Big Data Research 26 (2021) 100246

Contents lists available at ScienceDirect

Big Data Research

www.elsevier.com/locate/bdr

Route Search and Planning: A Survey

Ke Li, Xuan Rao, XiaoBing Pang, Lisi Chen *, Siqi Fan

University of Electronic Science and Technology of China, Chengdu, China

ARTICLE INFO

Article history:
Received 30 April 2021
Received in revised form 11 June 2021
Accepted 20 June 2021
Available online 18 August 2021

Keywords:
Route
Trajectory
Search
Planning

ABSTRACT

Route search and planning have been playing an important role in spatial data management and location-based social services. In this light, we conduct a survey on existing literature regarding route search and planning. We present detailed description for each representative study of route search and route planning. This survey summarizes the findings of existing route search and route planning studies, thus uncovering some new insights that may guide researchers and software engineers in the fields of spatial data management and geographical information systems.

1. Introduction

The continued proliferation of smart transportation and location-based social media allows location-based service providers and users to generate massive-scale location-based data and location-aware queries [1]. Basically, location-based data is formulated as objects consisting of location information and other information including time, text, and attribute-value pairs. Some popular examples include sample points from GPS-equipped devices, check-ins from location-based social media (e.g., Google Maps and Foursquare), and short-text documents with locations (e.g., geo-tagged tweets). Effective management of these kinds of stand-alone location-based objects has been extensively investigated by existing studies.

Recently, connected location-based data has been receiving increasing attention (e.g., [2–4]). Unlike stand-alone location-based data where objects are independent of each other, connected location-based data considers inter-object connection among objects. Since this kind of location-based data is popular in routing services, we may also call it as connected routing data. According to existing studies, each item of connected routing data is defined as a sequence of objects from a road network. Based on the information from objects, connected routing data can be classified into two categories: route data and trajectory data. Basically, route data is defined as a sequence of locations on road networks. Each location may contain additional information such as text description. Similarly, trajectory data is defined as a temporally ordered sequence of location-based objects. Each object consists of spatial, time, and other optional information (e.g., text description).

Over the last couple of decades, a bunch of research has been conducted that aimed at developing effective and efficient route search and planning methods. In this paper, we survey existing studies regarding route search and route planning. In particular, route search problem can be generally defined as finding objective routes and trajectories over a collection of routes and trajectories, respectively, based on query information. While route planning problem can be defined as discovering one or multiple new routes from road networks based on trip requirements given by users. Existing relevant surveys investigate the literature of trajectory data management [5], trajectory data mining [6], trajectory data analytics [7], and spatio-temporal query processing [8–12]. These surveys target the literature of general trajectory and spatio-temporal data management.

In this survey, we first present the formulation of routing data. Next, we present and summarize the formulation of existing problems regarding route search and route planning. Then we present the high-level solutions proposed by existing studies regarding the problems of route search and planning. We also delineate our classifications of these problems and solutions. Finally, we point out some future directions regarding route search and planning.

2. Routing data management

This section presents the classification of routing data based on existing studies (e.g., [1–4,13–17]). A host of related studies on this topic investigate the problem of processing route-based queries (or trajectory-based queries) over existing routes (resp., existing

* Corresponding author.
E-mail addresses: like_like@std.uestc.edu.cn (K. Li), aqua_rao@foxmail.com (X. Rao), p2axelrod@gmail.com (X. Pang), lchen012@e.ntu.edu.sg (L. Chen), sqfann@gmail.com (S. Fan).

https://doi.org/10.1016/j.bdr.2021.100246
2214-5796/

trajectories). In this section, we proceed to introduce the data formulation, route search, and trajectory search, respectively.

2.1. Data and problem formulations

According to the data dependence, the routing data generally can be classified into two categories: stand-alone routing data (e.g., single location, geo-textual object) and connected routing data (e.g., route, trajectory). The main difference between them is that the latter consists of a set of dependent objects (e.g., adjacent location points). We define some primary terminologies in this part, including routing data space (within which the routing data are positioned and applied) and the specific classifications of routing data.

2.1.1. Data space

Two representative data spaces are investigated: Euclidean space and road network space. In Euclidean space, the spatial distance between two items can be regarded as the great-circle distance on the earth [1]. In road network space, the weights of edges generally do not satisfy the triangle inequality. When considering route planning, existing studies mainly focus on road network space.

Road networks. Basically, a road network is formulated by a connected and directed graph $G(V, E, t_m)$, which consists of a set of vertexes V representing road intersections or end-points, and edges $E \subset V \times V$ representing road segments. Each edge $e(v_i, v_j) \in E$ connects two end-points v_i and v_j where $v_i, v_j \in V$. Associated with each road segment e is a weight (or cost) function, denoted by t_m. Function $t_m : E \mapsto R$ assigns a real-valued weight $t_m(e)$ to edge e (e.g., travel time, distance, etc). In most cases, $t_m(e_i) \neq t_m(e_j)$ if $i \neq j$. Regarding different weight functions, the road networks can be further divided into several distinct categories, including but not limited to static road networks, dynamic road networks (e.g., [18–21]), and time-dependent road networks (e.g., [22–26]). For static road networks, the weight of each edge is invariant, namely a constant. Whilst in dynamic road networks the weight of each edge will be dynamically updated according to specific settings (e.g., traffic conditions [18,27,28], self-aware planning [19–21]). In time-dependent road networks, the weight of an edge (e.g., travel time) is time-varying, which is generally formulated based on massive scales of historical trajectories.

2.1.2. Stand-alone routing data

Location. A location is defined as a tuple <x, y> in geographical space (e.g., $39°54'27''N$, $116°23'17''E$). Each tuple corresponds to a unique point recorded by a GPS-enabled device.

Geo-textual point. A geo-textual point is a spatial location by additionally considering some text attributes (e.g., a park with garden) or category attribute (i.e., extracted text attributes such as park, garden). The most common example is geo-tagged check-in tweets.

Spatio-temporal point. The spatio-temporal point is a tuple of timestamp t and a discrete coordinate point $[x, y]$, denoted by $s =< [x, y], t >$. Specifically, the tuple $[x, y]$ denotes the recorded spatial location and t is corresponding timestamp when this location is traveled. One representative type is generated by GPS-equipped devices.

2.1.3. Connected routing data

Exact trace. An exact trace of a moving object is a continuous curve that connects each spatio-temporal point $s =< [x, y], t >$ (cf. section 2.1.2) traveled by the object. A moving object can be a person, an animal, or a mobile location-tracing device. Note that an exact trace cannot be captured in reality as location-tracking devices do not record these locations continuously.

Trajectory. A trajectory $\tau = \{s_1, \ldots, s_n\}$ of a moving object is a finite, temporally ordered sequence of discrete spatio-temporal points, which is sampled from an exact trace with a sampling rate by some location-tracing devices. Each item $s_i \in \tau$ is sorted by its timestamp $s_i.t$. For simplicity, we use a spatio-temporal point $s_i = ([x_i, y_i], t_i)$, $i \in [1, n]$ in a 2D space to denote each connected data item within a trajectory. Note that our modeling of trajectories aligns with existing studies [29,30]. We suppose that all trajectory sample points have already been map matched onto corresponding spatial vertices using some map-matching algorithms (e.g., [31]).

Route. A route (a.k.a., path) **R** is defined as a finite sequence of vertices $v_1, v_2, \ldots, v_k$, which can be represented by a sequence of edges in a road network $G(V, E)$ (i.e., $e(v_i, v_{i+1}) \in E$). Whilst a trajectory cannot be found on a road network, it is generated by sampling from a continuous trace. The trajectory itself relies on an exact trace and its sampling rates: Different sampling rates may lead to generate distinct trajectories, even if the moving object travels the same route.

Fig. 1 illustrates a toy example of the stand-alone routing data (location and the spatio-temporal point) and connected routing data (exact moving trace, trajectory, and the route). The road network, denoted by $G(V, E)$, consists of vertex set $V = \{v_1, v_2, v_3, v_4\}$ and edge set $E = \{v_1v_2, v_2v_3, v_3v_4\}$. Assume that a moving object (e.g., a pedestrian) travels from a source location v_1 to a destination location v_4, through a sequence of spatio-temporal points, a possible trace is shown by a curve. Each spatio-temporal point in the collected trajectory has been matched to a specific road segment on the road network. The corresponding trajectory is generated according to these sampled spatio-temporal points. Here, the route that traveled by the moving object after map matching is $< v_1, v_2, v_3, v_4 >$. Note that an exact moving trace in this example is non-existent, because it cannot be captured continuously by location-tracking devices.

