potential applications of the RR-RA and the DORR solutions. The first one is a carpool service, which has been provided for many internet-based transportation companies such as Uber, Waze, Takescoop. In the setting of carpool service, many drivers usually have fixed origin (e.g., home) and destination (e.g., work place) and expect to pick up passengers on the way from origin to destination. How to recommend optimal routes for carpool drivers is an interesting and practically important problem. This routing problem can be essentially formulated as the DORR problem, where the objective is to find an optimal driving route that could maximum carpooling probability under given driving distance and direction constraints. The approaches developed for the DORR problem can be used for solving this carpool routing problem. The second

one is searching for a solution to the parking lot problem. Searching for a parking lot is currently a challenging problem in many big cities (such as LA in the U.S.). For instance, it is estimated that people spend an average of 85 hours a year searching for a parking lot in LA and that an average of 30% of traffic in LA is cruising for parking. The present solutions for the RR-RA problem can be used for solving this problem. We can replace the pick-up location and pick-up probability with the parking spot location and spot available probability, respectively. The objective of RR-RA in this setting is then to search for an optimal route that leads to the minimum driving distance before finding a parking spot. The solution methods and algorithms developed for RR-RA problem and described herein could be directly used for solving this parking lot search problem. In summary, it should be emphasized that the present developed methods and algorithms for route recommendations could provide useful inspiration and insight for solving other transportation problems.

Experimental Results

[0076] In this section, we use both real-world and synthetic data to evaluate the efficiency of developed search algorithms and the effectiveness of route recommendations.

The Experimental Setup

[0077] Data and Parameters. In the experiments, we use real-world taxi GPS data that were collected in the San Francisco Bay Area. This data set contains around 30-day GPS location traces of approximately 500 taxis in San Francisco. For each recorded GPS point, there are four attributes: latitude, longitude, fare identifier and time stamp. The fare identifier could be 1 or 0, where 1 indicates taking passenger(s) and 0 means empty cab. In addition, we generate some synthetic data (i.e., pick-up points and probabilities) for testing the scalability of algorithms. We set the parameters for generating pickup location and probability as: $\alpha = 30 \text{ minutes}$, $\tau_1 = 3.0 \text{ miles}$ and $\tau_2 = \text{Dist}(S \leftarrow \mathcal{E})/2$, where $\text{Dist}(S \leftarrow \mathcal{E})$ is the driving distance from $\mathcal{S}$ to $\mathcal{E}$.

[0078] Experimental Environment. The algorithms were implemented in Python. All the experiments were conducted on a Windows 10 with Intel Core i7 6-Core and 12.00GB RAM. The search time for the optimal driving route is the major metric we use to evaluate the efficiency of our algorithms.

[0079] Competing Methods. We compare our search algorithms for the RR-RA and DORR problems with two baseline methods. To the best of our

knowledge, existing work focuses on addressing the MSR problem or other problems. There is no existing method for solving either RR-RA or DORR problem. Therefore, the two baselines that we compare to are straightforward methods for searching optimal route of the RR-RA and DORR problems. The idea of baseline method for RR-RA is to directly compute potential driving distance for all candidate routes and find the optimal one with minimum value. The idea of the baseline algorithm for DORR is directly computing the business success and cost for each possible candidate route and picking the one with maximum business success and cost lower than the cost constraint TC. All acronyms of evaluated algorithms are given in Table 1.

TABLE 1: Acronyms of Competing Methods

SF-RR-RA:	The baseline search algorithm for RR-RA
Pr-RR-RA:	Our pruning-based search algorithm for RR-RA
SF-DORR:	The baseline search algorithm for DORR
Pr-DORR:	Our pruning-based search algorithm for DORR

Efficiency of Recommendations

Performance using RR-RA

[0080] In this subsection, we show the efficiency of different algorithms for addressing the RR-RA problem. We randomly pick some taxi cabs that are empty around 4pm on Friday from our historical GPS data and compute the optimal route for each of them based on its location *PoCab*. To demonstrate the efficiency of our algorithm Pr-RR-RA, we compare its performance with that of baseline method SF-RR-RA.

[0081] As described herein, the number of pick-up locations depends on the position and time of an empty cab that requests recommendation service. For each selected empty cab, we first use the method above to generate the pick-up locations and estimate their pick-up probabilities, then estimate $\tilde{D}_i$ for each pick-up location, and finally compute the optimal route with different algorithms and record their running time. After recording the running time of each algorithm for all cabs, we compute the average of running time for the same number of pick-up points for each individual algorithm. In Table 2, we show the average running time for three different numbers of pick-up locations. The running time shown here does not include the

time for generating pick-up locations, their probabilities and $\tilde{D}_i$ for both algorithms. From Table 2, we can find that our algorithm Pr-RR-RA outperforms the baseline method significantly. Note that in this set of results we set the max number of pick-up locations of driving route as $\mathcal{L} = 4$.

TABLE 2: Comparisons of Computational Time ($\mathcal{L} = 4$)

	SF-RR-RA	Pr-RR-RA
10 Pick-up Points		
Computation Time (Sec)	3.19	1.81
15 Pick-up Points		
Computation Time (Sec)	6.54	3.08
20 Pick-up Points		
Computation Time (Sec)	8.95	4.93

[0082] $\mathcal{L}$ is an important parameter that affects computational complexity. Thus we demonstrate the efficiency of different algorithms with different values of $\mathcal{L}$. Specifically, we repeat the same experiments as the above with different values of $\mathcal{L}$. In Table 3, we show the results with 10 pick-up points. The results demonstrate that the computational time of both algorithms generally increases with the increase of $\mathcal{L}$ and our algorithm consistently outperforms the baseline one.

TABLE 3: Computational Time (Sec) with Different $\mathcal{L}$

	10 Pick-up Points	
	SF-RR-RA	Pr-RR-RA
$\mathcal{L} = 4$	3.19	1.81
$\mathcal{L} = 6$	4.38	2.12
$\mathcal{L} = 8$	7.23	4.02

[0083] For the Pr-RR-RA algorithm, pruning plays an important role to save computational time. We show the average percentage of candidates that are pruned by our algorithm with different values of $\mathcal{L}$ and different numbers of pickup points in Table 4. As can be seen, a large portion of candidates are pruned without computing EDD value in our algorithm.

[0084] To examine the scalability of our algorithm for a large number of pick-up points, we randomly generate potential pick-up points within a specified area,

the pick-up probability and $\tilde{D}_i$ associated with each pick-up point. In total, we have 3 synthetic data sets with 50, 100 and 150 pick-up points, respectively.

TABLE 4: A Summary of Pruning Percentage

10 Pick-up Points			
	$\mathcal{L} = 4$	$\mathcal{L} = 6$	$\mathcal{L} = 8$
Pr-RR-RA	39.7%	41.8%	48.5%
15 Pick-up Points			
	$\mathcal{L} = 4$	$\mathcal{L} = 6$	$\mathcal{L} = 8$
Pr-RR-RA	42.1%	45.3%	51.2%
20 Pick-up Points			
	$\mathcal{L} = 4$	$\mathcal{L} = 6$	$\mathcal{L} = 8$
Pr-RR-RA	41.7%	44.3%	50.5%

[0085] Also we randomly generate 10 positions of empty cab within the same area. For each synthetic data set, we compute and recommend the optimal route for each position of empty cab, and record the computational time of both algorithms. Note that we use Euclidean distance instead of driving distance to measure the traveling distance between pick-up points. In total, we produce 10 recommendations for each data set. We take the average of computational time of these 10 recommendations and show the results in Table 5, where we set parameter $\mathcal{L}$ as different values for the three data sets. As can be seen, our algorithm could scale well with more pick-up points and have more efficiency improvement against the baseline method as the number of pick-up points increases.

TABLE 5: Computational Time (Sec) on Synthetic Data

	SP-RR-RA	Pr-RR-RA
50 Pick-up Points, $\mathcal{L} = 4$		
Computational Time	44.27	21.09
100 Pick-up Points, $\mathcal{L} = 6$		
Computational Time	203.97	82.76
200 Pick-up Points, $\mathcal{L} = 8$		
Computational Time	487.33	158.47

Performances using DORR

[0086] In this section, we show the efficiency of different algorithms for the DORR problem. We randomly pick some taxi cabs that are empty and far away from their common operation areas around 4pm on Friday from our historical GPS

data, and then compute the optimal route for each of them. For each selected cab, we use its current location around 4pm as the original location and the central of its common operation area as the destination location. To demonstrate the efficiency of our algorithm in Figure 6, we compare its performance with that of baseline algorithm SF-DORR.

