Information Systems 61 (2016) 24–39

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

Group meetup in the presence of obstacles

Nusrat Sultana [a], Tanzima Hashem [a,*], Lars Kulik [b]

[a] Department of CSE, Bangladesh University of Engineering and Technology, Bangladesh
[b] Department of CIS, University of Melbourne, Australia

ARTICLE INFO

Article history:
Received 27 October 2015
Accepted 8 April 2016
Recommended by: F. Carino Jr.
Available online 4 May 2016

Keywords:
Obstacles
Obstructed distance
Group nearest neighbor queries

ABSTRACT

In this paper, we introduce an obstructed group nearest neighbor (OGNN) query that enables a group of pedestrians to meet at a common point of interest (e.g., a restaurant) with the minimum aggregate travel distance in the presence of obstacles such as buildings and lakes. The aggregate travel distance can be measured in terms of the total, the maximum or the minimum travel distance of all group members. In recent years, researchers have focused on developing efficient algorithms for processing group nearest neighbor (GNN) queries in the Euclidean space and road networks, which ignores the impact of obstacles in computing travel distances. We propose the first comprehensive approach to process an OGNN query. We present efficient algorithms to compute aggregate obstructed distances, which is an essential component of processing OGNN queries. We exploit geometric properties to develop pruning techniques that reduce the search space and incur less processing overhead. Based on various search space refinement techniques, we propose two algorithms: a Group Based Query Method (GBQM) and a Centroid Based Query Method (CBQM) to evaluate OGNN queries. We validate the efficacy and efficiency of our solution through extensive experiments using both real and synthetic datasets and present a comparative analysis among our proposed algorithms in terms of query processing overhead.

1. Introduction

An important class of information and enquiry service is the group nearest neighbor (GNN) query [1,2] that enables a group to meet with the minimum aggregate travel distance. For example, through a GNN query, a group of friends, located at different places in a city center, can determine the location of a common point of interest (POI) such as a restaurant, a shopping mall or a movie theater that minimizes their *total travel distance* for all group members. The aggregate distance can be also measured in terms of minimizing the *maximum or the minimum travel distance* of all group members. An OGNN query that minimizes the maximum distance of the group members enables the group to meet at a POI within the *shortest possible time*. On the other hand, sometimes a group member may need to reach a POI at the *earliest possible time*, for example, to secure a table in a restaurant, and in such a scenario, minimizing the minimum distance of the group members for a GNN query is important. GNN queries have applications in different domains including geographic information system (GIS) and mobile applications [2], clustering [3] and outlier detection [4].

GNN queries have been investigated for the Euclidean space [1,5–10] and road networks [2,11], but these GNN algorithms did not consider the impact of obstacles on the travel distance. For example, a park has a number of obstacles such as trees, lakes or fences that obstruct a pedestrian's direct path and require a detour. Similarly, in a city center some buildings might restrict a pedestrian's direct path or some roads might only permit vehicles.

* Corresponding author.
E-mail addresses: nusrat.cse@gmail.com (N. Sultana), tanzimahashem@cse.buet.ac.bd (T. Hashem), lkulik@unimelb.edu.au (L. Kulik).

http://dx.doi.org/10.1016/j.is.2016.04.003
0306-4379/

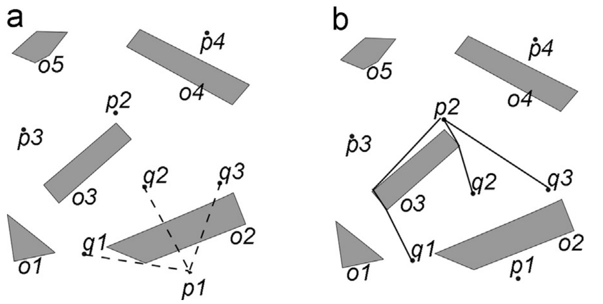

Fig. 1. An example scenario, where $o_1, o_2, \ldots, o_5$ represent obstacles, q_1, q_2 and q_3 represent group members, and p_1, p_2 and p_3 represent POIs.

Current GNN queries are tailored to Euclidean space or road networks but ignore the impact of obstacles. In this paper, we introduce a novel form of GNN queries in the presence of obstacles, which we call *Obstructed Group Nearest Neighbor* (*OGNN*) queries. To the best of our knowledge, we propose the first comprehensive approach to find group nearest neighbors (GNNs) in the presence of obstacles.

