Reverse nearest neighbourhood query based on road social networks

Yaoyu Liu[1], Shaopeng Wang[1](✉), Chunkai Feng[1] and Wei Guo[1]

[1]College of computer science, Inner Mongolia University, Hohhot 010021, Inner Mongolia, China
wangshaopeng1984@163.com

Abstract. With the increasing popularity of mobile devices that support spatial positioning, numerous location-based service (LBS) systems have been put into place and widely adopted by users of mobile devices. Reverse nearest neighbor (RNN) queries are essential supporting techniques in these systems. A new and useful variant of RNN queries has emerged recently, known as reverse nearest neighbourhood concept for road networks (RNNH-RN), to discover the neighbourhood that finds the query point is the nearest facilities among all other facilities. However, existing research has primarily focused on spatial queries, and to the best of our knowledge, there is no technique available for computing queries that incorporate social network information. The questions people currently ask about road networks are not applicable to road social networks directly. In this paper, we introduce the reverse nearest neighbourhood query based on road social networks(RNNH-RS), where a neighbourhood is a set of at least m objects, ensuring that the maximum road network distance between any objects is at most d, and the objects within the neighbourhood have at least k, familiar acquaintances. We validated the flexibility and effectiveness of the proposed query through experiments on a real-world road social network dataset.

Keywords: Road social network, Reverse nearest neighbor, Social influence, Spatial databases.

1 Introduction

Many location-based service (LBS) systems, such as Baidu Maps and Google Maps, have been adopted and widely embraced by mobile users due to the growing use of mobile devices with GPS technology.An increasing number of individuals are utilizing these services, as these services facilitate the easy sharing, recording, and enhancement of their lifestyles. In these services, spatial queries have played a significant role. The reverse nearest neighbourhood query (RNNH) proposed by Nasser Allheeib in 2021 is a significant research direction in spatial queries and is intended to find a group of neighborhoods that is influenced by the query point. A neighbourhood is defined as a collection of at least m POIs such that the euclidean distance between any pair of these objects is at most d. RNNH-RN is the RNNH in road networks, and get the result neighborhoods based on road networks rather than Euclidean space. The relationship between any member and the RNNH-RN result neighbourhood is scaled solely based on the road network distance. This method is inaccurate in a case such as that shown in Figure 1.

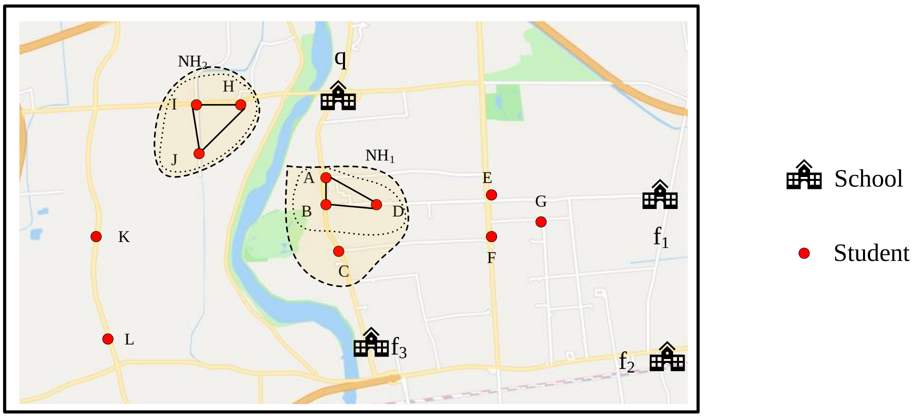


Fig. 1. A motivating example.

As shown in Fig.1, consider twelve students A to L shown as red circles, four schools q, f_1, f_2, f_3 , where each school tends to attract students for enrollment. The edge between two student represents social relation of them. For the q, the neighborhoods NH_1 and NH_2 are the outputs of RNNH-RN. All students in these two neighborhoods are more likely to go to school q. Let us consider the student C, the reason why this student chooses to go to school q instead of school f_3 is the distance of this student to the neighbourhood NH_1(the distance to the nearest student B inside the NH_1) is less than the distance to f_3. However, if there not exist any social relationship between student C and any member of the NH_1 (for example, they do not know each other, are not friends, not colleagues, or not in any similar situations),the student C may be influenced by f_3 more than NH_1(such as f_3 has better price or transportation), and chooses to go to school f_3.

This example motivates us to consider a novel type of query on RNNH, namely RNNH based on road- social networks (RNNH-RS), and to propose efficient processing algorithms. Specifically, the RNNH-RS incorporates social relation compared to the RNNH-RN: Given point p of the RNNH-RS result neighbourhood NH-RS, the road network distance between NH-RS and p should be small enough, and at least k members of the NH-RS have social relationships with p.

Below we summarize the contributions we make in this paper:

1. Introducing the idea of neighbourhoods in road social networks and formally outlining the Reverse nearest neighbourhood query based on road social networks (RNNH-RS).

2. We propose a basic algorithm to implement RNNH-RS.

3. We propose an optimization algorithm based on an efficient road network index GT-NVD and an early termination strategy to complete query processing.

4. Providing a comprehensive empirical study to offer insights into the effectiveness and efficiency of our proposed algorithms.

2 Related work

Nasser first proposed Reverse Nearest Neighbourhood query (RNNH), but it only considers queries in Euclidean space[1]. He proposed reverse nearest neighbourhood query on road networks, yet they don't account for social connectivity[2].Reverse query (RNN) finds interest points that have a given query point q as their nearest neighbor. The earliest paper proposing the Reverse Nearest Neighbor (RNN) query was published by F Korn in 2000, introducing an algorithm based on the R-tree to solve the RNN query problem [4]. Paper [5] presents an efficient method to handle RNN queries in arbitrary-dimensional spaces and provides performance analysis and experimental results to validate its effectiveness. It proposes an algorithm based on indexing and pruning techniques to accelerate the execution of RNN queries [6]. The paper [7] investigates efficient algorithms for RNN queries with arbitrary shape preferences. It proposes a method based on spatial partitioning and indexing techniques to handle RNN queries with complex shape preferences. The paper [8], published in 2007, introduces a novel query type known as reverse top-k query, along with the concept and algorithm of the query. Some existing works have also proposed methods for pruning top-k queries[9][10][11][12].

Geosocial querying is an application method that combines geographic information with social interaction. It allows users to search, share, and communicate information about specific locations or geographic places on a map, typically through mobile applications or online platforms[13] [14][15].The paper [16] designs a time-constrained incremental personalized proximity search by considering spatial, social, and temporal information to retrieve cohesive shared groups. The paper [17] investigates the problem of computing radius-bounded k-core, aiming to find communities that satisfy social and spatial constraints. The paper [18] introduces keyword queries on geosocial top-k road networks and geosocial skyline keyword queries. Some studies focus on finding combinations of relationships that are closely related[19][20][21] .