[0087] As shown above, the number of generated pick-up points depends on original and destination locations. This number may vary among all selected cabs. For each of them, we first use the method above to generate the pick-up locations and estimate their pick-up probabilities, then we compute the optimal route with different algorithms and record their running time. After recording the running time of each algorithm for all cabs, we take the average of running time with the same number of pick-up points for an individual algorithm, and finally show the results in Table 6. Note that the running time shown here excludes the time for clustering of pick up points and probability estimation that is also a part of online search but is the same for all algorithms compared here. By comparing with the baseline method, we can see that our algorithm could consistently save significant computational time over different numbers of pick-up points. Also the computation cost of baseline method increases much faster than that of our algorithm when the number of pick-up points increases.

TABLE 6: Computational Time (Sec) with Different Numbers of Pick-up Point

	SF-DORR	Pr-DORR
6 Pick-up Points		
Computational Time	1.37	0.76
7 Pick-up Points		
Computational Time	7.09	2.06
8 Pick-up Points		
Computational Time	43.85	7.93

[0088] The underlying reason for the improved efficiency of our algorithm is the pruning effect, i.e., filtering out candidates prior to computing their cost and benefit. To demonstrate the pruning effect, we compute the average percentage of pruned candidates at different steps of our algorithm in Table 7, where we use the same experimental setting as in Table 6. As can be seen, while both Lemma 1 and Lemma 2 enable us to prune many candidates, Lemma 1 leads to a greater pruning percentage than Lemma 2 does.

TABLE 7: A Summary of Pruning Percentage

Pruning Percentage by Lemma 1	Pruning Percentage by Lemma 2
6 Pick-up Points	
60.4%	18.7%
7 Pick-up Points	
63.9%	19.6%
8 Pick-up Points	
65.5%	20.9%

[0089] The cost constraint (i.e., TC) may also affect the efficiency because a lower (higher) value of TC leads to more (fewer) candidates pruned in advance based on Lemma 1 and Lemma 2. To show this effect, we repeat the same experiments as the above yet with different values of TC and show the results in Table 8, where the number of pick-up points is 8. In general, the computation time decreases as the value of TC increases for our algorithm. For real practice, we may need to set up this cost constraint parameter based on the empirical experience and possible input from end users. We may obtain different optimal driving routes with different values of TC with the same other inputs.

TABLE 8: Computational Time (Sec) with Different Values of TC

Values of TC	50 Miles	40 Miles	30 Miles
Computational Time of Pr-DORR	7.93	6.87	5.69

[0090] As we mentioned above, the value of TC affects the pruning effect of algorithm. To further demonstrate this, we show the pruning percentage versus different values of TC in Table 9. From Table 9, we could see higher values of TC lead to more percentage of candidates pruned, and consequently less computation time for our algorithm.

Effectiveness of Recommendations

[0091] Other than the computational advantage that we showed above, we examine the effectiveness of our route recommendations. The purpose of this examination is to demonstrate that the recommended driving routes could help taxi drivers pick up customers.

TABLE 9: A Summary of Pruning Percentage

Pruning Percentage by Lemma 1	Pruning Percentage by Lemma 2
8 Pick-up Points, $TC = 8$ Miles	
65.5%	20.9%
8 Pick-up Points, $TC = 7$ Miles	
67.7%	21.3%
8 Pick-up Points, $TC = 6$ Miles	
70.3%	23.6%

[0092] While the ideal way is to conduct field experiments (a.k.a., A/B testing), it is very difficult to collaborate with taxi drivers or taxi companies to run such experiments. Usually, as the first step, offline empirical tests are conducted using classical machine learning leave-out validation methods on historical data, and we follow this approach herein. The basic idea of the leave-out method is to split historical GPS data into training and testing sets along the time dimension, obtain pick-up points and probabilities using training data, compute the optimal routes for empty cabs in testing data, and compare recommended optimal routes with the routes actually taken by drivers in testing data. Specifically, we use 90% GPS data of each taxi cab as training data and hold the remaining 10% GPS data for testing.

Performances of RR-RA

[0093] In this experiment, we extract raw drop off locations from testing data, and compute a recommended route by treating each drop off location as the origin of an empty cab (i.e., *PoCab*), where we set the parameter $\mathcal{L}$ as 4. In testing data, we have the route actually taken by cab starting from each selected drop off location. We then compare each recommended route with the corresponding actually-taken route by a cab. The actually-taken route starting from a raw drop off location may be longer than the corresponding recommended route. To make them comparable, we cut the actually-taken route to the same length as the corresponding recommended one. Consequently, we measure similarity between each pair of routes (i.e., the recommended route and the cut actually-taken route). In total, there are 42,080 pairs of routes. Because the dynamic time warping algorithm for measuring trajectory similarity is very time-consuming, we randomly sample 5% of pairs for our evaluation. In total we have obtained 2,104 pairs of routes. For each actually-taken route, we know whether there is a pick-up event or not within the cut length based on GPS logs in testing data. Based on this indicator, we split all samples of similarity between each pair of routes into two groups: Group 1 including

all similarity samples that has the indicator indicating there is a pick-up event within the really-taken route, and Group 2 including all similarity samples that has the indicator indicating there is no pick-up event. Finally, to get the quantitative measurement of effectiveness, we use a statistical t-test to examine the difference between Group 1 and Group 2. There are 1,046 similarity samples in Group 1 and its mean is 0.69. On the contrary, there are 1,058 similarity samples in Group 2 and its mean is 0.23. The p-value of t-test is less than 0.01, which indicates that the similarity between each pair of routes with actual pick-up event (i.e., Group 1) is significantly bigger than that without actual pick-up event (i.e., Group 2). From these results, we can see that there is a strong correlation between the similarity and the binary indicator. This suggests that recommended routes are very close to the corresponding actually-taken routes that include pick-up events, and that those actually-taken routes that do not include pick-up events are very different from the corresponding recommended routes.

Performances of DORR

[0094] We conduct the similar effectiveness evaluation for the DORR problem. However, unlike the evaluation of RR-RA, we need to extract drop-off locations, each of which is far away from the common operation area of the corresponding cab. In total, we obtain 890 drop-off locations that belong to the destination-constrained scenario. We consider each selected drop-off location as original location (i.e., δ), and the center of the corresponding common operation area as destination location (i.e., ε). We search the optimal route to recommend for each selected drop-off location. In total, we also obtain 890 recommended optimal routes. For each selected drop-off location, we obtain the actually-taken route that starts from the drop-off location from testing data. In total, we initially get 890 pairs of recommended route and actually-taken route. But, some actually-taken routes may end at different locations than the corresponding destination locations (i.e., common operation area). To make a fair comparison between each pair of routes, we filter out those actually-taken routes that do not share the same destination location as the corresponding recommended route. After this filtering, we obtain 803 pairs of recommended route and actually-taken route. We use the same method as used when evaluating the performance of RR-RA to compare the similarity between two routes of a pair. For each pair, we could obtain a binary variable indicating whether there is a pick-up event or not along the actually-taken route from the testing data.

Similarly we split the similarity samples into two groups: Group 1 including all similarity samples where there is a pick-up event within the actually-taken route, and Group 2 including all similarity samples where there is no pick-up event. We also use statistical t-test to examine the difference between Group 1 and Group 2. There are 397 similarity samples in Group 1 and its mean is 0.63. On the contrary, there are 406 similarity samples in Group 2 and its mean is 0.28. The p-value of t-test is less than 0.01. In summary, these results suggest that recommended routes are very close to actually-taken routes that include pick-up events, and those actually-taken routes that do not include pick-up events are very different from the corresponding recommended routes.

Practical Value of Route Recommendation

[0095] To further demonstrate the practical value of route recommendation, we estimate the business improvement of our recommendations that is measured as *empty rate* (ER). ER is defined as the ratio of driving distance without a passenger. We can compute ER using historical GPS data for each taxi driver. The lower value of ER means better business profit of a taxi driver. In our data set, the average ER is 55% over all drivers. 10% of drivers have ER lower than 0.24, which means that they carry passenger(s) and earn money for 76% of their driving. 10% of drivers have ER over 0.7. The goal of our evaluation is to use a simulation method to examine whether we can significantly decrease ERs of taxi drivers with our route recommendations.