2.2. Route search

With the growing popularity of location-based services, a large number of route planning applications (e.g., Google Maps) and ride-sharing services (e.g., DiDi, Uber, and Grab) are playing an indispensable role in our lives. Instead of merely returning a set of entities (e.g., points of interest), route search is a task of computing a route that starts at a given location (e.g., current location of the user), ends at a specified location and goes via a few geographic entities of user-defined types. Specifically, the geographic entities can be regarded as the user preferences and are specified by route queries. The goal is to find a route that visits at least one satisfying entity of each type. According to the data types of user-defined entities, existing studies in this area can be further divided into two categories: location-based route search (e.g., [32,33]) and keyword-aware route search (e.g., [4,34,35]). We briefly present their problem formulations for comparison here. The details of the two kinds of search problems are discussed in section 3 and section 4, respectively.

Basically, the input of location-based route search consists of a source location s, a destination location t, a set of specific locations **L** (e.g., geographic coordinates), an optional budget limit Δ (e.g., travel time, keyword access order and preference), and a function f that calculates the objective score of a route. The goal is to find an optimal route (i.e., a route with maximum objective score f) from s to t that passes through **L** under the budget limit

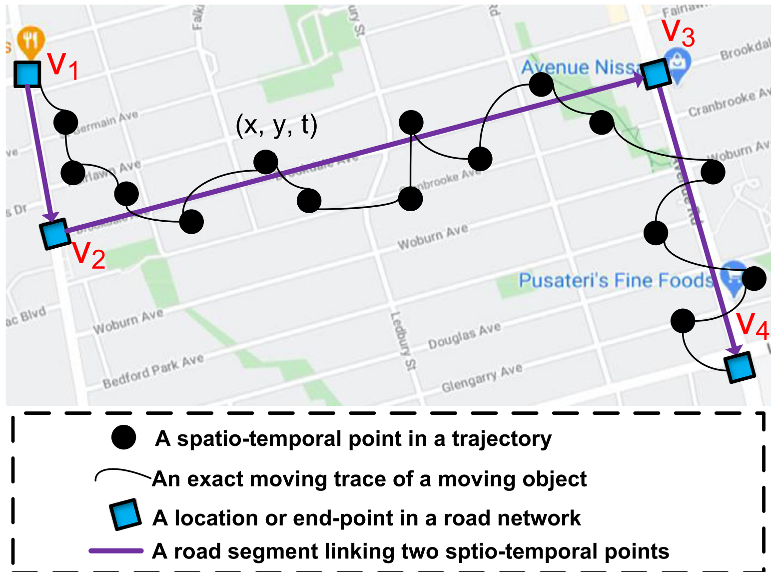

Fig. 1. An example of trajectory and route.

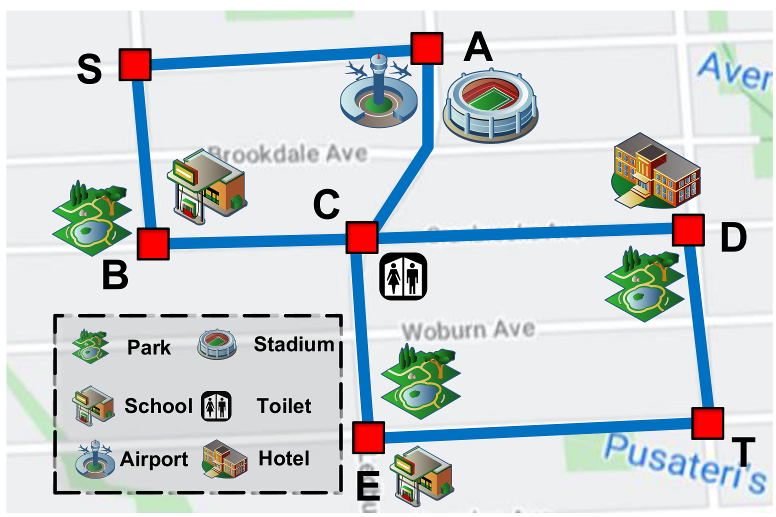

Fig. 2. Location-based versus keyword-aware route search.

Δ. In contrast, the input of keyword-aware route planning problem includes a set of query keywords Ψ instead of some particular locations. This class of query generally returns a path from s to t, such that the objective score f is maximum under budget constraints Δ and keyword-based requirements (e.g., all the keywords or points of interest (POIs) in Ψ must be covered).

Fig. 2 presents a toy example to illustrate the difference between location-based route query and keyword-aware route query. Given a set of spatial locations $\{S, A, B, C, D, E, T\}$, each of which is associated with categorical information (e.g., hotel, park, etc.). As shown in Fig. 2, spatial locations A, B, C, D, E are associated with <airport, stadium>, <park, school>, <toilet>, <hotel, park>, and <school, park>, respectively. Consider a location-based route query such as “Find an optimal route from S to T, passing through A and C”, two candidate routes $\{SA, AC, CD, DT\}$ and $\{SA, AC, CE, ET\}$ are satisfied. Further, a final route is recommended to user based on specific optimization goal (e.g., minimal travel time cost). Instead of providing some particular locations, keyword-aware route planning takes a set of user-defined keywords (e.g., POIs) into account, which aims to find an optimal route that starts from the query location (e.g., a home or hotel), and covers a set of user-specified categories (e.g., pub, restaurant, museum). Given a keyword-aware route query like “Find an optimal route from S to T, passing through a stadium, a toilet and a park”, two candidate routes $\{SA, AC, CD, DT\}$ and $\{SA, AC, CE, ET\}$ are satisfied. Besides, users may also specify partial order constraints between different categories (e.g., a park must be visited before a stadium), which is also a popular query in route planning.

2.3. Trajectory search

How to retrieve close trajectories in spatio-temporal databases is a fundamental problem for trajectory analysis tasks. In this light, we investigate related studies on the topic of trajectory search. In the field of route planning, trajectory search generally can be divided into two sub-categories: trajectory-to-trajectory search and trajectory-to-route search.

2.3.1. Trajectory similarity measure

Before introducing the trajectory search problem, we would like to discuss the existing literature on trajectory similarity measure.

Table 1
Robustness of popular trajectory similarity measures.

Methods	Local Time Shift	Noise	Non-uniform sampling rates	Threshold-free
EDB [36]	✗	✗	✗	✗
DTW [37]	✓	✗	✗	✓
LCSS [38]	✓	✓	✗	✗
ERP [39]	✓	✗	✗	✗
EDR [40]	✓	✓	✗	✗
EDwP [29]	✓	✓	✓	✓
t2vec [43]	✓	✓	✓	✗

Traditional similarity measures. Unlike straightforward distance measures among simple data types (e.g., ordinal variables or geometric points), the distance among trajectories needs to be carefully designed due to their diversity of spatio-temporal attributes. Besides, trajectories are probably generated with different sampling strategies or non-uniform sampling rates. Thus, to develop such a trajectory similarity measure method is non-trivial.

A host of classic methods have been proposed to measure trajectory similarity, among which Euclidean distance based measure (EDB) [36], Dynamic Time Wrapping (DTW) [37], Longest Common Subsequences (LCSS) [38], Edit distance with Real Penalty (ERP) [39], Edit Distance on Real sequence (EDR) [40], and Edit Distance with Projections (EDwP) [29] are the most representative.

A basic idea to match trajectories is to create a one-to-one alignment between the sampled points through the L_p-norm, one work [36] attempts to adapt Euclidean distance as similarity measure. However, this measure suffers in cases of local time shifts: For two trajectories that sampled from the same trace, where one of them is slower in the first half of the distance, and the other is slower in the second half. At a uniform sampling rate, the first trajectory would have more points in the first spatial half than the second. Consequently, the true distance may not be captured. Dynamic Time Warping (DTW) [37] first recognizes this issue and accounts for local time shifts using many-to-one mappings. In particular, DTW adapts a recursive manner to search all possible point combinations between two trajectories to find one with minimal distance. However, DTW is time-consuming when processing a large scale of database. Further, Yi et al. [37] introduce new solutions to improve efficiency.

To better figure out similarities among low-quality trajectories (i.e. noisy trajectories), LCSS [38] is developed to compare two trajectories with the consideration of spatial space shifting. Compared with DTW, LCSS can not only support trajectories of different sizes, but also be more robust to noise. Nevertheless, LCSS is less accurate in comparison to DTW because it is unable to take unmatched trajectory sequences into account.

ERP [39] is an edit distance based trajectory similarity measure. ERP adapts L_1-norm and it is considered to be a metric measure. In contrast, DTW and LCSS do not satisfy the criteria for metric measure. As such, the pruning power of ERP is stronger than those of DTW and LCSS. Similar to DTW, ERP is capable of handling time series data with local time shifts. Nevertheless, ERP is sensitive to noise. The reason is that ERP regards the real-value discrepancies as the distance. Another edit distance based trajectory similarity measure is EDR [40]. Similar to LCSS, EDR uses a threshold ϵ to detect sample points matching and is robust to noise. Different from LCSS, EDR assigns penalties according to the sizes of the gaps, making it capable of addressing some deficiencies hidden in LCSS.