3 Preliminaries

In this paper, the road network is an undirected weighted graph $G_r = (V_r, E_r)$, where V_r is a set of vertices, and $E_r = \{(v_r^i, v_r^j) | v_r^i, v_r^j \in V_r \wedge v_r^i \neq v_r^j\}$ is a set of edges. Vertex $v_r^i \epsilon V_r$ represents an intersection or the endpoint of a road, and each edge $e_r = (v_r^i, v_r^j) \in E_r$ represents a road segment that connects vertices v_r^i and v_r^j and each edge $e_r = (v_r^i, v_r^j)$ has a weight (e.g., distance), which is a positive value. Given a start vertex v_r^i and a destination vertex v_r^j, the sum of weights of individual edges on the shortest path is referred to as the road network distance between v_r^i and v_r^j, denoted as

$dist(v_r^i, v_r^j)$. A set of points of users $P = \{p_1, p_2, \dots, p_n\}$, a set of facilities $F = \{f_1, f_2, \dots, f_n\}$, a query facility $q \in F$ reside in G_r.

Table 1. Symbols and descriptions

Symbol	Meaning
$G_r(G_s)$	Road network (social network)
$dist(v_r^i, v_r^j)$	The network distance of the shortest path between v_r^i and v_r^j
F，P	A set of facility points, a set of user points
d	The road network distance between a member point p_i and the nearest member point p_j in the group does not exceed d
m	Number of users in the neighborhood
k	Each user in the neighborhood can have up to k unfamiliar people
NVC_q	Voronoi unit of query point q
$d_{RSH}(p_i, NH - RS)$	Minimum distance between point p_i and neighbourhood $NH - RS$
$NH - RS_i$	Neighborhood composed of a set of points
$vlist$	$Set\ of\ candidate\ user\ points$

Social network is modeled as an unweighted and undirected graph$G_s = (V_s, E_s, L_s)$, where V_s is the set of vertices (representing users), $E_s \subseteq V_s \times V_s$ is the set of edges (i.e., social relationships), and L_s is a mapping set defined on the vertices v_s in V_s. $L_s(v_s)$ specifies the attributes of v_s (e.g., location). In our definition, $L_s(v_s)$ provides a mapping of each user's location in the road network.

A road-social network is a pair of graphs $G = (G_r，G_s)$, where G_r represents the road network and G_s represents the social network. Each vertex $v_s^i \in G_s$ is associated with a vertex v_r^i or a spacial point sp_i in G_r, indicating that the user v_s^i is currently in location v_r^i or sp_i, i.e., $L_s(v_s^i) = v_r^i\ or\ sp_i$.

In the road social network, there are two types of objects with geographic and social labels, namely, the set of facility points F and the set of user points P. The geographic locations of these user points and facility points correspond to nodes in G_r, and the social relationships of user points correspond to edges in G_s.

Definition 1. Network Voronoi cell of query q: Given a graph G,a set of facility points F, a query facility point $q \in F$, and a set of user points P, n Network Voronoi cell of q, denoted as NVC_q, consists of pi ($pi \in P$) which considers the query facility q as the nearest facility based on the road network distance (e.g., shortest-path).i.e., $\forall pi \in NVC_q，dist(pi, q) \leq dist(pi, f), \forall f \in F \backslash q$.

We divide the road network graph into various small regions; each region contains a set of road segments that share the nearest facility points. This structure is referred to as the Network Voronoi Diagram (NVD). In the NVD, these small regions are called network Voronoi cells, and the facility points are generating points. Consider the dataset given in Figure 2. The network Voronoi cell for query point q is $\{p_4, p_5, p_6, p_{11}, p_{12}, p_{13}, p_{14}\}$. Given its structure and definition, several fundamental properties of NVD can be derived. First, the Voronoi diagram has many network Voronoi cells. Each network Voronoi cell has a generating point located at the center of the network Voronoi cell. Considering other properties of the Voronoi diagram, if point p_i is in the network Voronoi cell of q, then the distance from point p_i to point q is not greater than the distance from point p_i to other network Voronoi cells. For more properties, the distance from any point on the edge of the network Voronoi cell to the query point is equal to the distance to another generating point adjacent to the network Voronoi cell.

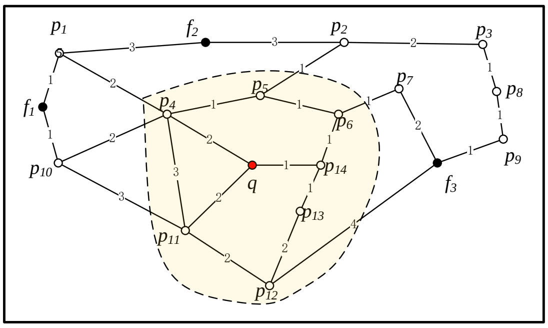


Figure. 2. Voronoi diagram of query point q

Definition 2. Road-Social Neighbourhood: Given graphs G, three parameters m ,d and k, a Road-Social neighbourhood refers to a region consisting of at least m member points of G_s where (1) the road networks distance between a member point p_i and the nearest member point p_j in the region does not exceed d ;(2) at least $k(k <= m)$ member points have social relationships with p_i . The road-social neighbourhood is denoted by $NH - RS(d, m, k)$.

The neighbourhood $NH - RS(d, m, k) = \{p_1, p_2, \ldots, p_m\} \in P$ represents a neighbourhood with m members and a distance parameter of d, where d represents the maximum road distance between the member points of $NH - RS(d, m, k)$, k represents that in the neighborhood, each user point can have at most k unfamiliar member points. For simplicity, we use $NH - RS$ to represent $NH - RS(d, m, k)$.

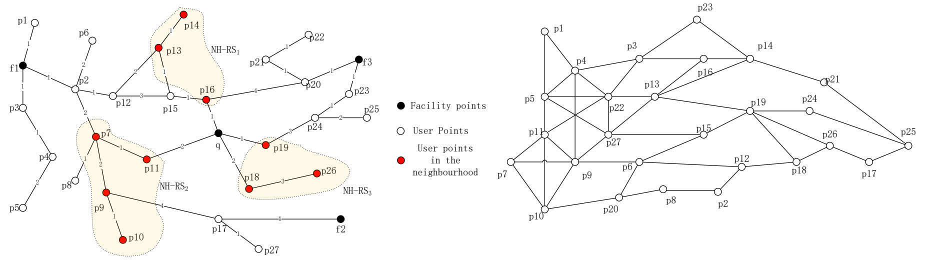
To illustrate the concept of road-social neighbourhoods, Figure 3 gives an example of road-social neighbourhoods queries in spatial databases. Figure a is a road network, and Figure b is a social network diagram. The positions of user points on the social network will be projected onto the road network. In the figure, there are three road-social neighbourhoods $\{NH - RS_1, NH - RS_2, NH - RS_3\}$, with a minimum number of m for each neighbourhood of 3; The distance between each user point and the nearest point in the neighbourhood is less than 3; And each user point knows at least 2 other users in the same neighborhood.