[0096] Specifically, we randomly select 100 taxi cabs from the testing data set. For each taxi cab, we extract all driving routes without a passenger from 10% of the historical GPS data in the testing data. For each of these routes, there are a drop-off location at the beginning and one pick-up location in the end. In total, we get 8,030 routes, i.e., 8,030 pairs of drop-off and pick-up locations. We use each drop-off location as an input to generate recommended route. Among these 8,030 pairs, we identify 165 of them as corresponding to the DORR problem. For the remaining 7,865 ones, we consider them as corresponding to RR-RA problem. We use different recommendation methods to search optimal routes for these two groups. For each drop-off location in RR-RA group, assuming the recommended optimal route is taken, we estimate the driving distance before picking up the next passenger(s) as the EDD value. This estimated EDD value is considered as the simulated driving distance before picking up the next passenger(s) when our

recommended route is taken. It is expected that this simulated driving distance is equal to or lower than the actual driving distance prior to the pick-up event. For each drop-off location in the DORR group, we similarly estimate the simulated driving distance without a passenger from S to E when our recommended route is taken. Finally, for each of 100 taxi cabs, we compute two ER values: one (denoted as ER_i) is obtained with the observed actual driving distance without a passenger; the other one (denoted as ER_i^S) is obtained with the simulated driving distance without a passenger. We then further compute the decreasing percentage of ER value as $DP_i^{ER} = (ER_i - ER_i^S) / ER_i (1 \leq i \leq 100)$. The average of $\{DP_i^{ER}\}$ is 0.18, which means our recommendations could potentially decrease their empty rate of driving by 18%. The t-test also shows that DP_i^{ER} is significantly bigger than 0 with p-value less than 0.01. Moreover, we visualize the values of ER_i and $ER_i^S (1 \leq i \leq 100)$ in FIG. 7, where we sort all 100 taxi cabs according to ER_i and align them along x axis. From this figure, we can see that our route recommendations lead to different levels of ER decrease (i.e., $ER_i - ER_i^S$) for different drivers. The higher ER_i is, the greater the decrease of ER is achieved. All these results indicate that *our route recommendations could decrease ER and improve business performance* when they are accepted by taxi drivers.

[0097] Referring to FIG. 8, an exemplary computer-implemented system (hereinafter “system”) **100** for implementing functionality associated with generating an advanced route recommendation as described herein, is shown. The system **100** may include and/or generally support functionality defined by an application **102** that configures one or more processors or computing devices to execute any of the route recommendation functionality described herein. The application **102** may be hosted on one or more of a computing device **104**, which may include a server, controller, a personal computer, a terminal, a workstation, a portable computer, a mobile device, a tablet, a mainframe, or other such computing device. Further, aspects of the application **102** may be made accessible to, displayed upon, or otherwise transmitted to a separate computing device **105** as shown. In general, the application **102** may be programmed to execute one or more of the algorithms described herein for building and/or solving, e.g., the described RR-

RA and DO-RR problems. For example, the application **102** may be implemented as code and/or machine-executable instructions executable by a processor of the computing device **104** that may represent one or more of a procedure, a function, a subprogram, a program, a routine, a subroutine, a module, an object, a software package, a class, or any combination of instructions, data structures, or program statements, and the like. In other words, the application **102** described herein may be implemented by hardware, software, firmware, middleware, microcode, hardware description languages, or any combination thereof. When implemented in software, firmware, middleware or microcode, the program code or code segments to perform the necessary tasks (e.g., a computer-program product) may be stored in a computer-readable or machine-readable medium and the computing device **104** performs the tasks defined by the code.

[0098] The computing device **104** may be configured for administering and providing functionality of the application **102** via a network (not shown), which may include the Internet, an intranet, a virtual private network (VPN), and the like. In some embodiments, a cloud (not shown) may be implemented to execute one or more components of the system **100**. In addition, aspects of the system **100** and/or the application **102** may be implemented using platform as a service (PaaS), and/or software as a service (SaaS) using e.g., Amazon Web Services, or other distributed systems.

[0099] In addition, the computing device **104** may store GPS data, travel data, and/or other such data and/or metadata associated with the application **102** in a database **115**. Further, the database **115** may store metadata associated with operations of the application **102**, such a queries, and historical data, and store information about users of the application **102**.

[00100] As further shown, a vehicle **120** may request service or otherwise access aspects of the application **102**. The vehicle **120** may include a mobile device, infotainment system, or any other device capable of receiving a route recommendation through implementation of the application **102**. In general, the vehicle **120** represents the cab or other driver described herein associated with a driver desiring an optimal route for picking up passengers and monetization thereof.

[00101] In addition, the system **100** may include a plurality of vehicles **130** with GPS and GPS data. Information from the plurality of vehicles **130** may be

generated via one or more GPS satellites **140** and may be accessible to the computing device **104** and the application **102**.

[00102] In some embodiments, at least some features of the application **102** may be made available to a plurality of user devices (not shown) in communication with the computing device **104** of the system **100**. The plurality of user devices may include, without limitation, a controller, a personal computer, a terminal, a workstation, a portable computer, a mobile device, a tablet, a phone, a pager, and a multimedia console.

[00103] Referring to FIG. **9**, a process flow **200** is shown illustrating one possible method or algorithm related to the described RR-RA problem. At block **202** of FIG. **9**, historical GPS data may be accessed or aggregated from, e.g., the plurality of vehicles **130**. In block **204**, pick-up locations and associated pick-up probability may be identified, and the maximum number of pick-up locations may be denoted as L . In block **206**, a list of all possible sequences of pick-up locations may be generated, which may be a number of pick-up locations less than or equal to L , and this list may be stored or otherwise defined within a tree structure.

[00104] In block **208**, a pruning process may be applied to the tree structure based on the proved theories described herein for all candidate routes in the list of possible sequences defined in the tree structure. In block **210**, nodes of the tree structure may be labeled according to predetermined conditions. In block **212**, a potential travel distance may be computed for all remaining candidates of route sequences based using the labels applied to the tree structure. In block **214**, taking into account a current location of a car requesting service, the process flow **200** may output an optimal route with a least potential travel distance.

[00105] Referring to FIG. **10**, a process flow **300** is shown illustrating one possible method or algorithm related to the described DO-RR problem; i.e., a method for solving the DO-RR problem. At block **302**, historical GPS data may be accessed. In block **304**, original and destination locations may be defined, pick-up locations and associated pick-up probability and pick-up events may be defined, as well as a cost threshold.

[00106] In block **306**, L is set to $=1$, the number of pick-up locations in a candidate route. In block **308**, all routes with L pick-up locations may be enumerated based on remaining candidate routes. In block **310**, partial candidate routes may be

pruned with L pick-up locations based on the proved theories described herein. In block **312**, the remaining candidate routes with L pick-up locations may be identified.

[00107] In block **314**, all remaining candidate routes may be collected over rounds. In block **316**, the cost for all remaining candidate routes may be computed, and those candidate routes with costs that exceed a predetermined threshold may be disregarded. In block **318**, the business success may be computed for all remaining candidate routes and a candidate route may be selected based on the maximum rate of success.

Conclusions

[00108] In this disclosure, we introduced a Route Recommendation with Relaxed Assumptions (RR-RA) problem and a Destination-Oriented Route Recommendation (DORR) problem by exploiting a massive GPS log. Our study contributes innovative ideas to the area of mobile recommendations. The solution package provides intelligent route recommendations that could proactively navigate drivers to potential passengers improving their business performance. To generate route recommendations in an efficient way, we developed two complementary smart algorithms for RR-RA and DORR problems, respectively. Both algorithms work based on the identified important monotone property of evaluation functions for RR-RA and DORR problems. Our evaluation results with large-scale GPS data demonstrate the improved efficiency of the proposed recommendation algorithms, and the practical value of our route recommendations.

[00109] In some embodiments, weather information and special events could be further considered into the estimation of pick-up probability and the generation of a recommendation. Also taxi drivers may have preferences for other dimensions such as traffic and long/short trips. One of ordinary skill in the art would appreciate that the disclosed methods could be used for a variety of different applications for routing, planning, and general logistics.