In Euclidean space, the trip distance is measured ignoring the presence of obstacles. The nearest restaurant for a group of friends located at different location of a city center using the Euclidean distance might not be the nearest restaurant for an OGNN query if buildings obstruct the direct path. For example, in Fig. 1(a), a GNN query in the Euclidean space returns p_1 as answer, while an OGNN query returns p_2 as the answer as p_2 provides the minimum aggregate obstructed distance due to the presence of obstacles (Fig. 1(b)). In road networks, the movement is permitted in a predefined structure and the distance is measured as the length of the shortest path between two points. On the other hand, in an obstructed space obstacles determine areas that cannot be crossed and the distance is measured as the length of the shortest path between two points in the presence of obstacles. For example, in Fig. 1 (b), the obstructed distance between q_1 and p_2 is measured as the sum of the length of 3 segments.

Although there are algorithms [12–16] to evaluate nearest neighbor queries, moving nearest neighbor queries and reverse nearest neighbor queries in the obstructed space, a straightforward application of an obstructed nearest neighbor (ONN) algorithm to evaluate OGNN queries equals to an exhaustive search and incurs prohibitively expensive computations. Such a straightforward technique requires to incrementally evaluate obstructed nearest neighbors (i.e., POIs) for every query point (i.e., every group member's location) independently and compute the aggregate obstructed distance for each retrieved POI until the obstructed GNN has been identified.

The smaller the number of POIs that an OGNN algorithm has to consider for computing the actual obstructed GNN, the more efficient the algorithm is because for every retrieved POI, an OGNN algorithm has to compute the aggregate obstructed distance with respect to the query points. To develop an efficient approach for OGNN queries, we focus on developing (i) pruning techniques that refine the search space and (ii) efficient algorithms that compute aggregate distances under the presence of obstacles.

We propose two algorithms, a *Group Based Query Method* (GBQM) and a *Centroid Based Query Method* (CBQM), for processing OGNN queries. GBQM incrementally retrieves Euclidean GNNs and refines the search space by exploiting the fact that the obstructed distance is always greater or equal to the Euclidean distance between two points. The search for obstructed GNNs stops when the aggregate obstructed distance for a retrieved GNN becomes less or equal to the aggregate Euclidean distance of the last retrieved GNN. On the other hand, CBQM incrementally retrieves Euclidean NNs for the centroid of the query points and refines the search space based on geometric properties. The search space becomes smaller with the retrieval of POIs and the search terminates when the POIs inside the refined region have been retrieved.

We present two aggregate obstructed distance computation techniques, *Single Point Aggregate Obstructed Distance* (SPAOD) and *Multi-Point Aggregate Obstructed Distance* (MPAOD) computation. In computational geometry [17], researchers assume a small set of obstacles and preprocess them to compute obstructed distances, whereas spatial applications require to handle a large set of obstacles that are indexed using an *R*-Tree on the database. Considering all obstacles every time for distance computations is not a feasible solution. Though there are algorithms [13,16] to compute the obstructed distance between two points for a large set of obstacles, applying those algorithms independently for every obstructed distance computation for an aggregate distance would incur high query processing overheads.

The underlying idea of SPAOD and MPAOD is to reduce the number of obstacle retrievals from the database. SPAOD does not retrieve an obstacle multiple times but may retrieve few additional obstacles, which are not required for computing the aggregate obstructed distances. On the other hand, MPAOD may retrieve few obstacles multiple times but does not retrieve any obstacle, which is not required for computing aggregate obstructed distances.

We summarize our key contributions as follows:

- We introduce GNN queries in the obstructed space.
- We develop pruning techniques to efficiently evaluate GNN queries in an obstructed space.
- We propose two efficient techniques to compute aggregate distances under obstruction for POIs with respect to the group member locations.
- We conduct an extensive experimental analysis using both real and synthetic datasets and show a comparative analysis between our algorithms based on different parameters. Our proposed algorithms are scalable and can determine obstructed GNNs with reduced processing overhead.

This paper extends the work in [18], where we proposed the first approach for processing GNN queries in the obstructed space using the aggregate functions SUM and MAX. In this paper, we enhance our work in the following ways: (i) we introduce a new algorithm, SPAOD, to