Basically, all the aforementioned methods focus on identifying the optimal alignment based on point matching. As a result, they are inherently sensitive to non-uniform sampling rates. To solve this issue, APM [30] and Edit Distance with Projections (EDwP) [29] are developed. APM learns transition patterns of anchor points from historical trajectories. To calculate the similarity between two trajectories, EDwP finds the cheapest set of insertion and swap operations by adopting linear interpolation to make one trajectory be the same as the other trajectory. Besides, another study [41] proposes a landmark model using landmark similarity, which is consistent with human intuition and episodic memory. The subsequent work [42] reviews the weaknesses of existing warping distances and develops a novel wrapping distance to handle shifting and scaling.

Deep learning based methods. Li et al. [43] propose the first deep learning model (i.e., t2vec) to learn representations of trajectories. The similarity between two trajectories is regarded as corresponding vector distance between their learned representations. t2vec model is robust to low-quality trajectories (e.g., with noise or non-uniform sampling rates), and is capable of achieving high efficiency in trajectory match. Generally, the traditional measures are based on the dynamic programming technique to identify the optimal alignment, which lead to $O(n^2)$ computation complexity. In contrast, t2vec has a linear time complexity $O(n+|v|)$ to measure the similarity of two trajectories, where n is node size and v is the size of representation vector. As a result, t2vec is able to support analysis on big trajectory data. Further, Yao et al. [44] improve the efficiency of deep learning based trajectory similarity computation, in which a generic seed-guided neural learning approach is developed.

Table 1 summaries some popular trajectory similarity measures proposed by existing studies. In order to compare their robustness ability of processing low-quality trajectories, we take four main metrics into account: local time shift, noise, non-uniform sampling rates, and threshold-free, respectively. It is worth noting that only EDB [36] (i.e., Euclidean distance based measure) cannot handle local time shift. When processing noisy trajectories, EDwP and t2vec are both able to achieve promising performance. Besides, only DTW and EDwP are threshold-free, namely no extra augments are required.

2.3.2. Trajectory-to-trajectory search

This category of existing studies aims to retrieve existing (segments of) trajectories that match certain similarity query from a trajectory database based on specific similarity measure (e.g., [29,36–38]). A location-based trajectory-to-trajectory search attempts to retrieve trajectories with the lowest aggregate distance to query locations. Chen et al. [45] are the first to explore this problem, which aims to find the k-Best Connected Trajectories (k-BCT) from a database such that the k-BCT best connect the predefined locations geographically. Shang et al. [46] define a user oriented trajectory search (UOTS) problem, The problem takes a set of query locations and a set of perferences (textual attributes) defined by the traveler as input. If a trajectory is close to the query locations and the textual attributes are similar to the traveler's preference, the trajectory will be returned to the traveler as a query result. A pair of upper and lower bounds are developed for the pruning purpose. Besides, a heuristic searching strategy is developed in [46]. Zheng et al. [47] study the problem of efficient similarity search on activity trajectory database, which is named as activity trajectory similarity query (ATSQ). Given a sequence of query locations, each associated with a set of desired activities, ATSQ returns k trajectories that cover the query activities and yield the shortest minimum match distance. Subsequently, Shang et al. [48] study the personalized trajectory matching (PTM) problem. In comparison to traditional trajectory similarity computation problem, the PTM problem additionally considers the significance of each sample point in a query trajectory.

In recent years, Shang et al. [49] perform a TS-Join and a Tb-TS-Join on two trajectory sets P and Q. The TS-Join returns all pairs of trajectories from P and Q with a similarity that exceeds a pre-defined threshold θ. In contrast, the k-TS-Join does not take a threshold as a parameter, and it returns the top-k most similar trajectory pairs from the two sets. Successively, Shang et al. [2] present a two-phase divide-and-conquer search framework that relies on different pruning techniques to achieve high efficiency. Given an argument set of trajectories, a TSR query [50] is defined by a set of regions of interest. It returns the trajectory with the highest spatial-density correlation to the query regions. Given a set of trajectories and a set of locations, Shang et al. [51] attempt to find all (trajectory, location) pairs from the two sets with spatio-temporal correlation above a pre-defined threshold θ. The authors conduct a novel parallel collaborative search method based on a divide-and-conquer strategy. In addition, an upper bound on the spatio-temporal correlation and a heuristic scheduling strategy are developed to prune the search space.

2.3.3. Trajectory-to-route search

This category of existing studies aims to construct a route based on the retrieved trajectories or sub-trajectories from a spatio-temporal database. Chen et al. [52] investigate the problem of discovering the most popular route (MPR) between two locations, by observing the traveling behaviors of previous users from historical trajectories. A maximum probability product algorithm is proposed to discover the MPR from a transfer network. Dai et al. [53] focus on the problem of how to recommend personalized routes to individual drivers using big trajectory data. Another work [27] studies a novel problem of planning unobstructed paths in traffic-aware spatial networks (TAUP queries) to avoid potential traffic congestions, and constructs a traffic-aware spatial network by analyzing uncertain trajectories of moving objects. As for the sparse trajectory sets, Guo et al. [54] construct a graph-like structure from trajectories as the routing infrastructure, which enables trajectory-based routing with an arbitrary (source, destination) pair as input. Given a set O of query locations, a set T of routes, a threshold θ, and the maximum number of combination M, Chen et al. [55] find the route P composed of sub-routes such that P has similarity to O no less than threshold θ and contains the minimum number of sub-routes combinations. To address it, two parallel algorithms are developed. The high-level idea of this work is to split candidate routes into sub-routes and combine them to construct new routes. Recently, one study [56] proposes a Behavior-based Route Recommendation (BR2) method, which aims to find personalized routes based exclusively on users' travel preferences using trajectory data.

3. Location-based route planning

Route planning has been playing an indispensable role in our lives. In this section, we discuss the problem of location-based route planning. Three types of related studies are investigated in this area: source-destination based route planning, multi-location based route planning and multi-user based route planning, respectively.

3.1. Route planning based on source and destination

This category of studies perform basic route queries in presence of some particular constraints. Given a tuple <source, destination> and perhaps some time-based constraints, the source-destination based route planning aims to derive an optimal route (e.g., the shortest distance, the minimum congestion probability) from source location to destination location under user-specific constraints (e.g., departure time). In this area, many efforts have been made to handle the distance based shortest path search over static graph. Dijkstra [57] is a well known as a shortest path algorithm that finds a path from a place s to another t on static graph. And A* algorithm [58] further improves the search efficiency by adapting heuristic strategies.

3.1.1. Route planning over dynamic road network

In most cases, route planning is in presence of frequent updates to the traffic graph due to the dynamic nature of traffic network, and such updates always greatly affect the performance of route planning [18,22–27]. We proceed to introduce time-dependent shortest problem (TDSP), traffic-aware route planning, self-aware route planning, and user preference based route planning.

Time-dependent shortest problem (TDSP). Given a time-dependent road network (cf. section 2.1.1) and a user-given starting-time interval, the TDSP query aims to find the best departure time that minimizes the total travel time from a place to another, in which the traffic conditions dynamically change from time to time [22–26]. Particularly, it is studied to either find approximate answers with discrete-time approaches [23] or find optimal answers with continuous-time approaches [24,25]. Ding et al. [22] propose a new Dijkstra-based algorithm to find the optimal departure time. A subsequent study [26] constructs a novel height-balanced tree-structured index, called TD-G-tree. Based on TD-G-tree, two efficient algorithms are developed to support TDSP queries through dynamic programming and chronological divide-and-conquer.

Traffic-aware route planning. Basically, traffic-aware route planning mainly focus on finding a route with the least travel time by taking real-time traffic conditions into account. Xu et al. [18] analyze the traffic conditions on the road network and explore spatial-temporal knowledge to guide effective route planning. To avoid both unnecessary calculations on huge graph and excessive recalculations caused by traffic conditions updates, a set of effective techniques are adapted. Shang et al. [27] address a novel problem of planning unobstructed paths in traffic-aware spatial networks to avoid potential traffic congestions, called TAUP queries. A traffic-aware spatial network is constructed by analyzing uncertain trajectories of moving objects, and then two efficient algorithms are developed. Another study [28] aims to find unobstructed routes to avoid over-crowded stations. The solution involves two steps: using human-tracking data to predict human mobility and finding over-crowded stations. In addition, Malviya et al. [59] develop a continuous query system that updates the shortest path for each user in the presence of updates to current traffic, in which two new classes of approximate techniques, K-paths and proximity measures, are presented.

Self-aware route planning. As routing systems become more pervasive, it is obvious that the planned routes may significantly influence the future traffic conditions. That means, prior plans may become relevant to future plans [19–21]. To this end, self-aware route planning is presented to avoid potential traffic congestions caused by previously planned routes. The earliest literature [19] presents a novel algorithm for self-aware route planning that uses the routes it plans for current vehicle traffic to more accurately predict future traffic conditions for subsequent cars. In subsequent study [21], if routes are created for a specific (or sizable) percentage of the total vehicle population, an estimate for the overall traffic pattern is attainable. The authors present an approach that is suitable for realistic, city-scale scenarios.