[00110] FIG. **11** is an example schematic diagram of a computing device **700** that may implement various methodologies discussed herein. For example, the computing device **700** may comprise the computing device **104** executing or accessing functionality and/or aspects of the application **102**, or otherwise configured to apply the routing functionality described herein. The computing device **700** includes a bus **701** (i.e., interconnect), at least one processor **702** or other computing

element, at least one communication port **703**, a main memory **704**, a removable storage media **705**, a read-only memory **706**, and a mass storage device **707**. Processor(s) **702** can be any known processor, such as, but not limited to, an Intel® Itanium® or Itanium 2® processor(s), AMD® Opteron® or Athlon MP® processor(s), or Motorola® lines of processors. Communication port **703** can be any of an RS-232 port for use with a modem based dial-up connection, a 10/100 Ethernet port, a Gigabit port using copper or fiber, or a USB port. Communication port(s) **703** may be chosen depending on a network such as a Local Area Network (LAN), a Wide Area Network (WAN), or any network to which the computer device **700** connects. Computing device may further include a transport and/or transit network **755**, a display screen **760**, an I/O port **740**, and an input device **745** such as a mouse or keyboard.

[00111] Main memory **704** can be Random Access Memory (RAM) or any other dynamic storage device(s) commonly known in the art. Read-only memory **706** can be any static storage device(s) such as Programmable Read-Only Memory (PROM) chips for storing static information such as instructions for processor **702**. Mass storage device **707** can be used to store information and instructions. For example, hard disks such as the Adaptec® family of Small Computer Serial Interface (SCSI) drives, an optical disc, an array of disks such as Redundant Array of Independent Disks (RAID), such as the Adaptec® family of RAID drives, or any other mass storage devices, may be used.

[00112] Bus **701** communicatively couples processor(s) **702** with the other memory, storage, and communications blocks. Bus **701** can be a PCI / PCI-X, SCSI, or Universal Serial Bus (USB) based system bus (or other) depending on the storage devices used. Removable storage media **705** can be any kind of external hard drives, thumb drives, Compact Disc – Read Only Memory (CD-ROM), Compact Disc – Re-Writable (CD-RW), Digital Video Disk – Read Only Memory (DVD-ROM), etc.

[00113] Embodiments herein may be provided as a computer program product, which may include a machine-readable medium having stored thereon instructions which may be used to program a computer (or other electronic devices) to perform a process. The machine-readable medium may include, but is not limited to optical discs, CD-ROMs, magneto-optical disks, ROMs, RAMs, erasable programmable read-only memories (EPROMs), electrically erasable programmable

read-only memories (EEPROMs), magnetic or optical cards, flash memory, or other type of media/machine-readable medium suitable for storing electronic instructions. Moreover, embodiments herein may also be downloaded as a computer program product, wherein the program may be transferred from a remote computer to a requesting computer by way of data signals embodied in a carrier wave or other propagation medium via a communication link (e.g., modem or network connection).

[00114] As shown, main memory **704** may be encoded with the application **102** that supports functionality discussed above. In other words, aspects of the application **102** (and/or other resources as described herein) can be embodied as software code such as data and/or logic instructions (e.g., code stored in the memory or on another computer readable medium such as a disk) that supports processing functionality according to different embodiments described herein. During operation of one embodiment, processor(s) **702** accesses main memory **704** via the use of bus **701** in order to launch, run, execute, interpret, or otherwise perform processes, such as through logic instructions, executing on the processor **702** and based on the application **102** stored in main memory or otherwise tangibly stored.

[00115] The description above includes example systems, methods, techniques, instruction sequences, and/or computer program products that embody techniques of the present disclosure. However, it is understood that the described disclosure may be practiced without these specific details. In the present disclosure, the methods disclosed may be implemented as sets of instructions or software readable by a device. Further, it is understood that the specific order or hierarchy of steps in the methods disclosed are instances of example approaches. Based upon design preferences, it is understood that the specific order or hierarchy of steps in the method can be rearranged while remaining within the disclosed subject matter. The accompanying method claims present elements of the various steps in a sample order, and are not necessarily meant to be limited to the specific order or hierarchy presented.

[00116] The described disclosure may be provided as a computer program product, or software, that may include a machine-readable medium having stored thereon instructions, which may be used to program a computer system (or other electronic devices) to perform a process according to the present disclosure. A machine-readable medium includes any mechanism for storing information in a form

(e.g., software, processing application) readable by a machine (e.g., a computer). The machine-readable medium may include, but is not limited to optical storage medium (e.g., CD-ROM); magneto-optical storage medium, read only memory (ROM); random access memory (RAM); erasable programmable memory (e.g., EPROM and EEPROM); flash memory; or other types of medium suitable for storing electronic instructions.

[00117] Certain embodiments may be described herein as including one or more modules or services, such as the components of the application **102**. Such modules are hardware-implemented, and thus include at least one tangible unit capable of performing certain operations and may be configured or arranged in a certain manner. For example, a hardware-implemented module may comprise dedicated circuitry that is permanently configured (e.g., as a special-purpose processor, such as a field-programmable gate array (FPGA) or an application-specific integrated circuit (ASIC)) to perform certain operations. A hardware-implemented module may also comprise programmable circuitry (e.g., as encompassed within a general-purpose processor or other programmable processor) that is temporarily configured by software or firmware to perform certain operations. In some example embodiments, one or more computer systems (e.g., a standalone system, a client and/or server computer system, or a peer-to-peer computer system) or one or more processors may be configured by software (e.g., an application or application portion) as a hardware-implemented module that operates to perform certain operations as described herein.

[00118] Accordingly, the term “hardware-implemented module” or “module” encompasses a tangible entity, be that an entity that is physically constructed, permanently configured (e.g., hardwired), or temporarily configured (e.g., programmed) to operate in a certain manner and/or to perform certain operations described herein. Considering embodiments in which hardware-implemented modules are temporarily configured (e.g., programmed), each of the hardware-implemented modules need not be configured or instantiated at any one instance in time. For example, where the hardware-implemented modules comprise a general-purpose processor configured using software, the general-purpose processor may be configured as respective different hardware-implemented modules at different times. Software may accordingly configure a processor, for example, to

constitute a particular hardware-implemented module at one instance of time and to constitute a different hardware-implemented module at a different instance of time.

[00119] Hardware-implemented modules may provide information to, and/or receive information from, other hardware-implemented modules. Accordingly, the described hardware-implemented modules may be regarded as being communicatively coupled. Where multiple of such hardware-implemented modules exist contemporaneously, communications may be achieved through signal transmission (e.g., over appropriate circuits and buses) that connect the hardware-implemented modules. In embodiments in which multiple hardware-implemented modules are configured or instantiated at different times, communications between such hardware-implemented modules may be achieved, for example, through the storage and retrieval of information in memory structures to which the multiple hardware-implemented modules have access. For example, one hardware-implemented module may perform an operation, and may store the output of that operation in a memory device to which it is communicatively coupled. A further hardware-implemented module may then, at a later time, access the memory device to retrieve and process the stored output. Hardware-implemented modules may also initiate communications with input or output devices.

[00120] It is believed that the present disclosure and many of its attendant advantages should be understood by the foregoing description, and it should be apparent that various changes may be made in the form, construction, and arrangement of the components without departing from the disclosed subject matter or without sacrificing all of its material advantages. The form described is merely explanatory, and it is the intention of the following claims to encompass and include such changes.

[00121] While the present disclosure has been described with reference to various embodiments, it should be understood that these embodiments are illustrative and that the scope of the disclosure is not limited to them. Many variations, modifications, additions, and improvements are possible. More generally, embodiments in accordance with the present disclosure have been described in the context of particular implementations. Functionality may be separated or combined in blocks differently in various embodiments of the disclosure or described with different terminology. These and other variations, modifications, additions, and

improvements may fall within the scope of the disclosure as defined in the claims that follow.