Definition 3. Road-Social Neighbourhood Distance: Given a Road-Social Neighbourhood $NH - RS(d, m, k)$ and a point p_i, where p_i is a point of user or facility, the road-social neighbourhood distance refers to the distance on the road network between the point p_i and the closest point in the road-social neighbourhood $NH - RS(d, m, k)$. It is denoted by $d_{RSH}(p_i, NH - RS)$.

In Figure 3, where $d_{RSH}(p_{12}, NH - RS_1)$ is the distance between p_{12} and the nearest member point p_{13} of $NH - RS_1$, that is, the road network distance between p_{12} and p_{13}.

Definition 4. Reverse Spatial-Range Nearest Neighbourhood Query in Road Social Networks （$\mathbf{RNNH - RS}$）: Given graphs G, a set of facility points F, a set of user points P, a query point $q \in F$, three constraints d, m and k , a reverse nearest neighbourhood query on road-social network (RNNH-RS) return all road-social neighborhoods $\{NH - RS_i\}$ such that: (i) the road network distance of a point $p_j \in NH - RS_i$ to its nearest neighbour point $p_k \in NH - RS_i$ is less than or equal to d,i.e., $\mathrm{dist}(p_j, \ p_k) < d$;(ii) $\forall p_j \in NH_i, \ d_{RSH}(p_j, \ NH - RS_i) \leq \mathrm{dist}(p_j, \ f_{pj})$, where f_{pj} is the nearest facility point to P_j in F, and $\mathrm{dist}(p_j, \ f_{pj})$ is the road network distance; (iii) $|NH - RS_i| \geq$ m;(iv) $NH - RS_i$ identifies the query facility point q as the nearest facility point among all facility points in F, i.e., $d_{RSH}(q, \ NH - RS_i) \leq d_{RSH}(\mathrm{f}, \ NH - RS_i), f \in F \backslash q$; (v) $\forall p_j \in NH - RS_i$, there exist at least k elements of $NH - RS_i$, $p_{j_1}, p_{j_2}, p_{j_2}, \dots, p_{j_k}$ which have social relationships with p_j respectively , i.e., there is edge between p_j and each $p_{j_i} (1 \leq i \leq k)$ in the G_s, it is denoted by $\mathrm{RNNH - RS}(q, d, m, k, P, F)$.

Here, k represents the degree of social relationships among group members. In $\mathrm{RNNH - RS}$, a smaller k will make users in the $NH - RS_i$ less familiar, while a larger k will make users in the $NH - RS_i$ more familiar with each other.



(a) road network (b) social network

Figure. 3. An example of an RNNH-RS query

Put simply, the RNNH-RN query is a snapshot query that retrieves road-social neighbourhoods $\{NH - RS_i\}$ that consider the query point as the nearest of all the other facilities in graphs G_r and G_s. In Figure 3, the RNNH-RS query is initiated by q with a distance constraint d set to 3, the minimum number of user points in the road-social neighbourhood m set to 3, and the social constraint k set to 2. It will return the neighbourhood set $\{NH - RS_1, NH - RS_2, NH - RS_3\}$ composed of red user points above.

4 Algorithm

This section explains our solution and the method used for processing a reverse nearest neighbourhood query based on road social networks (RNNH-RS) query. In response to the problem characteristics of RNNH-RS queries, this section first proposes a baseline method based on Dijkstra algorithm, and then proposes an optimization algorithm based on GT-NVD index.

4.1 Baseline method

This section introduces a baseline algorithm for computing road network distances based on Dijkstra's algorithm.

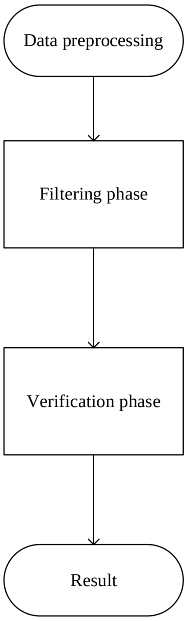


Figure. 4. Baseline algorithm flowchart

From Figure 4, it can be seen that the baseline algorithm is divided into three stages: data preprocessing, filtering, and verification.

Data preprocessing phase.

In the data preprocessing phase, adjacency tables were created for social networks and road networks respectively, in order to more effectively manage data and improve algorithm efficiency. Given the wide distribution of user points in social networks, the algorithm standardizes their longitude and latitude coordinates into a two-dimensional plane space, and maps the position of each user point to the nearest intersection or road segment in the road network, in order to calculate the road distance. When the data processing of the road social network dataset is completed, the baseline algorithm enters the filtering phase.

Filtering phase.

The aim of the filtering process's goal is to prune and optimise the processing of the data point, and then remove a large number of meaningless points. In this phase, we prune the search space instead of accessing all the objects and nodes. In the filtering stage, we use the Voronoi diagram to trim some meaningless points. In the Voronoi diagram, all user points are closer to the query point q than other facility points. Therefore, there is no need for a separate verification procedure because every object in NVC_q (Nearest Voronoi Cell to q) has to be a component of the RNNH-RS response. If NVC_q is empty, then q does not have a reverse nearest neighbor neighborhood.

Verification phase.

For the verification phase, the neighbourhood $NH-RS$ must satisfy the social constraint k. To ensure that each selected user point p_i has good social connectivity with the user points in the neighbourhood $NH-RS_i$, we propose the following conditions based on the paper on social spatial ordering.

Definition 5. Familiarity conditions within the neighbourhood (FNH): When extracting a user point p_i from the search space, we only choose the vertex if it satisfies the FNH. We add the vertex to $NH-RS_i$. FNH is described in formula 2 below.