User preference based route planning. Existing routing algorithms basically aim to identify an optimal route based on certain optimization goal (e.g., the shortest path). However, one work [60] shows that in reality drivers follow paths that are often neither fastest nor shortest. A recent work by Guo et al. [16] observes that drivers may have different contexts that are characterized by different routing preferences. Motivated by this, it studies a context-aware, preference-based routing problem, which takes contexts and routing preferences into account. Existing works on the topic of preference-based routing mainly focus on mining trajectory data to improve the quality of planned routes, where routing preferences are learned from historical trajectories and then used to guide routing algorithms to produce paths [60–62]. The problem of skyline path query is also studied in works [63,64] based on different criterions.

3.1.2. Route planning over uncertain road network

Route planning over stochastic road network. For a given source s and destination d in the graph, the stochastic route planning seeks a sd-path of lowest expected cost where the edge travel time is random variable. Lim et al. [65] present a stochastic motion planning algorithm, which copes with the uncertainty of road traffic conditions by stochastic modeling travel delays, it can be used to find paths that maximize the probability of reaching a destination within a particular travel deadline. Wilkie et al. [19] take the previously planned routes into account, it assumes the input data to be traffic densities of the road segments in the network and translates these densities to velocities and travel time. Yang et al. [66] define stochastic skyline routes that consider multiple costs and time-dependent uncertainty and propose an efficient algorithm. Another study [14] presents a practical approach to transform GPS trajectories into time-varying, uncertain edge weights that guarantee the first-in-first-out property, and it proposes a generic speed-up technique that supports a wide variety of stochastic route planning functionality. In a recent study [15], the authors use trajectory data to create a high-resolution, uncertain road-network graph, where edges are associated with travel-time distributions. It aims to find a path with the highest probability of arriving at a destination within a given time budget. Existing stochastic routing algorithms often assume that edge cost distributions are independent (e.g., [66]). In contrast, Yang et al. [67] consider cost dependence, but it adapts stochastic dominance-based pruning if two edges are independent and thus is inefficient. To address it, Pedersen et al. [15] take the spatial dependence among adjacent edges into account and propose an anytime routing algorithm.

Route planning over probabilistic road network. Time-dependent road network [22] and probabilistic road network [68] are two representative types of dynamic spatial networks. In time-dependent road network, each edge is augmented by a time-dependent weight, which is changing over time. In contrast, the edge weights on probabilistic road network can be either time-dependent or time-independent. Hua et al. [68] propose three novel types of probabilistic path queries. To better capture the uncertainty in traffic such as the travel time between two vertices, the weight of an edge is modeled as a random variable and is approximated by a set of samples. To answer these queries, a set of efficient probability calculation methods are developed.

3.2. Route planning based on multiple locations

Basically, the input of the problems in this category consists of a set of locations, or a sequence of locations, or a sequence of location-time pairs. The existing studies mainly focus on recommending an optimal driving route that covers some user-given locations such that certain cost (e.g., travel time, travel distance, match distance, etc.) is minimal, or certain revenue (e.g., user preferences, platform revenue) is maximum.

3.2.1. Mobile sequential recommendation (MSR)

As a representative problem of human mobility, the mobile sequential recommendation (MSR) problem proposed by [32] aims to search the optimal route for taxi drivers to minimize the potential traveling distance (PTD) before the next successful pick-up.

On this topic, Ge et al. [32] exploit the knowledge extracted from location traces and develop a mobile recommending system based on business success metrics. Two algorithms are developed for finding the recommended routes. The subsequent work by Huang et al. [69] proposes a dynamic programming based method to solve this problem. Further, Ye et al. [70] propose an expected traveling time (ETT) function to serve as the new objective function of the MSR problem. By combining parallel computing and the simulated annealing with novel global and local searches, two move generation algorithms are proposed in [71] to generate multiple route recommendations based on various time constraints and energy requirements. Ye et al. [72] are the first to study the multiple mobile sequential recommendation problem that generates optimal routes for a group of taxis with different locations.

3.2.2. Trajectory search by locations (TSL) problem

This category of problems takes a set of query locations (optional ordered), a trajectory database and a constant k as input, while it returns the k Best-Connected Trajectories (k-BCT) from a database such that the k-BCT best connect the designated locations geographically. Chen et al. [45] achieve efficient k-BCT search based on a simple Incremental k-NN based Algorithm (IKNN), which adapts the best-first and depth-first k-NN algorithms to the basic IKNN properly. The subsequent work [47] aims to retrieve k trajectories that cover the query activities (order-sensitive) and yield the shortest minimum match distance. Each query location is associated with a set of desired activities. And a novel hybrid grid index is developed to organize the trajectory segments and activities hierarchically.

Another popular query regarding TSL is route search by locations (RSL query). Given a sequence of user specified intended places O_q and a departure time t, Shang et al. [73] find the fastest path connecting O_q and satisfying that moving objects (e.g., travelers and bags) can arrive at the destination in time. Given a set O_q of query locations, a set T of routes, a threshold θ, and the maximum number of combination M, Chen et al. [55] target to find the route P composed of sub-route, such that P has similarity to O_q no less than θ and contains the minimum number of sub-routes combinations. To answer this problem, two parallel algorithms are developed. The high-level idea of the algorithms is to split candidate routes into sub-routes and then combine them to construct new routes. Recently, Chen et al. [74] study a novel Continuous Route-Search-by-Location (C-RSL) problem to enable real-time route search by locations.

3.2.3. Collective travel planning (CTP)

Multiple travelers sometimes may target the same destination (e.g., a stadium or a school), in such case they would like to assemble at some meeting locations and then go to the destination together by public transport to reduce their global travel cost. Motivated by this, Shang et al. [33] study the Collective Travel

Planning (CTP) query that finds the lowest-cost route connecting multiple sources and a destination, via at most k meeting locations. Another related work proposed by Li et al. [75] is optimal multi-meeting-point route (OMMPR) query: The inputs are a road network G, a source location s, a target location t, and a set of query locations. The query aims at finding the best route starting from s to t such that the weighted average cost is minimized. The final cost is the weighted average between the cost of the route and the total cost of the shortest paths from every query location to the route. The subsequent work [76] aims to recommend a suitable route and stops of a vehicle for a group of users intending to travel collectively, such that the aggregate cost of the individual traveler's paths and the shared route is minimal.

3.3. Route planning beyond individual trip query

The objective of the problem in this area is to plan some distinct driving routes (or shared route) for multiple users to meet some particular constraints. Given a set of user queries, the output of this problem is a collection of individual routes (or a shared route). To the best of our knowledge, existing studies mainly investigate the problem on this topic for two aspects: group based optimal route planning and global based optimal route planning.

3.3.1. Group based optimal route planning

Hashem et al. [77] are the first to propose a novel group trip planning (GTP) query: the group has an interest to minimize the total travel distance for all members. Formally, for a set of user source-destination pairs in a group and different types of data points (e.g., a movie theater versus a restaurant), a GTP query finds a shared route that can be traversed by all the users while minimizing the total distance traveled in this group. A set of pruning techniques are developed to enhance algorithm efficiency to eliminate invalid data points. The subsequent work [78] develops both optimal and approximation algorithms for (top-k) GTP queries for both Euclidean space and road networks.

3.3.2. Global route planning

Different from group based optimal route, global route planning generates a distinct route for each query such that the total cost (or revenue) of all the users is minimal (resp., maximum). Lim et al. [79] attempt to find the best route combination such that minimizes traffic congestion and the total cost of all users. The authors propose a distributed method to find paths by introducing a probabilistic path choice achieving global goals.

Recall that the MSR problem we discussed in section 3.2.1, the work by Ye et al. [72] is the first to study the multiple mobile sequential recommendation problem (MMSR) that generates optimal routes for a group of taxis with different locations. It recommends k optimal driving routes for multiple taxi drivers in a way such that the multi-user potential travel distance before the successful pick-up of each route is minimized. Two efficient methods, PSAD and PSAD-M, are developed for solving the MMSR problem by ganging parallel computing and simulated annealing. Another related work is route planning in ride-sharing services, Zeng et al. [80,81] find a route for each vehicle such that the total revenue of the platform is maximized. Zeng et al. [80] consider both the total revenue (i.e., the objective) and the travel cost (i.e., the shortest travel time constraint). Chen et al. [82] studies the route matching problem that finds a global optimal matching for a group of trip queries. Li et al. [20] studies the global optimal route (GOR) problem: Given a traffic-aware road network and a set of trip queries, the authors generate a travel route for each query such that the sum of travel time for all queries is minimized. To address GOR problem, a greedy algorithm and a refining algorithm are developed, and some pruning strategies are established to further enhance efficiency.

4. Keyword-aware route planning

The problem of keyword-aware route planning has been extensively studied in recent years. Existing studies in this area can be classified into two categories: keyword-aware route planning based on exact matching (e.g., [4,34,35]), keyword-aware route planning based on approximate matching (e.g., [83–87]). Roughly speaking, approximate matching is realistic extensions of exact matching in cases of miss-spelling or fuzzy user-defined conditions.