CLAIMS

What is claimed is:

1. A method for generating an advanced route recommendation, comprising:
 - accessing, by a processor, data associated with historical global positioning system information;
 - identifying, by the processor, a plurality of pick-up locations and associated pick-up probability for each of the plurality of pick-up locations;
 - generating, by the processor, a tree structure defining a list of sequences, each of the list of sequences defining candidate routes and including at least one of the plurality of pick-up locations; and
 - pruning, by the processor, the tree structure to compute a sequence for routing a vehicle from the list of sequences.
2. The method of claim 1, wherein the tree structure is associated with a route recommendation with relaxed assumptions solution.
3. The method of claim 2, further comprising, by the processor, solving for the route recommendation with relaxed assumptions solution and pruning the tree structure by:
 - labeling nodes of the tree structure with *I* or *D* when an associated condition is satisfied,
 - estimating a distance for each of the plurality of pick-up locations based on the location and the nodes of the tree structure as labeled, and
 - outputting the sequence for the vehicle, such that the sequence defines a route associated with a minimum expected driving distance (EDD).
4. The method of claim 1, wherein the tree structure is associated with a destination oriented route recommendation solution.

5. The method of claim 4, further comprising, by the processor, solving for the destination oriented route recommendation solution and pruning the tree structure by:
 - identifying an original location and a destination location,
 - enumerating each of the list of sequences with l pick-up locations by
 - adding one pick-up location to remaining candidates of the list of sequences with $l = 1$ pick-up locations,
 - pruning the list of sequences with l pick-up locations,
 - identifying remaining ones of the list of sequences, and
 - applying the remaining ones of the list of sequences to a cost function
 - and that filters out the remaining ones of the list of sequences that do not satisfy a cost threshold.
6. The method of claim 1, further comprising generating by the processor clusters from the past pick-up locations and identifying centroids of the clusters as potential pick-up locations.
7. A computer-implemented system for generating an advanced route recommendation, comprising:
 - a computing device associated with a vehicle, configured to:
 - access data associated with historical GPS information including information about past pick-up locations proximate to a location of the vehicle ,
 - identify a set of potential pick-up locations from the data including a probability for each of the set of potential pick-up locations,
 - estimate a distance between a starting location of the vehicle and each of the set of potential pick-up locations,
 - generate a tree structure defining a list of sequences for routing the vehicle to pick up on or more passengers, each of the list of sequences defining candidate routes and including at least one of the set of potential pick-up locations, and

prune the tree structure to compute a sequence from the list of sequences, the sequence defining an optimal route for picking up passengers from the starting location to an ending location of the vehicle.

8. The computer-implemented system of claim 7, wherein the computing device prunes the tree structure and is further configured to:
 - label nodes of the tree structure with *I* or *D* when an associated condition is satisfied,
 - estimate a distance for each of the set of pick-up locations based on the location and the nodes of the tree structure as labeled, and
 - output the sequence for the vehicle, such that the sequence defines a route associated with a minimum expected driving distance (EDD).
9. The computer-implemented system of claim 7, wherein the tree structure is associated with a destination oriented route recommendation solution.
10. The computer-implemented system of claim 9, wherein the computing device solves for the destination oriented route recommendation solution and is further configured to:
 - identify an original location and a destination location,
 - enumerate each of the list of sequences with *I* pick-up locations by adding one pick-up location to remaining candidates of the list of sequences with *I* = 1 pick-up locations,
 - prune the list of sequences with *I* pick-up locations,
 - identify remaining ones of the list of sequences, and
 - apply the remaining ones of the list of sequences to a cost function and that filters out the remaining ones of the list of sequences that do not satisfy a cost threshold.
11. The computer-implemented system of claim 7, wherein the computing device is further configured to generate clusters from the past pick-up locations and identify centroids of the clusters as potential pick-up locations for the vehicle.

12. A tangible, non-transitory, computer-readable media having instructions encoded thereon, the instructions, when executed by a processor, are operable to:
- access data associated with historical GPS information including information about past pick-up locations proximate to a location of a vehicle ,
 - identify a set of potential pick-up locations from the data including a probability for each of the set of potential pick-up locations,
 - estimate a distance between a starting location of the vehicle and each of the set of potential pick-up locations,
 - generate a tree structure defining a list of sequences for routing the vehicle to pick up on or more passengers, each of the list of sequences defining candidate routes and including at least one of the set of potential pick-up locations, and
 - prune the tree structure to compute a sequence from the list of sequences, the sequence defining an optimal route for picking up passengers from the starting location to an ending location of the vehicle.