To effectively consider the social network, this condition assumes that p_i is added to NH_i first, and then it checks whether the social connectivity of the new group $NH-RS_i \cup p_i$ meets the standard k.

$$F(NH-RS_i \cup p_i) = \frac{1}{NH-RS_i \cup p_i} \sum\nolimits_{p_i \in NH-RS_i \cup p_i} |N_{p_i}| \qquad (1)$$

Specifically, let $F(NH-RS_i \cup p_i)$ represent the internal familiarity of the neighborhood, where N_{p_i} is the neighbor set of p_i in $NH-RS_i \cup p_i$. For each user point in $NH-RS_i \cup p_i$, internal familiarity denotes the average number of users known in $NH-RS_i \cup p_i$. If vertex p_i satisfies the following FNH criteria, it is allowed to be selected and added to $NH-RS_i$:

$$F(NH-RS_i \cup p_i) \geq |NH-RS_i \cup p_i| - \frac{t(|NH-RS_i \cup p_i|)}{m-1} - 1 \qquad (2)$$

Here, the hat symbol t denotes a filtering parameter, with an initial value set to $t = m-k-1$.

In layman's terms, when $t = m-1$, the neighbourhood allows all users to be unfamiliar with each other. In this scenario, distance-based search is the optimal strategy. In fact, in this case, the FNH condition becomes $F(NH-RS_i \cup p_i) \geq -1$. On the other extreme, when $t = 0$, the FNH condition becomes $F(NH-RS_i \cup p_i) \geq |NH-RS_i \cup p_i| - 1$, and every user in $NH-RS_i \cup p_i$ needs to be acquainted with all users in $NH-RS_i \cup p_i$.It is worth noting that FNH introduces a filtering parameter t rather than directly incorporating k to correctly handle values of t within the range of $0 < t < m-1$[19].

The baseline algorithmic explanation is presented in Algorithm 1. Here, we demonstrate the procedure with an example.

Algorithm 1: Baseline

```
Input: User points, F Facility points, query point q ∈ F, Three constraint conditions ($d$ is distance constraint, $m$ is cardinality constraint, and $k$ is social constraint)
Output: Neighbourhood $NH - RS_i$ set
Calculate $NVC_q$ (user point closest to $q$)// Filtering phase
Put it into pile h
while h is not empty do
    de-heap an entry $p_i$;
    if isNotVisited($p_i$) then
        mark $p_i$ as visited;
        if Checkfamiliarity($p_i$) then// Verification phase
            initialize $NH - RS_i \leftarrow \{p_i\}$;
            $d_H(q, NH - RS_i) \leftarrow$ dist(q, $p_i$) ;
            vList ← Range($p_i, d$);
            while vList is not empty do
                de-heap an entry $p_j$;
                if $p_j \in NVC_q \wedge$ Checkfamiliarity($p_j$) then
                    mark $p_j$ as visited ;
                    $NH - RS_i \leftarrow$ append($p_j$);
                    vList ← Range($p_j, d$);
                else if dist($p_j, NH - RS_i$) ≤ dist($p_j, f_{p_j}$) then
                    if $d_H(q, NH - RS_i) \leq$ dist($p_j, f_{p_j}$) $\wedge$ Checkfamiliarity($p_i$) then
                        $NH - RS_i \leftarrow$ append($p_j$);
                        vList ← Range($p_j, d$);
                        mark $p_j$ as visited ;
            if $|NH - RS_i| \geq$ m then
                RNNH-RS← append($NH - RS_i$);
return Neighbourhood $NH - RS_i$ set ;
```

For example, in Figure 3, an $RS - RNNH$ query is initiated by f_2, where $k = 2$，$m = 3$，$d = 3$. Initially, $\hat{t} = 0$. Since p_{16} is the vertex with the minimum spatial distance to f_2, and $F(NH - RS_i \cup p_{16}) = \frac{0}{1} \geq 1 - \frac{0\times1}{2} - 1$ satisfies FNH, $NH - RS_1$ is initialized, and p_{16} is added to $NH - RS_1$. p_{16} is marked as visited. Then, a range query is performed on p_{16} with a range of d , resulting in the query results $\{p_{15}, p_{13}, p_{14}, p_{12}, p_2\}$, which are placed in the $vList$ queue. Next, the algorithm selects the user point p_{15} from the list and calculates $F(NH - RS_1 \cup p_{15}) = \frac{0}{2} < 2 - 1$, which does not satisfy FNH. Therefore, p_{15} is marked and discarded.

The algorithm proceeds to check the next point, p_{13}, and calculates $F(NH_1 \cup p_{13}) = \frac{1}{2} \times 2 = 1 \geq 2 - \frac{0\times2}{2} - 1$. Since p_{13} is in NVC_q, it is added to $NH - RS_1$, marked as visited, and then a range query is performed on p_{13} with a range of d. The query results $\{p_{16}, p_{15}, p_{14}, p_{12}, p_2\}$ are added to the $vList$ queue, and the list $vList$ is updated. The algorithm then selects p_{14} from the $vList$, calculates $F(NH - RS_1 \cup p_{14}) = \frac{1}{3} \times 6 = 2 \geq 3 - \frac{0\times3}{2} - 1$. Since p_{14} is in NVC_q, it is added to $NH - RS_1$, marked as visited. Next, retrieve p_{12} and p_2 from the list $vList$ in sequence, and find that neither satisfies FNH, so discard them. At this point, $NH - RS_1$ contains 3 user points, meeting the condition, so the first neighbourhood $NH - RS_1$ for query point q is generated.

4.2 Optimization method

To solve the RNNH-RS query problem, we first proposes a query method based on the Dijkstra algorithm. However, when querying two points that are far apart, its query efficiency is quite low. In this section, we propose a new index called GT-NVD, which is composed of the G-tree index and Network Voronoi Diagrams (NVD), aiming to further enhance query speed. Checking each possible user point based on social constraints is expensive. Therefore, in order to effectively prune unnecessary search space, the upper bound of social constraints for each possible user point growing from each road-social neighbourhood is obtained in this section, and an early termination strategy is proposed. And proposed an optimization algorithm called query optimization methods(QOM).

Index scheme.

As depicted in Figure 6, the backbone of GT-NVD is a G-tree, which recursively divides the road network into subnetworks. We employ state-of-the-art hierarchical tree structures to construct GT-NVD. Initially, the road network G_r is set as the root and divided into f equally sized subgraphs, each serving as a child node of the root.Each subgraph of the road network, denoted as $G_r^i = (V_r^i, E_r^i, B_r^i)$, is represented by a tree node T_i,where V_r^i represents a set of vertices in graph G_r^i, E_r^i represents a set of edges in graph G_r^i,and B_r^i is a set of boundary vertices.Each tree node is associated with a Network Voronoi Cell (NVC), which is constructed to partition the road network based on the locations of road network nodes within the road network G_r^i.The NVC is a data structure that partitions G_r^i into disjoint regions, such that each region corresponds to a specific facility point $f \in F$ and the set of user points $p \in P$ with f as their nearest neighbor(NN).