4.1. Keyword-aware route planning based on exact matching

Trip Planning Query (TPQ). In the field of route planning, the work by Li et al. [34] is one of the pioneering studies to introduce a new query called Trip Planning Query (TPQ) in spatial databases, where each spatial object has a location and a keyword, and the objects are indexed by an R-tree. A TPQ query consists of a start location s, a destination d, and a set of keywords Ψ. It aims at finding an optimal route (in terms of distance, traffic road condition, etc.) from s to d, passing through at least one object from each keyword in Ψ. It is shown that TPQ can be reduced from the Traveling Salesman (TSP) problem [88], which is NP-hard. This study [34] offers a greedy algorithm and an integer programming algorithm to handle TPQ based on the triangle inequality property of metric space. An approximation algorithm in [34] is also designed to handle the exponential growth of the problem's search space. Another work [89,90] introduce a modification of TPQ. The objective of [89] is to find an optimal route that maximizes user happiness under some constraints: all the keywords can be visited in this route, and the route can be completed within the user time budget with a probability no less than a user-defined threshold. The authors solve this problem by using the information and constraints to prune unpromising candidate trips. The proposed algorithm consists of an offline step and an online step. Further, [90] suggests to combine several preference functions to measure a candidate route, which attempts to balance the weights of keyword coverage and route length. To achieve high efficiency, the authors propose to drastically prune the search space through a goal-directed search.

Optimal Sequenced Route Query (OSR). An extension [35] defines a variant problem of TPQ problem, called optimal sequenced route (OSR) query. In OSR, a total order on the keywords Ψ is imposed and only the starting location s is specified. The goal is to find a shortest route starting from s and passing through a number of typed locations in a particular order. To tackle the problem, this study presents two elegant exact algorithms: L-LORD and R-LORD. Briefly speaking, L-LORD is a threshold-based algorithm for vector spaces, while R-LORD is an extension of L-LORD that uses R-tree to examine the threshold values more efficiently. Another related work [91] decomposes a general optimal route query to multiple total-order queries and processes them individually (e.g., using R-LORD). Compared to OSR, TPQ eliminates the sequence order. As a consequence, OSR algorithms are able to address TPQ queries in theory.

In practice, users prefer to specify a partial or arbitrary order rather than a total order on the intended locations [92–94]. Particularly, Chen et al. [92,93] attempt to address the multi-rule partial sequenced route (MRPSR) query, which is a unified query of TPQ and OSR. To enable efficient trip planning with user defined traveling rules, the authors develop three heuristic algorithms to answer MRPSR. Li et al. [94] point out that user may want to visit POIs in arbitrary order and introduce two different methodologies, namely forward search and backward search to find an optimal route. Subsequently, some studies [3,95,96] take into account several extra constraints. In particular, Sharifzadeh et al. [95] present a variant of OSR by additionally considering temporal constraints. A pre-computation algorithm is developed for processing OSR queries in vector spaces. Related work [3,96] attach user preferences to POIs where the weights of POIs depend on user-specific preferences. Specifically, Zeng et al. [3] target to find an optimal route that covers as many keywords as possible while meeting the constraints. A keyword coverage function is established to evaluate a route. To further satisfy various user preferences, Li et al. [96] investigate a new route search query called Keyword-aware Dominant Routes (KDR) query, which finds all possible routes that cover a set of keywords under a specified budget constraint. An exact algorithm and a heuristic algorithm are performed to answer a KDR query.

Besides, another extension is top-k OSR query (KOSR) [13,97]. In this area, Liu et al. [97] propose to find the top-k optimal routes from a source to a destination, which must visit a number of locations with specific keywords in a predefined order. The study develops two algorithms to efficiently answer this kind of query. To model a general route diversity requirement for user's quantity and variety trade-off, Liang et al. [13] models the personalized diversity requirements, and answers the query through indices for retrieving relevant POIs in both feature and route spaces. Some pruning strategies are also employed in [13].

Keyword-aware Optimal Route Search (KOR). Cao et al. [4] focus on the problem of keyword-aware optimal route query, denoted by KOR, which aims at finding an optimal route such that it covers a set of user-specified keywords, a specified budget constraint is satisfied, and an objective score of the route is optimal. Unlike TPQ [34] and OSR [35] that merely support keyword limit, KOR takes into account both the user preferences and the budget constraint in route search. Specifically, KOR includes an additional constraint, namely budget constraint, and thus is more expressive. Likewise, KOR does not restrict the order in which keywords can be accessed. In [4], two approximate algorithms and a greedy algorithm are developed to answer KOR. Another related but different work [3] labels a location by weighted keywords, whilst KOR treats each keyword equally and formulates the objective function as an accumulative function.

Interactive Route Search (IRS). This category of existing studies [98–102] report the route in an interactive fashion instead of giving a complete route directly. Specifically, in each step, only the next location of route is provided to the user, and after each visit of a location, the user provides a feedback specifying whether the location is indeed relevant to user-specific requirements. On this topic, Kanza et al. [98,99] address a different trip query on the spatial database: the length of the route should be smaller than a specified threshold while the total text relevant of the route is maximized. The two studies propose a greedy algorithm and a data mining-based approach to answer the query, respectively. The subsequent work [100] develops several heuristic algorithms for answering a similar query in an interactive way. Further, Levin et al. [101] consider an additional constraint, namely keywords access order, and develop an approximate algorithm for finding the shortest or the maximum probability route satisfying the user requirements. Besides, another work [102] claims a similar problem in which the users give feedback for the already suggested locations, and the itineraries are constructed interactively based on the users' preference and time budget.

Group Trip Planning (GTP). People sometimes tend to form groups to travel collectively due to their social activities. The objective of the problem in this area is to plan a shared route or multiple routes that can be traveled by all users with an optimization goal. Specifically, each user provides a starting location, a destination location, a set of keywords and some constraints (e.g., keyword access order and keyword preference), the goal is to find a route or multiple routes that can be traversed by all the users while minimizing the total distance or maximizing the total preference in the group. Hashem et al. [77] are the first to propose the GTP query, which processes GTP query through finding the data points with a single traversal on the database. A set of pruning techniques are developed to eliminate the data points from the search while evaluating GTP queries. The subsequent work [78] proposes to find a route that minimizes the total (or maximum) distance within the group. The authors devise both optimal and approximate algorithms to answer GTP query for both Euclidean space and road networks, and develop novel techniques for refining the POIs search space based on geometric properties of ellipses. Besides, some studies [103,104] propose to maximize the group's preference within individual constraints in the route. In particular, Fan et al. [103] consider that each user in the group has his or her own preference for the locations and has user-specific limits (e.g., distance, time). An approximate algorithm and an exact algorithm are developed. Similar to [103], another work [104] develops an exact algorithm to find an optimal route by exploring a finite route, and an approximate algorithm to speed up the dynamic programming process used for the exact solution.

Other related issues. Traffic-aware route search (TARS) [105] takes into account variations in the travel speed due to changes in traffic conditions. The authors develop two greedy algorithms and a heuristics algorithm to find the fastest route with an optimal departure time. To enable efficient route planning in indoor spaces, one work [106] tackles the problem of category aware multi-criteria route planning query by an approximation algorithm. The query returns the optimal route from an indoor source to an indoor destination that covers at least one indoor location from each given keyword while minimizing the total cost. Another research [107] proposes an indoor top-k keyword-aware routing query, aiming at find top-k routes that do not exceed a given distance constraint but have optimal ranking scores integrating keyword relevance and spatial distance. The authors design two search algorithms with different routing expansions to answer this query.

4.2. Keyword-aware route planning based on approximate matching

Basically, the input of the problems in this category consists of a starting location, a destination location and a set of (keywords, threshold) pairs. The existing studies mainly focus on utilizing a similarity threshold to determine whether a keyword associated with location matches the keyword in a query or not. A representative work [83] proposes the multi-approximate-keyword routing (MAKR) query. The objective of MAKR query is to find the shortest route that covers at least one matching point per given keyword while satisfying the textual similarity constraints. In particular, the matching point is required to have an edit distance smaller than a pre-defined threshold. Another study [84] explores the clue-based route search (CRS) problem, which enables a user to define clues on keywords and spatial relationships. CRS aims at returning a route starting at source location such that it cover a sequence of POIs w.r.t. the clues and the network distances between two contagious matched vertices are close to the corresponding user-defined distance such that the user's search intention is satisfied. The authors propose a greedy algorithm, a dynamic programming algorithm as baselines and a branch-and-bound algorithm to boost performance.

A branch of researches [85,86]considers arbitrary text descriptions as keywords for personalized requirements. Particularly, Wen

Table 2
Overview of some keyword-aware route planning queries.