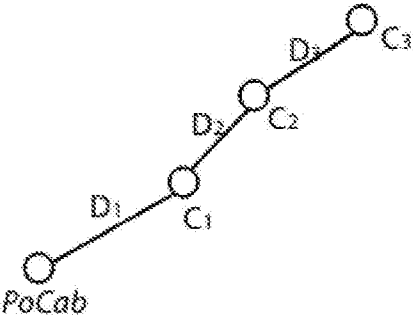


FIG. 1

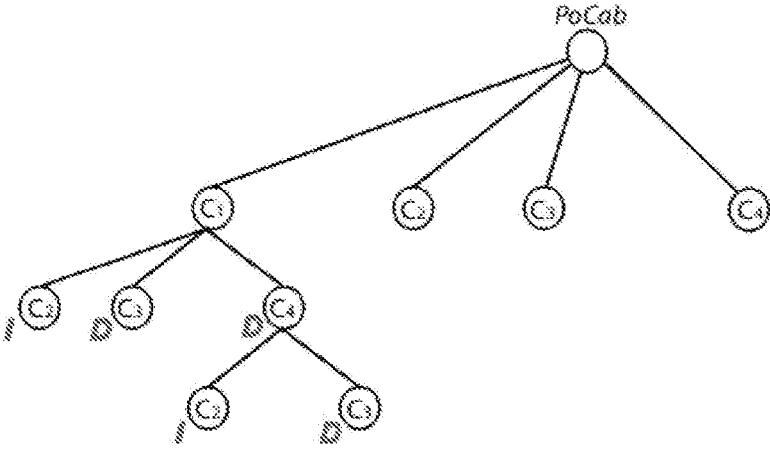


FIG. 2

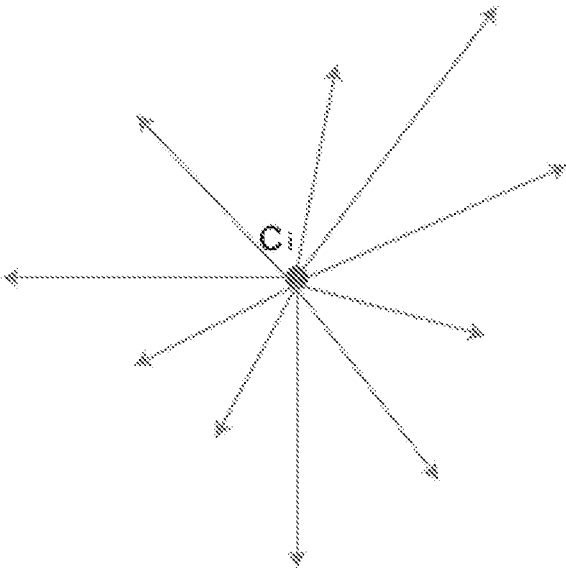


FIG. 3A

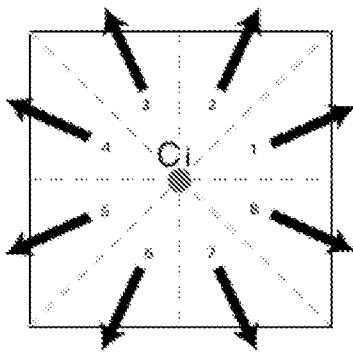


FIG. 3B

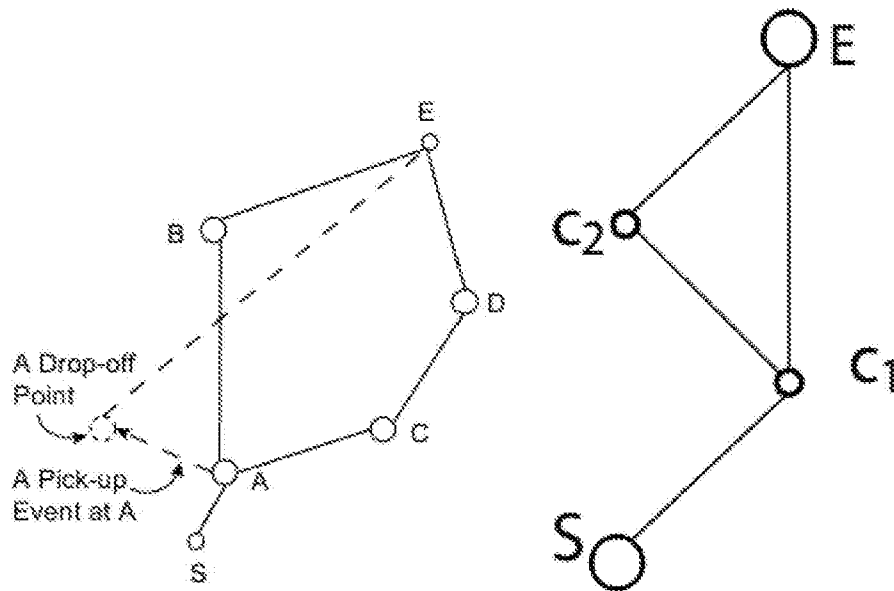


FIG. 4A

FIG. 4B

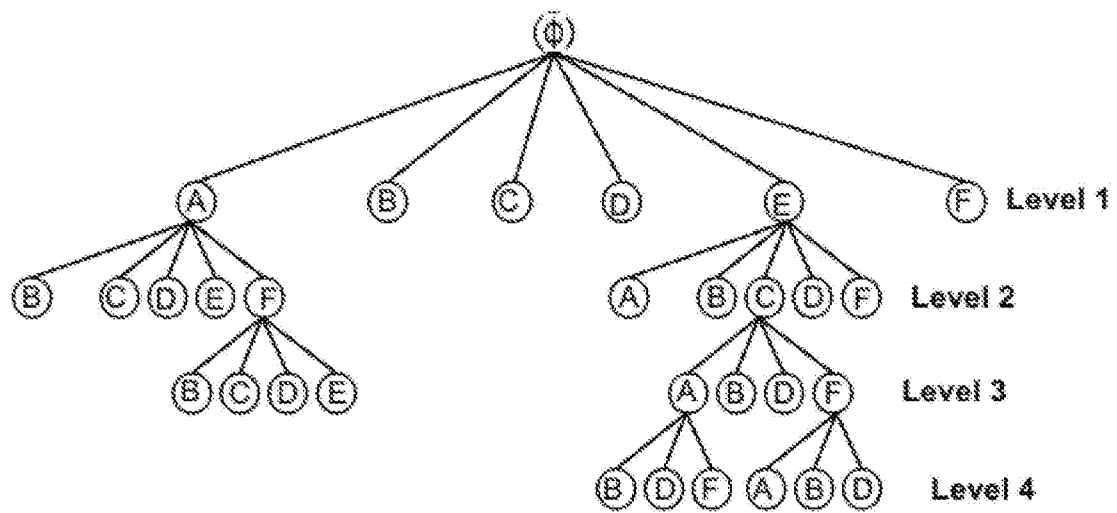


FIG. 5

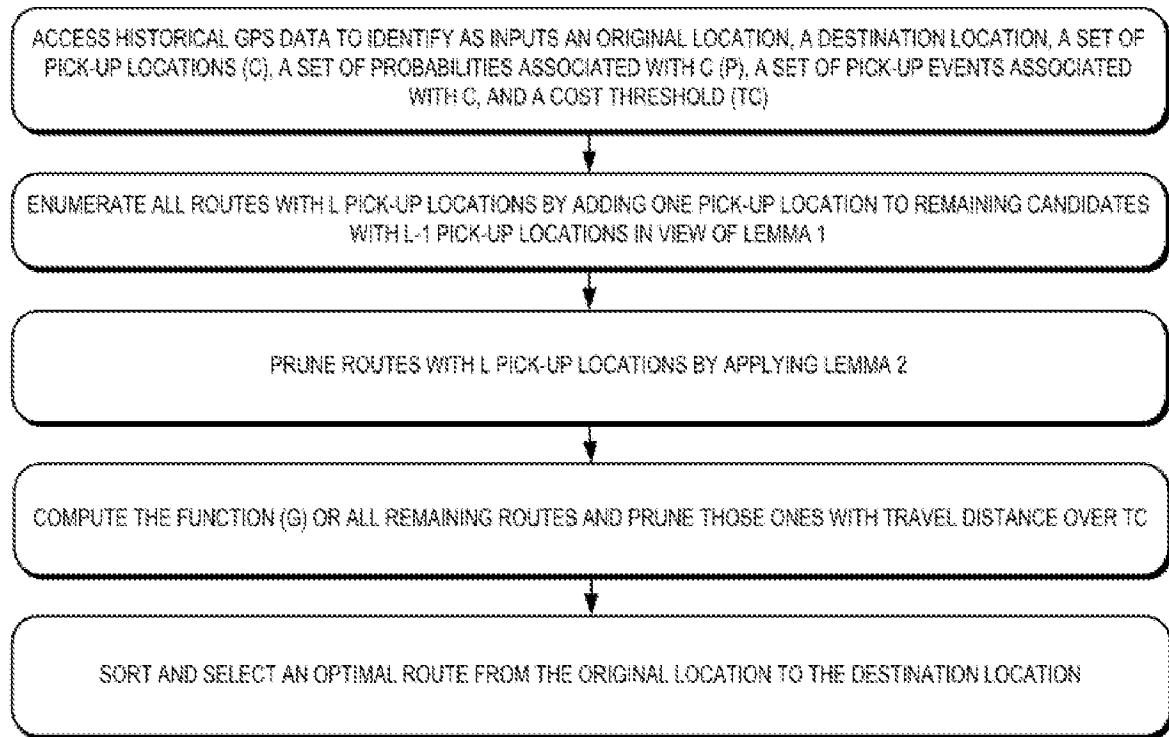


FIG. 6

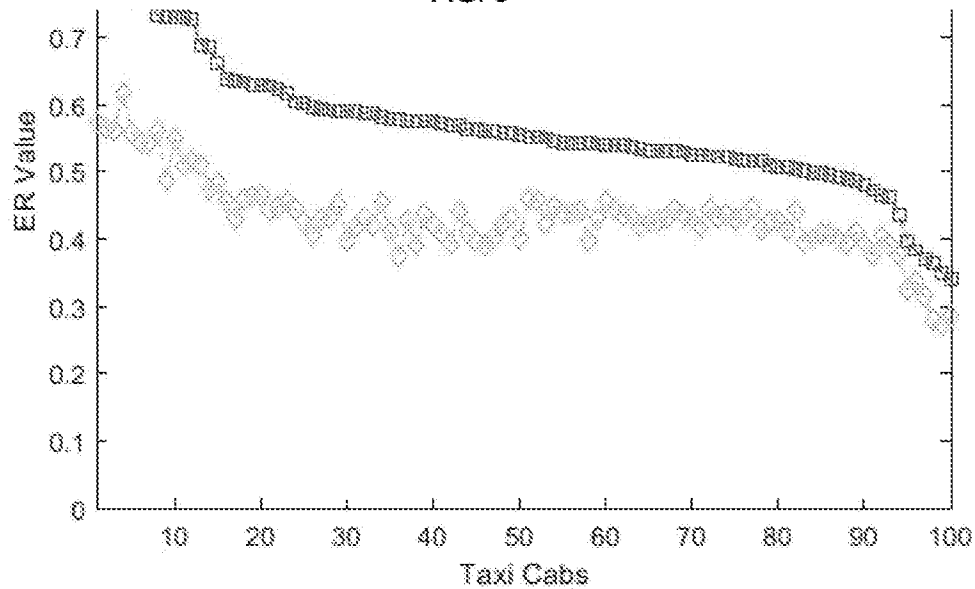


FIG. 7

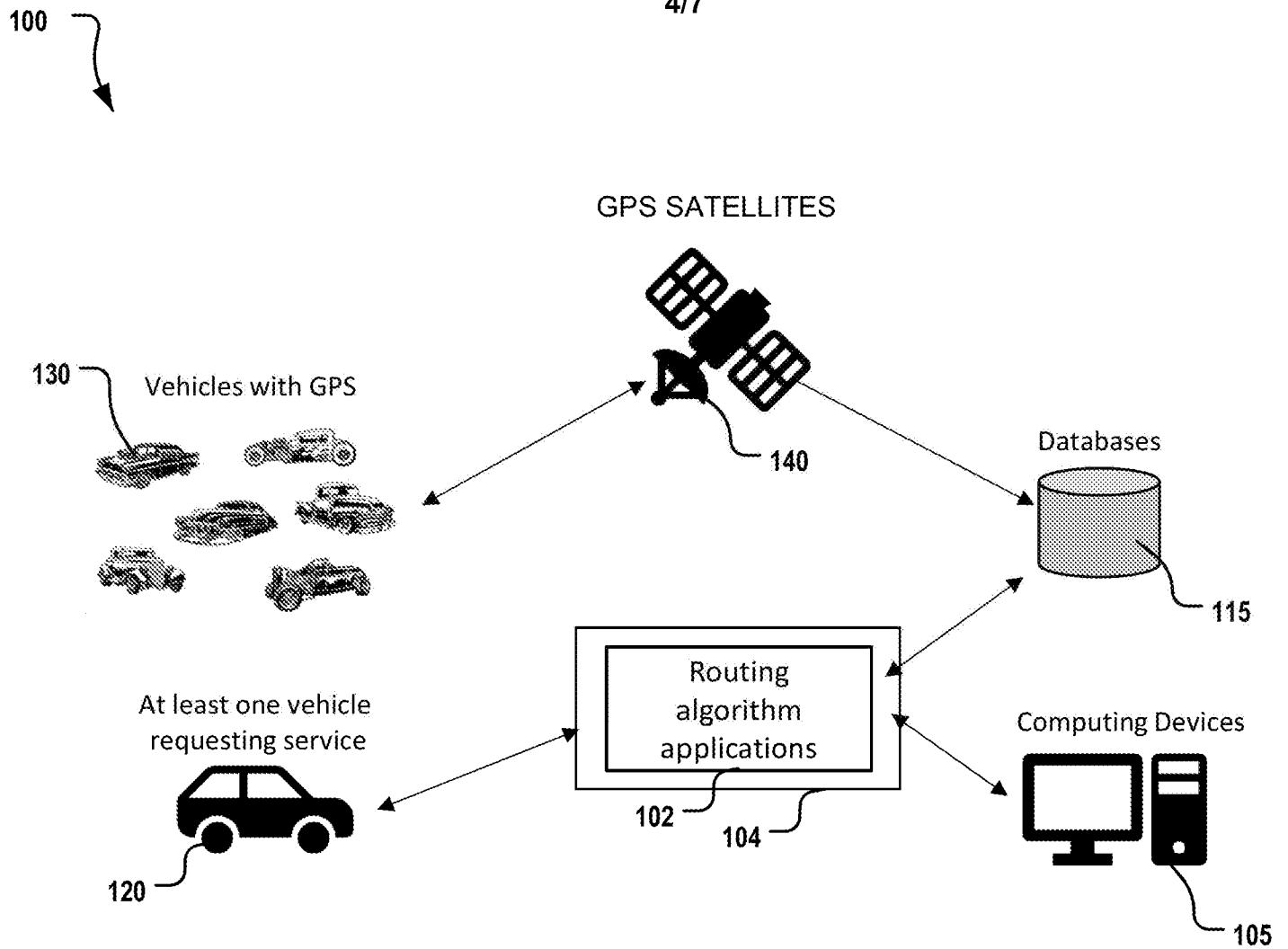


FIG. 8

200

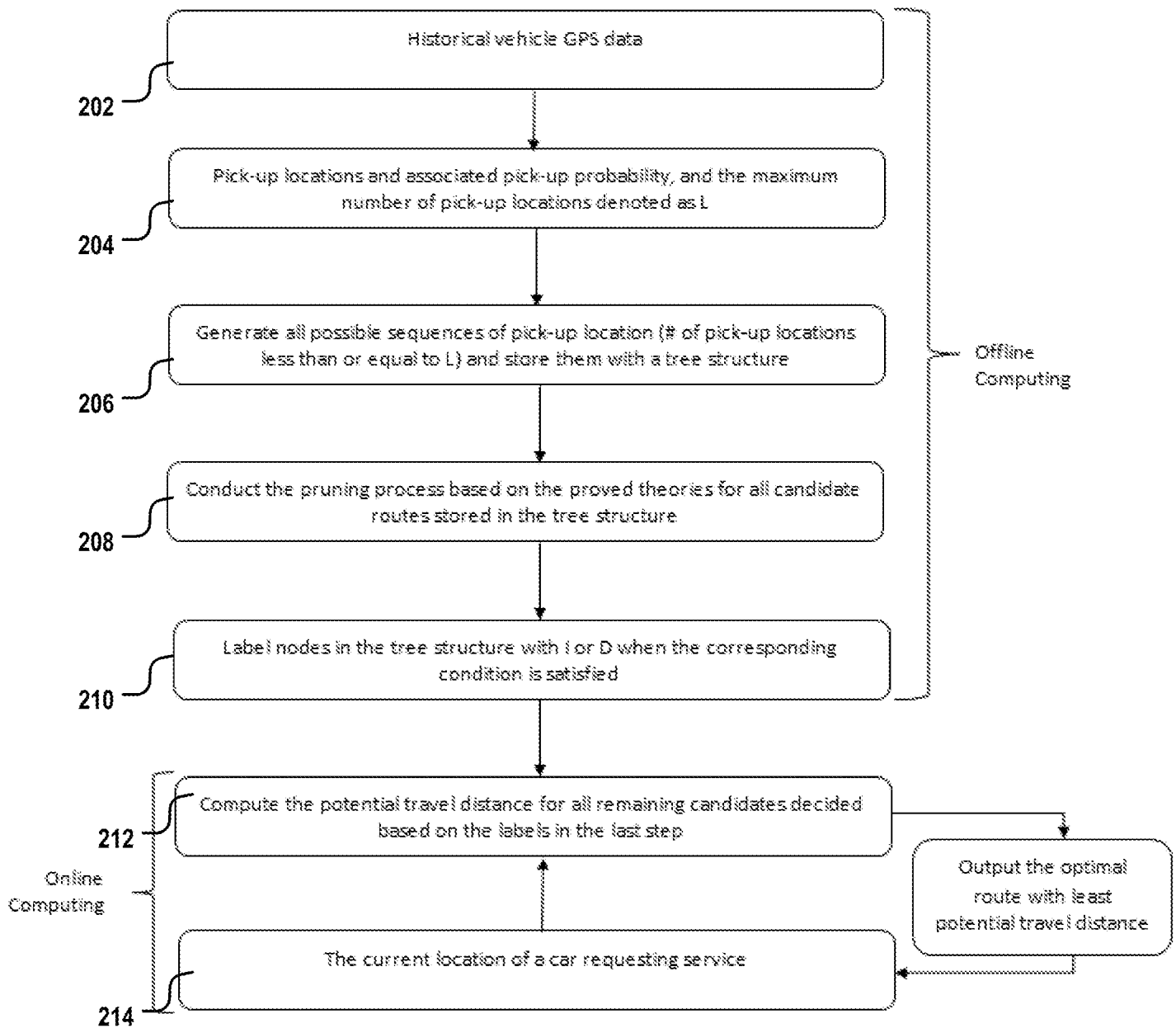


FIG. 9

300

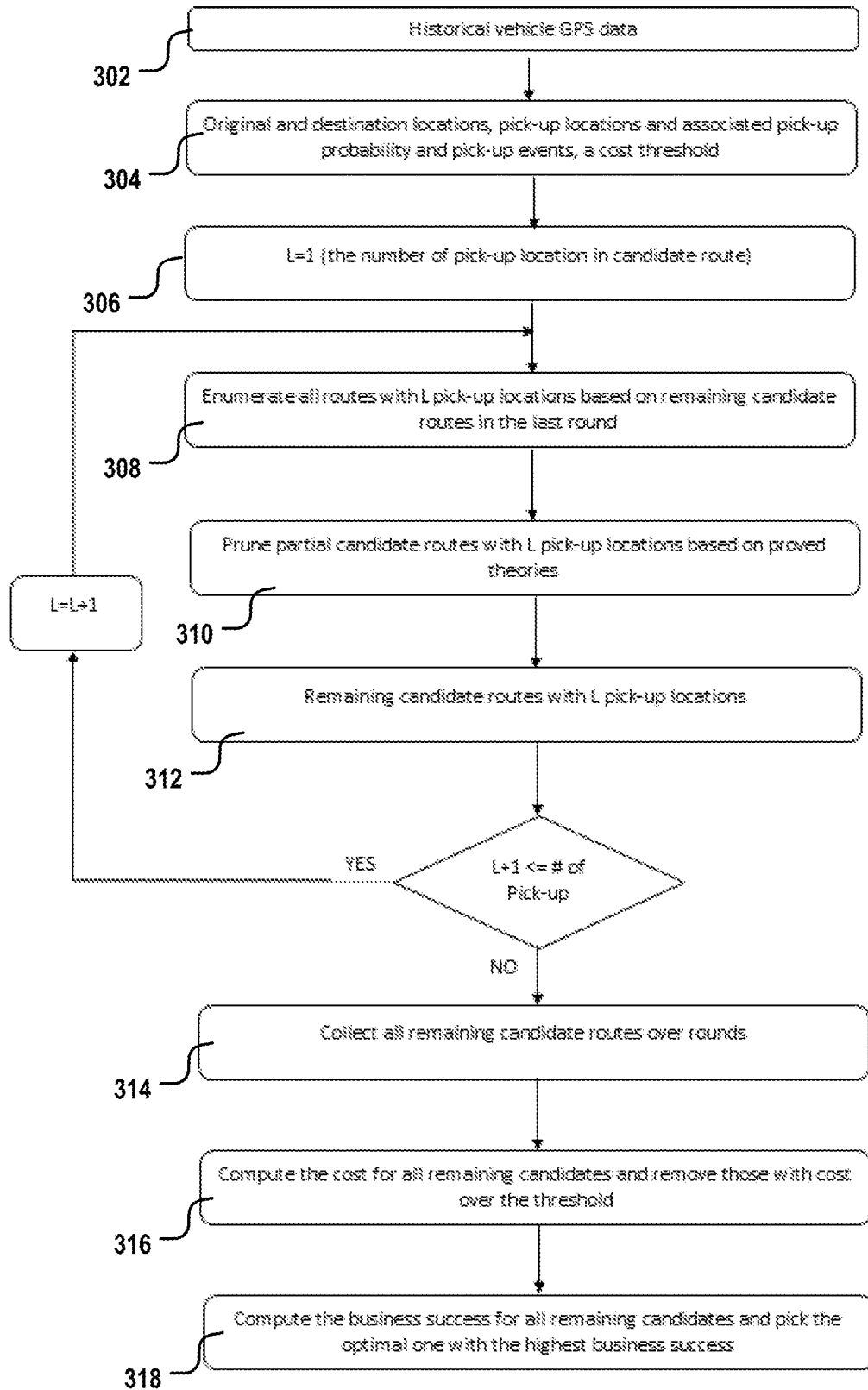


FIG. 10

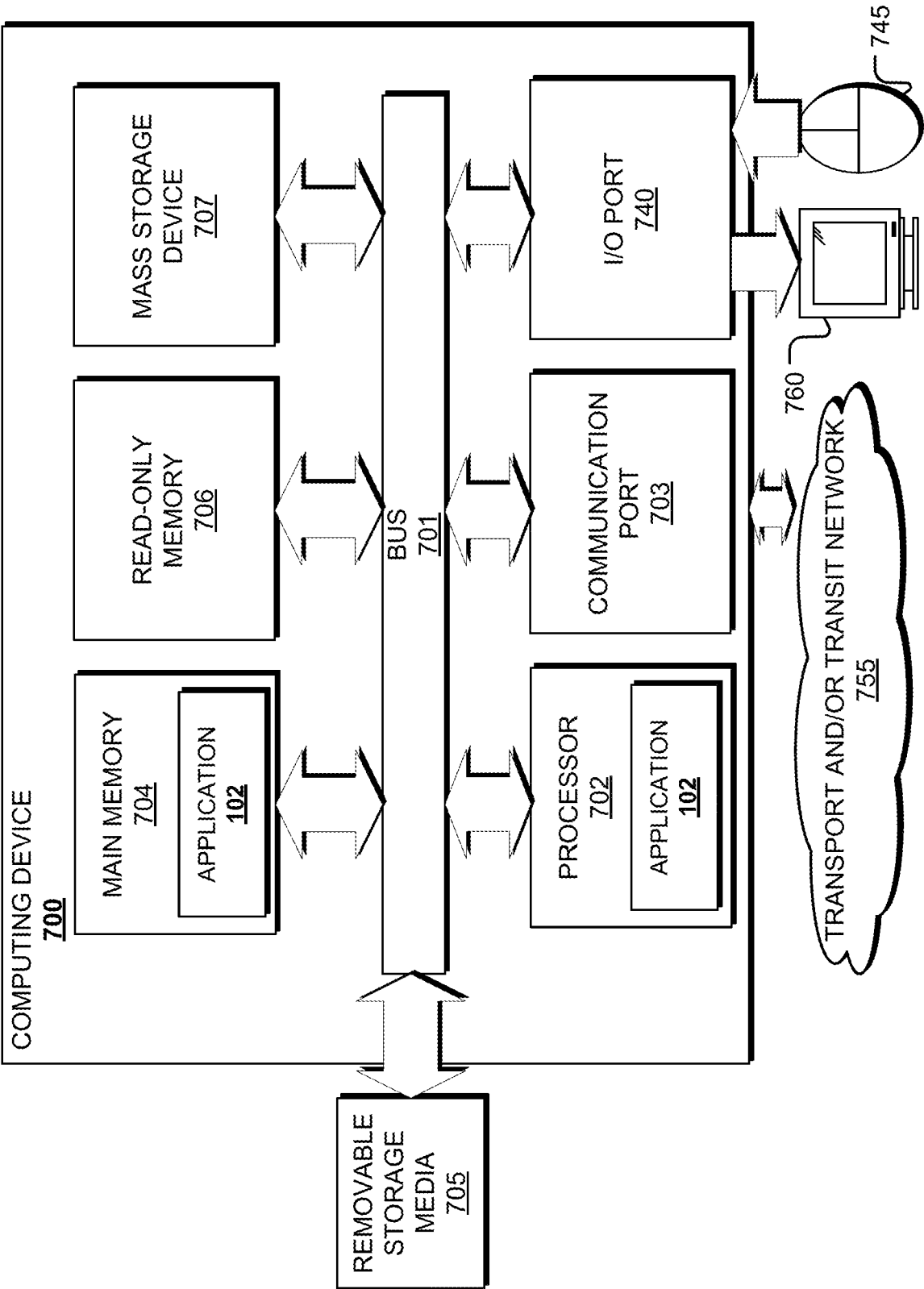


FIG. 11

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US 20/17923

A. CLASSIFICATION OF SUBJECT MATTER