Figure 5 illustrates an example of the NVD constructed based on $F = \{f_1, f_2, f_3, f_4, f_5\}$.The road network G_r^i is partitioned into 5 Voronoi network cells, with each NVC containing a facility point and the nearest user point to the facility point. By utilizing the NVC, the nearest facility point $f \in F$ for any user point $p \in P$ can be obtained. The nearest facility point for user point p_1 is f_2,as p_1 belongs to the NVC of f_2.

Therefore, the non-leaf node T_i of GT-NVD is represented as (G_r^i, Pt, B_i, NN_i),where G_r^i is the subgraph indexed by T_i, Pt is a set of pointers pointing to the child nodes of T_i, B_i is a set of boundary points, and NN_i associates each $b \in B_i$ with the corresponding nearest neighbors (NN) of the NVC. For instance, in Figure 6, for the tree node T_1,$B_1 = \{p_{11}\}$.According to NVC$_{f_3}$,it is known that the nearest neighbor (NN) of p_{11} is f_3,so $NN_{p_{11}} = f_3$ is stored in NN_1.The leaf node T_j is represented as $\left(G_r^j, B_j, V_r^j, NN_j\right)$,where V_r^j is the set of vertices in G_r^j, NN_j associates each vertex $v \in V_r^i$ with the corresponding nearest neighbors (NN), and the remaining elements are the same as those in non-leaf nodes.

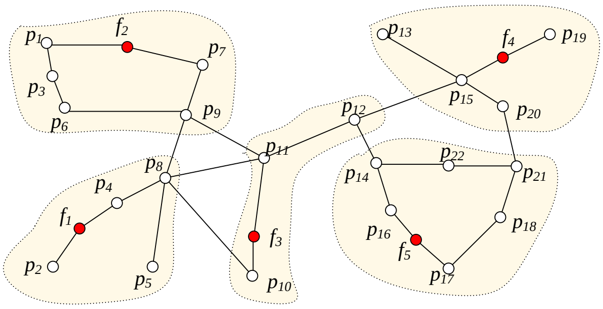
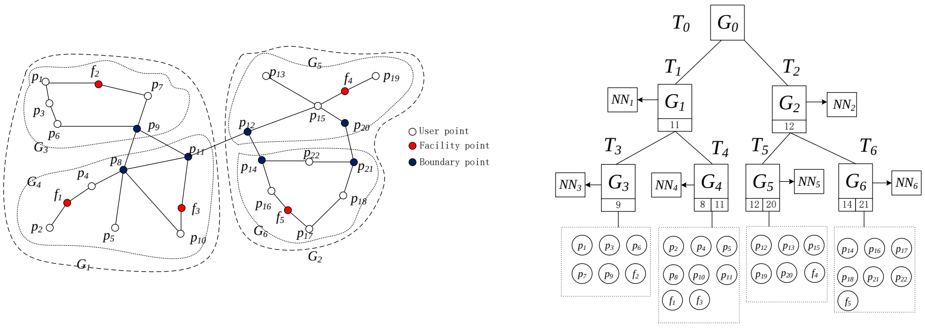


Figure. 5. Illustration of NVD structures.



(a) Recursive road network partition by G-tree

(b) GT-NVD

Figure. 6. GT-NVD.

In summary, given two vertices (u, v), to calculate $dist(u, v)$. If (u, v) is in the same leaf node, we first calculate their minimum common ancestor and all vertices on the path from the leaf node containing u to the leaf node containing v, then use dynamic programming to compute $dist(u, v)$. If (u, v) are not in the same leaf node, there are two cases: if the path from u to v does not include vertices outside the leaf nodes, we directly calculate the shortest network distance using the Dijkstra algorithm, which is efficient enough due to the small size of leaf node subgraphs. If the shortest path from u to v includes vertices not in the same leaf node as (u, v), then the shortest network distance between u and v must involve two boundaries. We calculate it by adding the distance from u to the boundary and the distance from v to the boundary.

So the GT-NVD index can be used to calculate the shortest path on the road network, reducing the time required to calculate the distance between two points; When calculating the distance between user point p_i and his nearest neighbor, the NN in the index can also be directly referenced for calculation, which improves the calculation speed.

Early termination strategy.

The optimization method also proposes an upper bound on the social intimacy of each possible neighbourhood growing from $vlist$, in order to effectively prune unnecessary search space. Every feasible neighbourhood cannot satisfy social restrictions if the upper bound shows that on average, each user knows less than k other user points. Therefore, when the following conditions are met, the algorithm stops processing $NH - RS_i$:

Theorem 1: Given a road-social neighbourhood $NH - RS_i$ to be processed and a set $vlist$ of candidate user points, if the upper bound Ub of the social intimacy of neighbourhood $NH - RS_i$ is less than k, the processing is stopped and the road-social neighbourhood $NH - RS_i$ is discarded.

Proof. Specifically, the edges in any solution growing from $NH - RS_i$ can be divided into three categories: 1) the set of edges E_n connecting any two vertices in $NH - RS_i$, 2) the edges of edges E_v connecting any two vertices selected from $vlist$, and 3) the set of edges E_{nv} connecting any two vertices in $NH - RS_i$ and the vertices selected from $vlist$.

Formula 3 can be derived, where $N_{p_i}^{I}$is the set of acquainted neighbors of p_i in $NH - RS_i$.

$$|E_n| = \frac{1}{2}\sum\nolimits_{p_i \in NH - RS_i} |N_{p_i}^{I}| \tag{3}$$

Since the selected vertices in $vlist$ is not clear now, a good way is to find an upper bound on $|E_v|$, the following formula 4.

$$E_v Ub = \frac{1}{2}(m - |NH - RS_i|) max_{p_i \in vlist} |N_{p_i}^{vlist}| \tag{4}$$

In formula 4 $N_{p_i}^{vlist}$ is the set of acquainted neighbors of p_i in $vlist$. $E_v Ub$ is an upper bound because the vertex with the maximum degree in $vlist$ is identified, and $(m - |NH - RS_i|)$ vertices are selected from $vlist$.

Similarly, the upper bound of $|E_{nv}|$, $E_{nv}Ub$, is described in formula 5 as follows. $InterEdge(p_i)$ is the set of edges connecting p_i in $NH - RS_i$ to any vertices in $vlist$.

$$E_{nv}Ub = \sum\nolimits_{p_i \in NH - RS_i} |InterEdge(p_i)| \tag{5}$$

Notice that the number of edges in a feasible solution is half of the degree sum of all the vertices in the solution, where the degree of a vertex represents the number of acquainted neighbors in the solution. Therefore, with the above three categories of edges, algorithm stops processing $NH - RS_i$ when the following condition holds,

$$\frac{2}{p}(|E_n| + E_v Ub + E_{nv} Ub) < k \tag{6}$$

In the above condition, the average number of other users known by users in each road-social neighbourhood is less than k. Algorithm stops processing $NH - RS_i$ if every possible solution growing from $NH - RS_i$ via the exploration of $vlist$ is not able to satisfy the Social constraint k.