Queries	Input	Output	Proposal
TPQ	a source location *s*, a destination *d*, a set of keywords Ψ	An optimal route from *s* to *d*, passing through at least one object from each keyword in Ψ	A_{NN} A_{MD} ILP
OSR	a start location s, a set of keywords Ψ, a total order Δ on Ψ	A shortest route starting from s and passing through at least one object from each keyword in Ψ in a total order Δ	L-LORD R-ROLD
MRPSR	a start location s, a set of keywords Ψ, a partial order Δ on Ψ	A shortest route starting from s and passing through at least one object from each keyword in Ψ in a partial order Δ	NNPSR NNPSR-LORD ASPSR
KOR	a start location s, a destination d, a set of keywords Ψ, a specified budget constraint Δ	An optimal route such that it covers a set of user-specified keywords, a specified budget constraint is specified	OSScaling BucketBound Greedy
GTP	**Provided by each user:** a starting location s, a destination location d, a set of keywords Ψ, some constraints Δ on Ψ	A route or multiple routes that can be traversed by all the users while minimizing the total distance or maximizing the total preference in the group	GTP-IA GTP-HA
MARK	a starting location s, a destination location d, a set of keywords, threshold value pairs Ψ	A route with the shortest length, such that it covers at least one matching object per given keyword while satisfying string similarity constraints	PER-full PER-partial A_{LMP} A_{GMP}

et al. [85] develop a keyword-aware representative travel route framework that takes advantage of knowledge extracted from users' historical mobility records and social interactions. Li et al. [86] propose a Bounded-Cost Informative Route (BCIR) query to retrieve the routes that are textually-relevant to query keywords while satisfying the travel-cost constraint. To answer the BCIR query efficiently, this study presents an exact algorithm with effective pruning techniques and two approximate algorithms. Table 2 summarizes the input, output and corresponding algorithms of some representative keyword-based route planning queries. We use *s* and *d* to denote the source location and destination location of a query, respectively. Ψ denotes a set of keywords. For MARK, $\Psi = \{(\sigma_1, \tau_1), \ldots, (\sigma_k, \tau_k)\}$, where σ_i is the *ith* query keyword and τ_i is the desired edit distance for deciding whether a keyword from an object matches with σ_i.

5. Conclusions and future directions

In this survey, we investigate and summarize existing studies on route search and planning. We introduce existing works from three aspects: routing data management, location-based route planning and keyword-aware route planning. For the first category, we briefly introduce the data formulation, route search, and trajectory search, respectively. Some primary terminologies are defined in this part. In addition, we make a comprehensive comparison among some popular trajectory similarity measures on their robustness especially when they process low-quality trajectories. As for location-based route planning, three types of related studies are investigated in this area: source-destination based route planning, multi-location based route planning and multi-user based route planning, respectively. According to the matching settings among keywords, we classify existing studies on keyword-aware route planning into two categories: exact matching and approximate matching based route planning. We summarize some representative queries according to the inputs, outputs and algorithms.

Though route search and planning have been extensively studied, we believe that there are still a host of problems to be addressed or approaches to be improved in this area. For the trajectory similarity measures, deep learning based approaches are robust to low-quality trajectories, and are capable of achieving high efficiency in trajectory match. However, existing studies on this topic do not take temporal components into account, making them unable to answer similarity-based queries in spatio-temporal databases, because both spatial information and temporal information are indispensable. As a result, it will be of great interest to extend previous deep learning based model by additionally consider temporal components. For the location-based route planning, there are few studies on global route planning for massive scale of trip queries, which aims to find the best route combination such that minimizes traffic congestions and the total cost of all users. Thus, it is important to enable such route planning for massive scale of trip queries or a stream of trip queries.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] L. Chen, S. Shang, C. Yang, J. Li, Spatial keyword search: a survey, GeoInformatica 24 (1) (2020) 85–106.
[2] S. Shang, L. Chen, Z. Wei, C.S. Jensen, K. Zheng, P. Kalnis, Parallel trajectory similarity joins in spatial networks, VLDB J. 27 (3) (2018) 395–420.
[3] Y. Zeng, X. Chen, X. Cao, S. Qin, M. Cavazza, Y. Xiang, Optimal route search with the coverage of users' preferences, in: IJCAI, AAAI Press, 2015, pp. 2118–2124.
[4] X. Cao, L. Chen, G. Cong, X. Xiao, Keyword-aware optimal route search, Proc. VLDB Endow. 5 (11) (2012) 1136–1147.
[5] S. Wang, Z. Bao, J.S. Culpepper, G. Cong, A survey on trajectory data management, analytics, and learning, ACM Comput. Surv. 54 (2) (2021) 39.
[6] J.D. Mazimpaka, S. Timpf, Trajectory data mining: a review of methods and applications, J. Spat. Inf. Sci. 13 (1) (2016) 61–99.
[7] H. Su, S. Liu, B. Zheng, X. Zhou, K. Zheng, A survey of trajectory distance measures and performance evaluation, VLDB J. 29 (1) (2020) 3–32.