IPC - G05D 1/00 (2020.01)

CPC - G05D 1/0272, G05D 1/0255, G05D 1/0278, G05D 1/027, G05D 1/0274, G05D 2201/0216, G05D 1/0255, G01C 21/3415, G01C 21/30, G01C 21/20, G01C 21/26, G01C 21/3492

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

See Search History document

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

See Search History document

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

See Search History document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2016/0201533 A1 (FORD GLOBAL TECHNOLOGIES, LLC) 14 July 2016 (14.07.2016), entire document, especially Fig. 1, 3, 4; para [0029], [0039], [0061], [0064], [0068], [0069], [0084], [0102]	1-12
A	US 2017/0269599 A1 (ANSARI) 21 September 2017 (21.09.2017), entire document	1-12
A	US 2005/0125148 A1 (VAN BUER et al.) 09 June 2005 (09.06.2005), entire document	1-12
A	US 2004/0128066 A1 (KUDO et al.) 01 July 2004 (01.07.2004), entire document	1-12

☐ Further documents are listed in the continuation of Box C.☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"D" document cited by the applicant in the international application

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

10 April 2020

Date of mailing of the international search report

11 MAY 2020

Name and mailing address of the ISA/US

Mail Stop PCT, Attn: ISA/US, Commissioner for Patents
P.O. Box 1450, Alexandria, Virginia 22313-1450

Facsimile No. 571-273-8300

Authorized officer

Lee Young

Telephone No. PCT Helpdesk: 571-272-4300