For the social graph with $m = 3$ and $k = 2$ in Figure 7, if $NH - RS_1 = \{e, f\}, vlist = \{c, d\}$, the pruning of social constraint degree stops processing $NH - RS_1$ because $\frac{2}{3}(0 + \frac{1}{2} \times 1 + 1) = 1 < 2$. In other words, under social constraints, moving any vertex from $vlist$ to $NH - RS_1$ will not produce a feasible solution.

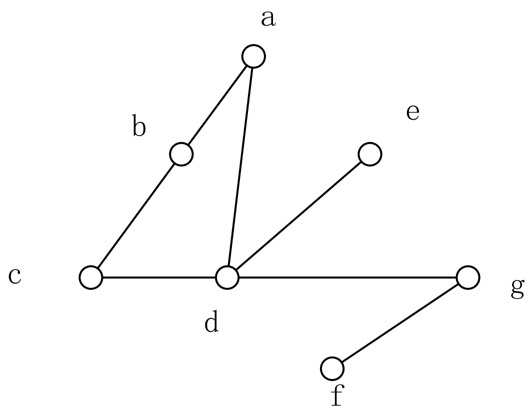


Figure. 7. An example of early termination.

The following is the pseudocode for the QOM algorithm based on GT-NVD index and early termination strateg.

```
Algorithm 2: QOM
Input: User points, F Facility points, query point q ∈ F, Three constraint conditions ($d$ is distance constraint, $m$ is cardinality constraint, and $k$ is social constraint)
Output: Neighbourhood $NH - RS_i$ set
Initialize GT-NVD index
Calculate $NVC_q$ (user point closest to $q$)// Filtering phase
Put it into pile h
while h is not empty do
    de-heap an entry $p_i$;
    if isNotVisited($p_i$) then
        mark $p_i$ as visited;
        if Checkfamiliarity($p_i$) then// Verification phase
            initialize $NH - RS_i \leftarrow \{p_i\}$;
            $d_H(q, NH - RS_i) \leftarrow \mathrm{dist}(\mathrm{q}, p_i)$ ;
            vList ← Range($p_i, d$);
            while vList is not empty do
                de-heap an entry $p_j$;
                if $p_j \in NVC_q \wedge$ Checkfamiliarity$(p_j)$ then
                    mark $p_j$ as visited ;
                    $NH - RS_i \leftarrow$ append($p_j$);
                    vList ← Range($p_j, d$);
                else if dist($p_j, NH - RS_i$) $\le$ dist($p_j, f_{p_j}$) then
                    if $d_H(q, NH - RS_i) \le \mathrm{dist}(p_j, f_{p_j}) \wedge$ Checkfamiliarity($p_i$) then
                        $NH - RS_i \leftarrow$ append($p_j$);
                        vList ← Range$(p_j, d)$;
                        mark $p_j$ as visited ;
                    if $Upper\ Bound\ Upper\ Bound < m - k - 1$ then
                        Break;// Early termination strategy
            if $|NH - RS_i| \ge$ m then
                RNNH-RS← append($NH - RS_i$);
return Neighbourhood $NH - RS_i$ set ;
```

5 Experiments

5.1 Experimental setup

Dataset. Our experiment studied the real-life social network Brightkite and the real road networks San Francisco Road Network(referred to as SF) , as well as the City of

San Joaquin Country Road Network(referred to as TG) and the City of Oldenburg(referred to as OL).Due to the global distribution of locations within social networks, we standardize the latitude and longitude coordinates of user locations in the social network into a two-dimensional plane. Subsequently, based on these coordinates, each user are mapped to the nearest intersection or road segment in the road network.We denote the road social network dataset composed of the social network Brightkite and road network SF as BR+SF. Similarly, the road social network dataset composed of Brightkite and TG is represented as BR+TG. BR+OL denotes the road social network dataset composed of Brightkite and OL. The specific information of the dataset for road social networks is shown in Table 2 and Table 3.

Table 2. Dataset details

Name	Vertices	Edges
BR	58228	428156
SF	174956	397415
TG	18263	23797
OL	6104	7034

Table 3. Detailed information table for road social networks

Name	User points	Facility points
BR+SF	58228	174956
BR+TG	58228	18263
BR+OL	58228	6104

Algorithm. As far as we know, the RNNH-RS problem on general road social networks has not been studied before. In this paper, we implement and evaluate a baseline algorithm 1 based on Dijkstra and a QOM algorithm 2 based on GT-NVD index and early termination strategy. We conducted experiments under different settings by varying three parameters: the number of users in the neighbourhood (m), distance constraint (d), and social constraint (k).

Table 4. RNNH-RS experimental parameter settings

Parameter	Range	Default value
d	5-30	5
m	3-30	3
k	0-4	0

All programs are implemented in standard C++ and compiled using g++ on Linux.

5.2 Experimental results and analysis

Effectiveness comparison.

The experimental comparison in this section verified the effectiveness of reverse nearest neighbourhood query based on road social networks(RNNH-RS) based on road social networks compared to reverse nearest neighbor queries (RNNH-RN) under road networks in road social networks.

Figures 8 and 9 illustrate the comparison of neighborhoods satisfying social relationship constraints in the BR + TG dataset when the number of user points within the neighborhood, m, is set to 5, the neighbourhood distance, d, is set to 30, and different values of social constraint k are applied. The comparison is conducted between the RNNH-RS and the RNNH-RN.