[8] A.R. Mahmood, S. Punni, W.G. Aref, Spatio-temporal access methods: a survey (2010 - 2017), GeoInformatica 23 (1) (2009) 1–36.
[9] A.R. Mahmood, W.G. Aref, Scalable Processing of Spatial-Keyword Queries, Synthesis Lectures on Data Management, Morgan & Claypool Publishers, 2019.
[10] X. Cao, L. Chen, G. Cong, C.S. Jensen, Q. Qu, A. Skovsgaard, D. Wu, M.L. Yiu, Spatial keyword querying, in: Conceptual Modeling, Springer, 2012, pp. 16–29.
[11] L. Chen, G. Cong, C.S. Jensen, D. Wu, Spatial keyword query processing: an experimental evaluation, Proc. VLDB Endow. 6 (3) (2013) 217–228.
[12] Z. Chen, L. Chen, G. Cong, C.S. Jensen, Location-and keyword-based querying of geo-textual data: a survey, VLDB J. (2021) 1–38.
[13] H. Liang, K. Wang, Top-k route search through submodularity modeling of recurrent POI features, in: SIGIR, ACM, 2018, pp. 545–554.
[14] S.A. Pedersen, B. Yang, C.S. Jensen, Fast stochastic routing under time-varying uncertainty, VLDB J. 29 (4) (2020) 819–839.
[15] S.A. Pedersen, B. Yang, C.S. Jensen, Anytime stochastic routing with hybrid learning, Proc. VLDB Endow. 13 (9) (2020) 1555–1567.
[16] C. Guo, B. Yang, J. Hu, C.S. Jensen, L. Chen, Context-aware, preference-based vehicle routing, VLDB J. 29 (5) (2020) 1149–1170.
[17] C. Yang, L. Chen, H. Wang, S. Shang, Towards efficient selection of activity trajectories based on diversity and coverage, in: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, AAAI Press, 2021, pp. 689–696.
[18] J. Xu, L. Guo, Z. Ding, X. Sun, C. Liu, Traffic aware route planning in dynamic road networks, in: Database Systems for Advanced Applications, in: Lecture Notes in Computer Science, vol. 7238, Springer, 2012, pp. 576–591.
[19] D. Wilkie, J.P. van den Berg, M.C. Lin, D. Manocha, Self-aware traffic route planning, in: Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7–11, 2011, AAAI Press, 2011.
[20] K. Li, L. Chen, S. Shang, Towards alleviating traffic congestion: optimal route planning for massive-scale trips, in: IJCAI, 2020, pp. 3400–3406.
[21] D. Wilkie, C. Baykal, M.C. Lin, Participatory route planning, in: SIGSPATIAL/GIS, ACM, 2014, pp. 213–222.
[22] B. Ding, J.X. Yu, L. Qin, Finding time-dependent shortest paths over large graphs, in: EDBT, in: ACM International Conference Proceeding Series, vol. 261, ACM, 2008, pp. 205–216.
[23] X. Cai, T. Kloks, C.K. Wong, Time-varying shortest path problems with constraints, Networks 29 (3) (1997) 141–150.
[24] E. Kanoulas, Y. Du, T. Xia, D. Zhang, Finding fastest paths on a road network with speed patterns, in: ICDE, IEEE Computer Society, 2006, p. 10.
[25] P. Narváez, K. Siu, H. Tzeng, New dynamic SPT algorithm based on a ball-and-string model, IEEE/ACM Trans. Netw. 9 (6) (2001) 706–718.
[26] Y. Wang, G. Li, N. Tang, Querying shortest paths on time dependent road networks, Proc. VLDB Endow. 12 (11) (2019) 1249–1261.
[27] S. Shang, J. Liu, K. Zheng, H. Lu, T.B. Pedersen, J. Wen, Planning unobstructed paths in traffic-aware spatial networks, GeoInformatica 19 (4) (2015) 723–746.
[28] S. Shang, D. Guo, J. Liu, J. Wen, Prediction-based unobstructed route planning, Neurocomputing 213 (2016) 147–154.
[29] S. Ranu, D. P, A.D. Telang, P. Deshpande, S. Raghavan, Indexing and matching trajectories under inconsistent sampling rates, in: 31st IEEE International Conference on Data Engineering, ICDE 2015, Seoul, South Korea, April 13–17, 2015, IEEE Computer Society, 2015.
[30] H. Su, K. Zheng, H. Wang, J. Huang, X. Zhou, Calibrating trajectory data for similarity-based analysis, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2013, New York, NY, USA, June 22–27, 2013, ACM, 2013, pp. 833–844.
[31] S. Brakatsoulas, D. Pfoser, R. Salas, C. Wenk, On map-matching vehicle tracking data, in: Proceedings of the 31st International Conference on Very Large Data Bases, ACM, 2005, pp. 853–864.
[32] Y. Ge, H. Xiong, A. Tuzhilin, K. Xiao, M. Gruteser, M.J. Pazzani, An energy-efficient mobile recommender system, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 25–28, 2010, ACM, 2010.
[33] S. Shang, L. Chen, Z. Wei, C.S. Jensen, J. Wen, P. Kalnis, Collective travel planning in spatial networks, in: ICDE, IEEE Computer Society, 2017, pp. 59–60.
[34] F. Li, D. Cheng, M. Hadjieleftheriou, G. Kollios, S. Teng, On trip planning queries in spatial databases, in: SSTD, in: Lecture Notes in Computer Science, vol. 3633, Springer, 2005, pp. 273–290.
[35] M. Sharifzadeh, M.R. Kolahdouzan, C. Shahabi, The optimal sequenced route query, VLDB J. 17 (4) (2008) 765–787.
[36] R. Agrawal, C. Faloutsos, A.N. Swami, Efficient similarity search in sequence databases, in: FODO, in: Lecture Notes in Computer Science, vol. 730, Springer, 1993, pp. 69–84.
[37] B. Yi, H.V. Jagadish, C. Faloutsos, Efficient retrieval of similar time sequences under time warping, in: Proceedings of the Fourteenth International Conference on Data Engineering, Orlando, Florida, USA, February 23–27, 1998, IEEE Computer Society, 1998.
[38] M. Vlachos, D. Gunopulos, G. Kollios, Discovering similar multidimensional trajectories, in: Proceedings of the 18th International Conference on Data Engineering, San Jose, CA, USA, February 26 – March 1, 2002, IEEE Computer Society, 2002.
[39] L. Chen, R.T. Ng, On the marriage of lp-norms and edit distance, in: (e)Proceedings of the Thirtieth International Conference on Very Large Data Bases, VLDB 2004, Toronto, Canada, August 31 – September 3 2004, Morgan Kaufmann, 2004, pp. 792–803.
[40] L. Chen, M.T. Özsu, V. Oria, Robust and fast similarity search for moving object trajectories, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, Baltimore, Maryland, USA, June 14–16, 2005, ACM, 2005.
[41] C. Perng, H. Wang, S.R. Zhang, D.S.P. Jr., Landmarks: a new model for similarity-based pattern querying in time series databases, in: Proceedings of the 16th International Conference on Data Engineering, San Diego, California, USA, February 28 – March 3, 2000, IEEE Computer Society, 2000.
[42] Y. Chen, M.A. Nascimento, B.C. Ooi, A.K.H. Tung Spade, On shape-based pattern detection in streaming time series, in: R. Chirkova, A. Dogac, M.T. Özsu, T.K. Sellis (Eds.), Proceedings of the 23rd International Conference on Data Engineering, ICDE 2007, The Marmara Hotel, Istanbul, Turkey, April 15–20, 2007, IEEE Computer Society, 2007, pp. 786–795.
[43] X. Li, K. Zhao, G. Cong, C.S. Jensen, W. Wei, Deep representation learning for trajectory similarity computation, in: 34th IEEE International Conference on Data Engineering, ICDE 2018, Paris, France, April 16–19, IEEE Computer Society, 2018, pp. 617–628.
[44] D. Yao, G. Cong, C. Zhang, J. Bi, Computing trajectory similarity in linear time: a generic seed-guided neural metric learning approach, in: 35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, April 8–11, 2019, IEEE, 2019, pp. 1358–1369.
[45] Z. Chen, H.T. Shen, X. Zhou, Y. Zheng, X. Xie, Searching trajectories by locations: an efficiency study, in: SIGMOD Conference, ACM, 2010, pp. 255–266.
[46] S. Shang, R. Ding, B. Yuan, K. Xie, K. Zheng, P. Kalnis, User oriented trajectory search for trip recommendation, in: 15th International Conference on Extending Database Technology, Proceedings, EDBT '12, Berlin, Germany, March 27–30, 2012, ACM, 2012, pp. 156–167.
[47] K. Zheng, S. Shang, N.J. Yuan, Y. Yang, Towards efficient search for activity trajectories, in: C.S. Jensen, C.M. Jermaine, X. Zhou (Eds.), 29th IEEE International Conference on Data Engineering, ICDE 2013, Brisbane, Australia, April 8–12, 2013, IEEE Computer Society, 2013, pp. 230–241.
[48] S. Shang, R. Ding, K. Zheng, C.S. Jensen, P. Kalnis, X. Zhou, Personalized trajectory matching in spatial networks, VLDB J. 23 (3) (2014) 449–468.
[49] S. Shang, L. Chen, Z. Wei, C.S. Jensen, K. Zheng, P. Kalnis, Trajectory similarity join in spatial networks, Proc. VLDB Endow. 10 (11) (2017) 1178–1189.
[50] S. Shang, L. Chen, C.S. Jensen, J. Wen, P. Kalnis, Searching trajectories by regions of interest, in: 34th IEEE International Conference on Data Engineering, ICDE 2018, Paris, France, April 16–19, 2018, IEEE Computer Society, 2018, pp. 1741–1742.
[51] S. Shang, L. Chen, K. Zheng, C.S. Jensen, Z. Wei, P. Kalnis, Parallel trajectory-to-location join, IEEE Trans. Knowl. Data Eng. 31 (6) (2019) 1194–1207.
[52] Z. Chen, H.T. Shen, X. Zhou, Discovering popular routes from trajectories, in: S. Abiteboul, K. Böhm, C. Koch, K. Tan (Eds.), Proceedings of the 27th International Conference on Data Engineering, ICDE 2011, April 11–16, 2011, Hannover, Germany, IEEE Computer Society, 2011, pp. 900–911.
[53] J. Dai, B. Yang, C. Guo, Z. Ding, Personalized route recommendation using big trajectory data, in: ICDE, IEEE Computer Society, 2015, pp. 543–554.
[54] C. Guo, B. Yang, J. Hu, C.S. Jensen, Learning to route with sparse trajectory sets, in: 34th IEEE International Conference on Data Engineering, ICDE 2018, Paris, France, April 16–19, 2018, IEEE Computer Society, 2018, pp. 1073–1084.
[55] L. Chen, S. Shang, C.S. Jensen, B. Yao, Z. Zhang, L. Shao, Effective and efficient reuse of past travel behavior for route recommendation, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4–8, 2019, ACM, 2019, pp. 488–498.
[56] R. de Oliveira, E. Silva, G. Cui, S.M. Rahimi, X. Wang, Br^2: a travel behavioral approach to personalized route recommendation based on GPS trajectories, in: Proceedings of the Workshops of the EDBT/ICDT 2020 Joint Conference, Copenhagen, Denmark, March 30, 2020, in: CEUR Workshop Proceedings, vol. 2578, CEUR-WS.org, 2020.
[57] E.W. Dijkstra, A note on two problems in connexion with graphs, Numer. Math. 1 (1959) 269–271.
[58] M. Likhachev, D.I. Ferguson, G.J. Gordon, A. Stentz, S. Thrun, Anytime dynamic a*: an anytime, replanning algorithm, in: ICAPS, AAAI, 2005, pp. 262–271.
[59] N. Malviya, S. Madden, A. Bhattacharya, A continuous query system for dynamic route planning, in: ICDE, IEEE Computer Society, 2011, pp. 792–803.
[60] B. Yang, C. Guo, Y. Ma, C.S. Jensen, Toward personalized, context-aware routing, VLDB J. 24 (2) (2015) 297–318.
[61] J. Letchner, J. Krumm, E. Horvitz, Trip router with individualized preferences (TRIP): incorporating personalization into route planning, in: AAAI, AAAI Press, 2006, pp. 1795–1800.
[62] A. Balteanu, G. Jossé, M. Schubert, Mining driving preferences in multi-cost networks, in: SSTD, in: Lecture Notes in Computer Science, vol. 8098, Springer, 2013, pp. 74–91.
[63] K. Deng, X. Zhou, H.T. Shen, Multi-source skyline query processing in road networks, in: ICDE, IEEE Computer Society, 2007, pp. 796–805.