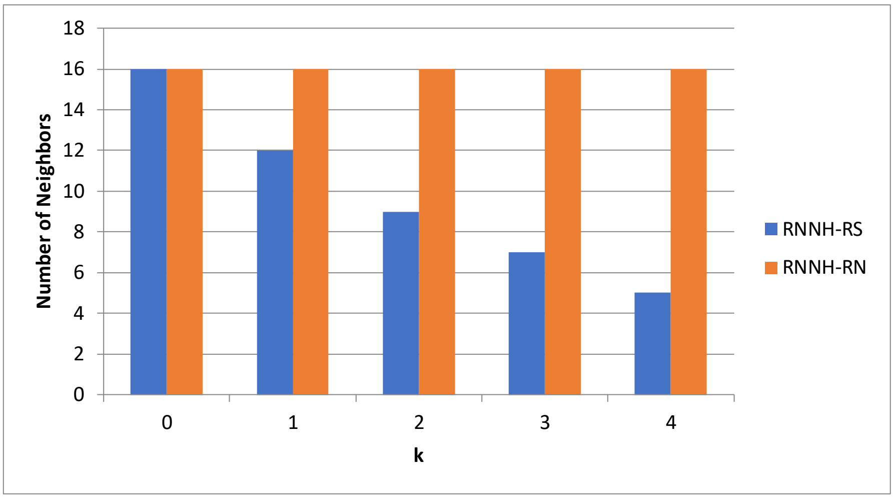


Figure. 8. The number of neighbourhoods of algorithm when varying k on the BR+TG dataset

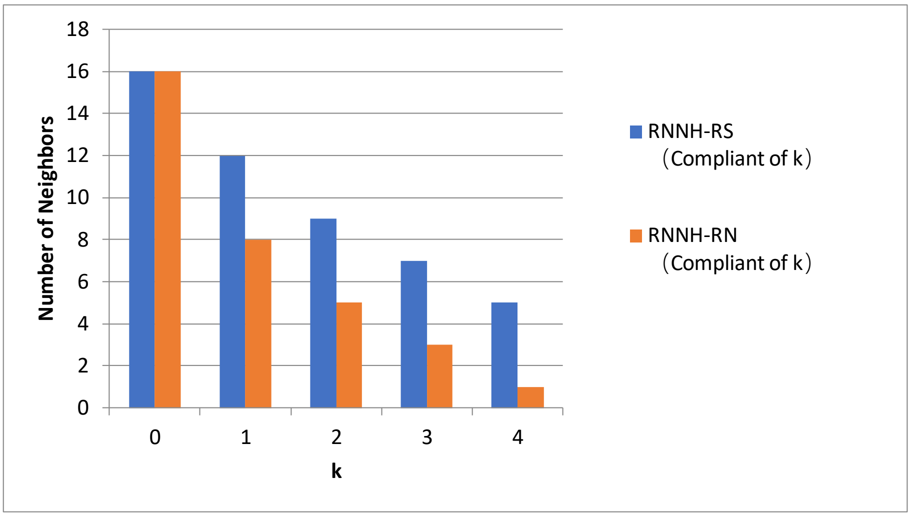


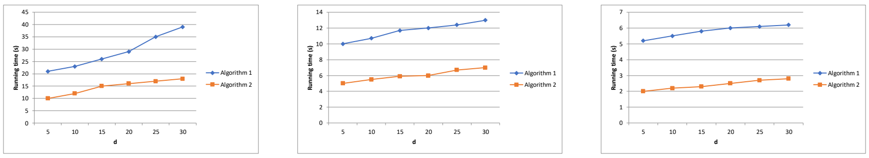
Figure. 9. The number of neighborhoods that algorithm conforms to social relationships when varying k on the BR+TG dataset

From Figure 8, it can be seen that as the social constraint k increases, that is, the number of people familiar to users in the neighbourhood increases, the number of neighborhoods returned by RNNH-RS gradually decreases because the social constraint increases, and the number of user points in the search space that meet the social constraint decreases, resulting in fewer neighborhoods; However, RNNH-RN queries do not consider the influence of social constraint k, resulting in an unchanged number of returned neighborhoods.

In Figure 9, it can be seen that the number of neighborhoods that comply with the social relationship k is calculated in the neighborhoods returned by RNNH-RN and RNNH-RS. When considering the influence of social constraints, as k continues to increase, the number of neighborhoods that comply with the social constraint k in the query returned by RNNH-RN decreases compared to the initial number returned in Figure 8; The number of neighborhoods returned by RNNH-RS remains unchanged compared to the initial number returned in Figure 8, as RNNH-RS queries consider the influence of social relationships during computation. However, in the neighborhoods returned by RNNH-RN queries, very few neighborhoods that meet the social constraint k were found. This indicates that in the results of RNNH-RN queries, if users want to consider the impact of social interactions, and RNNH-RN cannot effectively solve the problem of solving user social relationships, the RNNH-RS algorithm provides a more effective way to solve this problem. So it has been proven that RNNH-RS is more effective than RNNH-RN.

Efficiency comparison.

Figure 10 compares the execution efficiency between Baseline Algorithm 1 and Optimized Algorithm 2 by varying the distance parameter d among member points within the neighborhood. The comparison is performed across three different datasets, with the default values of parameter m set to 3 and parameter k set to 2.

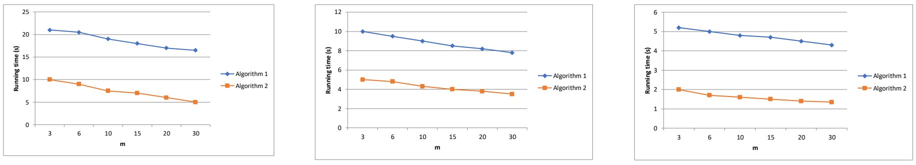


(a)BR+SF (b)BR+TG (c)BR+OL

Figure. 10. Performance comparison of changing parameter d on different datasets

We evaluated the performance of RNNH-RS queries on real and synthetic datasets, setting the value of d to 5, 10, 15, 20, 25, and 30. As shown in the figure 10, we draw the following conclusions: The optimized RNNH-RS algorithm has a much faster average processing time compared to the basic RNNH-RS algorithm. As d increases, the query time for both algorithms gradually increases. However, the optimized algorithm always consumes less time than the baseline algorithm, indicating that the larger the distance constraint d, the more objects can be queried, and filtering is also more time-consuming. However, Algorithm 2 based on GT-NVD index has a shorter running time, so it is more efficient than the baseline based on Dijkstra.

Figure 11 compares the execution efficiency between Baseline Algorithm 1 and Optimized Algorithm 2 by varying the number of members within the neighborhood, denoted as m, across three different datasets. The default values of parameter d are set to 5, and parameter k is set to 2.

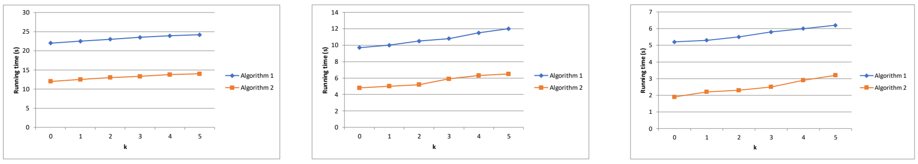


(a)BR+SF (b)BR+TG (c)BR+OL

Figure. 11. Performance comparison of changing parameter m on different datasets

We also investigated the impact of the minimum neighbourhood size (m). We found that in general, as the value of m increases, the number of neighbourhood queries decreases significantly, and the average neighbourhood size also decreases. This is because when the minimum neighbor count increases, neighborhoods with fewer neighbors are excluded. As shown in the figure 11, with the increasing m , the query time for both the baseline algorithm and the optimized algorithm decreases gradually. However, Algorithm 2 consistently consumes less time than Algorithm 1. The execution time of Algorithm 2, based on the GT-NVD index, is shorter, making it more efficient than the baseline based on Dijkstra.

Figure 12 compares the execution efficiency between Baseline Algorithm 1 and Optimized Algorithm 2 by varying the social constraint parameter, denoted as k, across three different datasets. The default values of parameter d are set to 5, and parameter m is set to 6.



(a)BR+SF (b)BR+TG (c)BR+OL

Figure. 12. Performance comparison of changing parameter k on different datasets

As shown in Figure 12, it can be seen that optimization algorithm 2 based on GT-NVD index has much shorter execution time than baseline algorithm 1 based on Dijkstra under different datasets and social constraint parameters k. As the value of k gradually increases, the running time of the algorithm also gradually decreases. This is because when k is 0, all users in the neighbourhood do not need to know each other. User points can be placed in the neighbourhood if they satisfy distance constraints, making it easier to find user points that meet the conditions. Each user point is checked less frequently by FNH. The larger the value of k, the more strict the familiarity check is performed, and there are fewer user points that meet social constraints. Moreover, each query involves familiarity checks between users, and only those who meet the standards are included in the neighborhood. This leads to the fact that, after the query point has been filtered through the filtering phase to produce a candidate user point, the number of candidate points satisfying the conditions is small, and therefore fewer user points need to be detected afterwards, and the algorithm needs to be executed in less time. However, the optimized algorithm always has a shorter runtime than the baseline algorithm. Algorithm 2 based on GT-NVD index has a shorter execution time and is more efficient than baseline algorithm 1 based on Dijkstra.

6 Conclusion

In this paper, we define a practical query type, RNNH-RS query, to identify suitable neighboring objects for query points on road social networks. To the best of our knowledge, there have been no previous solutions for such scenarios in the context of reverse nearest neighbor queries on road social networks. We first introduce this problem and designed an algorithm, including efficient filtering and verification techniques, to reduce processing time. Furthermore, the optimized query performance has been improved, requiring less time to find the optimal solution on road social networks.

Experimental results in real-world road social networks significantly demonstrate that our approach is highly scalable and robust in terms of efficiency and effectiveness. Possible directions for future work include integrating other user attributes, such as user preferences, to filter active participants or to identify the neighbourhood communities that have the most significant impact on query points.

7 Acknowledgments

This research is supported by: the National Natural Science Foundation of China(No.62066034 No.62262047). We thank Nasser[1] for sharing the RNNH source code.

References

1. Allheeib N, Islam M S, Taniar D, et al.: Density-based reverse nearest neighbourhood search in spatial databases. Journal of Ambient Intelligence and Humanized Computing, 12: 4335-4346(2021)
2. Allheeib N, Adhinugraha K, Taniar D, et al.: Computing reverse nearest neighbourhood on road maps. World Wide Web, 1-32(2022)
3. Jin P, Chen L, Gao Y, et al.: Maximizing the influence of bichromatic reverse k nearest neighbors in geo-social networks. World Wide Web,1-32(2022)
4. Korn F, Muthukrishnan S.: Influence sets based on reverse nearest neighbor queries. ACM Sigmod Record, 29(2): 201-212(2000)
5. Singh A.: Efficient Processing of Reverse Nearest Neighbor Queries on High Dimensional Data. Ohio State University, (2004)
6. Ying Lu, Gao Cong, Jiaheng Lu, and Cyrus Shahabi.:Efficient algorithms for answering reverse spatial-keyword nearest neighbor queries. In Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL '15). Article 82, 1–4. Association for Computing Machinery, New York, NY, USA, (2015)
7. Safar, M., Ibrahimi, D. & Taniar, D.: Voronoi-based reverse nearest neighbor query processing on spatial networks. Multimedia Systems 15, 295–308 (2009)
8. A. Vlachou, C. Doulkeridis, Y. Kotidis and K. Nørvåg.: Reverse top-k queries. 2010 IEEE 26th International Conference on Data Engineering (ICDE 2010), pp. 365-376. Long Beach, CA, USA, (2010)
9. Yang S, Cheema M A, Lin X, et al.: Reverse k nearest neighbors queries and spatial reverse top-k queries. The VLDB Journal, 26: 151-176(2017)

10. Nikitopoulos P, Sfyris G A, Vlachou A, et al.: Pruning techniques for parallel processing of reverse top-k queries. Distributed and Parallel Databases, 39: 169-199(2021)
11. Xiao G, Li K, Li K.: Reporting l most influential objects in uncertain databases based on probabilistic reverse top-k queries. Information Sciences, 405: 207-226(2017)
12. Chen Z, Wang X, Liu W.: Reverse keyword-based location search on road networks. GeoInformatica, 1-31(2022)
13. Ahuja R, Armenatzoglou N, Papadias D, et al.: Geo-social keyword search. Advances in Spatial and Temporal Databases: 14th International Symposium, 2015: 431-450. Springer International Publishing, Hong Kong, China. (2015)
14. Shim C, Kim W, Heo W, et al.: Nearest close friend search in geo-social networks. Information Sciences, 423: 235-256(2018)
15. Gao R, Li J, Li X, et al.: A personalized point-of-interest recommendation model via fusion of geo-social information. Neurocomputing, 273: 159-170(2018)
16. Shim C, Sim G, Chung Y D.: Cohesive ridesharing group queries in geo-social networks. IEEE Access, 8: 97418-97436(2020)
17. Wang K, Wang S, Cao X, et al.: Efficient radius-bounded community search in geo-social networks. IEEE Transactions on Knowledge and Data Engineering, 34(9): 4186-4200(2020)
18. Attique M, Afzal M, Ali F, et al.: Geo-social top-k and skyline keyword queries on road networks. Sensors, 20(3): 798(2020)
19. Yang D N, Shen C Y, Lee W C, et al.: On socio-spatial group query for location-based social networks. Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, 949-957(2012)
20. Guo F, Yuan Y, Wang G, et al.: Cohesive group nearest neighbor queries over road-social networks. 2019 IEEE 35th International Conference on Data Engineering (ICDE). IEEE, 434-445(2019)
21. Jin P, Chen L, Gao Y, et al.: Maximizing the influence of bichromatic reverse k nearest neighbors in geo-social networks. World Wide Web, 1-32(2022)
22. Allheeib N, Taniar D, Al-Khalidi H, et al.: Safe regions for moving reverse neighbourhood queries in a peer-to-peer environment. IEEE Access, 8: 50285-50298(2020)