[64] H. Kriegel, M. Renz, M. Schubert, Route skyline queries: a multi-preference path planning approach, in: ICDE, IEEE Computer Society, 2010, pp. 261–272.
[65] S. Lim, H. Balakrishnan, D. Gifford, S. Madden, D. Rus, Stochastic motion planning and applications to traffic, Int. J. Robot. Res. 30 (6) (2011) 699–712.
[66] B. Yang, C. Guo, C.S. Jensen, M. Kaul, S. Shang, Stochastic skyline route planning under time-varying uncertainty, in: ICDE, IEEE Computer Society, 2014, pp. 136–147.
[67] B. Yang, J. Dai, C. Guo, C.S. Jensen, J. Hu, PACE: a path-centric paradigm for stochastic path finding, VLDB J. 27 (2) (2018) 153–178.
[68] M. Hua, J. Pei, Probabilistic path queries in road networks: traffic uncertainty aware path selection, in: EDBT, in: ACM International Conference Proceeding Series, vol. 426, ACM, 2010, pp. 347–358.
[69] J. Huang, X. Huangfu, H. Sun, H. Li, P. Zhao, H. Cheng, Q. Song, Backward path growth for efficient mobile sequential recommendation, IEEE Trans. Knowl. Data Eng. 27 (1) (2015) 46–60.
[70] Z. Ye, K. Xiao, Y. Deng, A unified theory of the mobile sequential recommendation problem, in: IEEE International Conference on Data Mining, ICDM, 2018, Singapore, November 17–20, 2018, IEEE Computer Society, 2018, pp. 1380–1385.
[71] Z. Ye, K. Xiao, Y. Ge, Y. Deng, Applying simulated annealing and parallel computing to the mobile sequential recommendation, IEEE Trans. Knowl. Data Eng. 31 (2) (2019) 243–256.
[72] Z. Ye, L. Zhang, K. Xiao, W. Zhou, Y. Ge, Y. Deng, Multi-user mobile sequential recommendation: an efficient parallel computing paradigm, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19–23, 2018, ACM, 2018, pp. 2624–2633.
[73] S. Shang, H. Lu, T.B. Pedersen, X. Xie, Finding traffic-aware fastest paths in spatial networks, in: SSTD, in: Lecture Notes in Computer Science, vol. 8098, Springer, 2013, pp. 128–145.
[74] L. Chen, S. Shang, T. Guo, Real-time route search by locations, in: AAAI, AAAI Press, 2020, pp. 574–581.
[75] R. Li, L. Qin, J.X. Yu, R. Mao, Optimal multi-meeting-point route search, IEEE Trans. Knowl. Data Eng. 28 (3) (2016) 770–784.
[76] R.M. Reza, M.E. Ali, M.A. Cheema, The optimal route and stops for a group of users in a road network, in: Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS 2017, Redondo Beach, CA, USA, November 7–10, 2017, ACM, 2017, 4.
[77] T. Hashem, T. Hashem, M.E. Ali, L. Kulik, Group trip planning queries in spatial databases, in: Advances in Spatial and Temporal Databases - 13th International Symposium, Proceedings, SSTD 2013, Munich, Germany, August 21–23, 2013, in: Lecture Notes in Computer Science, vol. 8098, Springer, 2013, pp. 259–276.
[78] T. Hashem, S. Barua, M.E. Ali, L. Kulik, E. Tanin, Efficient computation of trips with friends and families, in: Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19–23, 2015, ACM, 2015, pp. 931–940.
[79] S. Lim, D. Rus, Stochastic distributed multi-agent planning and applications to traffic, in: ICRA, IEEE, 2012, pp. 2873–2879.
[80] Y. Zeng, Y. Tong, Y. Song, L. Chen, The simpler the better: an indexing approach for shared-route planning queries, Proc. VLDB Endow. 13 (13) (2020) 3517–3530.
[81] L. Zheng, L. Chen, J. Ye, Order dispatch in price-aware ridesharing, Proc. VLDB Endow. 11 (8) (2018) 853–865.
[82] L. Chen, S. Shang, B. Yao, J. Li, Pay your trip for traffic congestion: dynamic pricing in traffic-aware road networks, in: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, AAAI Press, 2020, pp. 582–589.
[83] B. Yao, M. Tang, F. Li, Multi-approximate-keyword routing in GIS data, in: GIS, ACM, 2011, pp. 201–210.
[84] B. Zheng, H. Su, W. Hua, K. Zheng, X. Zhou, G. Li, Efficient clue-based route search on road networks, IEEE Trans. Knowl. Data Eng. 29 (9) (2017) 1846–1859.
[85] Y.T. Wen, J. Yeo, W. Peng, S. Hwang, Efficient keyword-aware representative travel route recommendation, IEEE Trans. Knowl. Data Eng. 29 (8) (2017) 1639–1652.
[86] W. Li, J. Cao, J. Guan, M.L. Yiu, S. Zhou, Efficient retrieval of bounded-cost informative routes, in: ICDE, IEEE Computer Society, 2018, pp. 1811–1812.
[87] G. Jossé, K.A. Schmid, M. Schubert, Probabilistic resource route queries with reappearance, in: Proceedings of the 18th International Conference on Extending Database Technology, EDBT 2015, Brussels, Belgium, March 23–27, 2015, OpenProceedings.org, 2015, pp. 445–456.
[88] N. Christofides, Worst-Case Analysis of a New Heuristic for the Traveling Salesman Problem, 1976.
[89] C. Zhang, H. Liang, K. Wang, J. Sun, Personalized trip recommendation with POI availability and uncertain traveling time, in: Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19–23, 2015, ACM, 2015, pp. 911–920.
[90] P. Yawalkar, S. Ranu, Route recommendations on road networks for arbitrary user preference functions, in: ICDE, IEEE, 2019, pp. 602–613.
[91] X. Ma, S. Shekhar, H. Xiong, P. Zhang, Exploiting a page-level upper bound for multi-type nearest neighbor queries, in: GIS, ACM, 2006, pp. 179–186.
[92] H. Chen, W. Ku, M. Sun, R. Zimmermann, The Multi-Rule Partial Sequenced Route Query, in: GIS, ACM, 2008, p. 10.
[93] H. Chen, W. Ku, M. Sun, R. Zimmermann, The partial sequenced route query with traveling rules in road networks, GeoInformatica 15 (3) (2011) 541–569.
[94] J. Li, Y.D. Yang, N. Mamoulis, Optimal route queries with arbitrary order constraints, IEEE Trans. Knowl. Data Eng. 25 (5) (2013) 1097–1110.
[95] M. Sharifzadeh, C. Shahabi, Processing optimal sequenced route queries using Voronoi diagrams, GeoInformatica 12 (4) (2008) 411–433.
[96] Y. Li, W. Yang, W. Dan, Z. Xie, Keyword-aware dominant route search for various user preferences, in: Database Systems for Advanced Applications - 20th International Conference, Proceedings, Part II, DASFAA 2015, Hanoi, Vietnam, April 20–23, 2015, in: Lecture Notes in Computer Science, vol. 9050, Springer, 2015, pp. 207–222.
[97] H. Liu, C. Jin, B. Yang, A. Zhou, Finding top-k optimal sequenced routes, in: ICDE, IEEE Computer Society, 2018, pp. 569–580.
[98] Y. Kanza, E. Safra, Y. Sagiv, Y. Doytsher, Heuristic algorithms for route-search queries over geographical data, in: GIS, ACM, 2008, p. 11.
[99] E.H. Lu, C. Lin, V.S. Tseng, Trip-mine: an efficient trip planning approach with travel time constraints, in: Mobile Data Management (1), IEEE Computer Society, 2011, pp. 152–161.
[100] Y. Kanza, R. Levin, E. Safra, Y. Sagiv, An interactive approach to route search, in: GIS, ACM, 2009, pp. 408–411.
[101] R. Levin, Y. Kanza, E. Safra, Y. Sagiv, Interactive route search in the presence of order constraints, Proc. VLDB Endow. 3 (1) (2010) 117–128.
[102] S.B. Roy, G. Das, S. Amer-Yahia, C. Yu, Interactive itinerary planning, in: ICDE, IEEE Computer Society, 2011, pp. 15–26.
[103] L. Fan, L. Bonomi, C. Shahabi, L. Xiong, Multi-user itinerary planning for optimal group preference, in: Advances in Spatial and Temporal Databases - 15th International Symposium, Proceedings, SSTD 2017, Arlington, VA, USA, August 21–23, 2017, in: Lecture Notes in Computer Science, vol. 10411, Springer, 2017, pp. 3–23.
[104] L. Fan, L. Bonomi, C. Shahabi, L. Xiong, Optimal group route query: finding itinerary for group of users in spatial databases, GeoInformatica 22 (4) (2018) 845–867.
[105] R. Levin, Y. Kanza, TARS: traffic-aware route search, GeoInformatica 18 (3) (2014) 461–500.
[106] C. Salgado, M.A. Cheema, D. Taniar, An efficient approximation algorithm for multi-criteria indoor route planning queries, in: SIGSPATIAL/GIS, ACM, 2018, pp. 448–451.
[107] Z. Feng, T. Liu, H. Li, H. Lu, L. Shou, J. Xu, Indoor top-k keyword-aware routing query, in: ICDE, IEEE, 2020, pp. 1213–1224.