Computational Geometry 47 (2014) 695–709

Contents lists available at ScienceDirect

Computational Geometry: Theory and Applications

www.elsevier.com/locate/comgeo

Quickest path queries on transportation network

Radwa El Shawi [a,1], Joachim Gudmundsson [b,c,*,2], Christos Levcopoulos [d,3]

[a] College of Computer Science and Information Technology, University of Dammam, Saudi Arabia
[b] School of IT, University of Sydney, Australia
[c] NICTA, Sydney, Australia [4]
[d] Department of Computer Science, Lund University, Sweden

ARTICLE INFO

Article history:
Received 26 September 2011
Received in revised form 26 July 2012
Accepted 28 January 2014
Available online 31 January 2014
Communicated by J. Mitchell

Keywords:
Computational geometry
Approximation algorithms
Shortest path
Transport networks

ABSTRACT

This paper considers the problem of finding the cost of a quickest path between two points in the Euclidean plane in the presence of a transportation network. A transportation network consists of a planar network where each road (edge) has an individual speed. A traveler may enter and exit the network at any point on the roads. Along any road the traveler moves with a fixed speed depending on the road, and outside the network the traveler moves at unit speed in any direction.

We show how the transportation network with n edges in the Euclidean plane can be preprocessed in time $O((\frac{n}{\varepsilon})^2 \log n)$ into a data structure of size $O((\frac{n}{\varepsilon})^2)$ such that $(1+\varepsilon)$-approximate quickest path cost queries between any two points in the plane can be answered in time $O(\frac{1}{\varepsilon^4}\log n)$.

In addition we consider the nearest neighbor problem in a transportation network: given a transportation network with n edges in the Euclidean plane together with a set $\mathcal{Z}$ of m sites, a query point $q \in \mathcal{R}^2$, find the nearest site in $\mathcal{Z}$ from q. We show how the transportation network can be preprocessed in time $O((n^2+nm)\log(n+m))$ such that $(1+\varepsilon)$-nearest neighbor query can be answered in time $O(\frac{1}{\varepsilon^2}\log(n+m))$.

1. Introduction

Transportation networks are a natural part of our infrastructure. We use bus or trains in our daily commute, and often walk to connect between networks or to our final destination.

In this paper we focus on the quickest path problem in a transportation network, but we will also consider nearest neighbor queries and their variants. A transportation network consists of roads and nodes; roads are assumed to be directed straight-line segments along which one can move at a certain fixed speed and nodes are endpoints of roads [3,5–7,13,14]. Thus a transportation network is usually modelled as a plane graph $\mathcal{T}(S,\mathcal{C})$ in the Euclidean plane (or some other metric) whose vertices S are endpoints of roads or intersections and whose directed edges $\mathcal{C}$ are roads. Furthermore, each edge has a weight $\alpha \in (0,1]$ assigned to it. The cost of moving along a directed edge is the distance travelled along the edge times the weight of the edge. Movement outside the network has unit cost. One can access or leave a road through any point on

* Corresponding author at: School of IT, University of Sydney, Australia.
E-mail address: joachim.gudmundsson@sydney.edu.au (J. Gudmundsson).

1 Supported by the King Abdulaziz City for Science and Technology (KACST) under Grant number 11-INF1990-03.
2 Funded by the Australian Research Council FT100100755.
3 Research partially supported by VR Grants 621-2008-4649 and 621-2011-6179.
4 NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

0925-7721/$ – see front matter © 2014 Published by Elsevier B.V.
http://dx.doi.org/10.1016/j.comgeo.2014.01.004

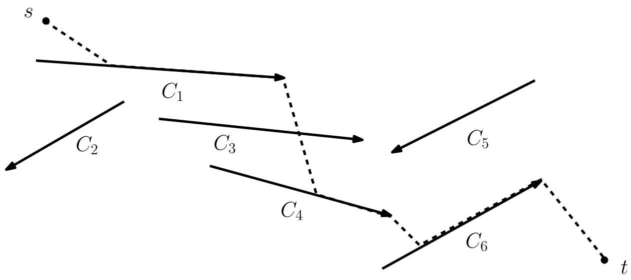

Fig. 1. Illustrating a shortest path from a source point s to a destination point t.

the road. This, as Aichholzer et al. [3] pointed out, makes the problem more difficult than those in other similar settings such as the airlift distance [4]. In the presence of a transportation network, the *transportation distance* between two points is defined to be the shortest path among all possible paths joining the two points, sometimes also referred to as the *quickest path* between two points. The induced distance metric on $\mathcal{R}^2$, called $d_\mathcal{T}$, is called a transportation metric.

Using these notations the quickest path problem is defined as follows (see also Fig. 1):

Problem 1. Given two points s and t in $\mathbb{R}^2$ and a transportation network $\mathcal{T}(S,\mathcal{C})$ in the Euclidean plane. The problem is to find a path with the smallest transportation distance from s to t, as shown in Fig. 1.

The second problem we consider is the nearest neighbor query and its variants. This problem is well studied in the literature, and its variants include the k-nearest neighbor query [19] and the k-path nearest neighbor query [11]. The nearest neighbor query in a transportation network is formally defined as follows:

Problem 2. Given a point $q \in \mathbb{R}^2$, a set $\mathcal{Z}$ of m sites, and a transportation network $\mathcal{T}(S,\mathcal{C})$ in the Euclidean plane. The problem is to find a closest site $z \in \mathcal{Z}$ with smallest transportation distance from q.

In this paper we introduce the (approximate) query version of Problem 1. That is, given a transportation network $\mathcal{T}$ with n roads in the Euclidean plane and a positive constant ε, preprocess $\mathcal{T}$ into a data structure such that given any two points s and t in the plane, the cost of a $(1+\varepsilon)$-approximate quickest path between s and t can be answered efficiently. The exact complexity bounds are given in Theorem 7. In the case when all the edge weights are bounded from below by $\alpha_{\min}$ and from above by $\alpha_{\max}$ and both are constants independent of n then $\mathcal{T}$ can be preprocessed in $O((\frac{n}{\varepsilon})^2 \log n)$ time into a data structure of size $O((\frac{n}{\varepsilon})^2)$ such that $(1+\varepsilon)$-approximate queries can be answered in $O(\frac{1}{\varepsilon^4} \cdot \log n)$ time. In the case when the path should be reported the query time increases to $O(\frac{1}{\varepsilon^4} \cdot \log n + L)$ time, where L is the number of segments along the reported path.

We also introduce the (approximate) query version of Problem 2. That is, a transportation network with n roads and m sites in the Euclidean plane can be preprocessed in $O((n^2+nm)\log(m+n))$ time such that given a query point q in the plane a $(1+\varepsilon)$-approximate nearest neighbor site to q can be answered in $O(\frac{1}{\varepsilon^2} \cdot \log(n+m))$ time; the exact bound is stated in Theorem 9.

Most of the previous work has focussed on shortest paths and Voronoi diagrams. Gewali et al. [13] gave the basis for several structural results exploited in later papers (including this paper), see also [16,17]. They [13] gave an $O(n^2 \log n)$ algorithm using $O(n^2)$ space for Problem 1. Bae and Chwa [5] showed how to construct a Voronoi diagram and how to compute a shortest path. This result was later extended to more general metrics including asymmetric convex distance functions [6].

Abellanas et al. [1,2] considered the Voronoi diagram of a point set and shortest paths given a horizontal highway under the L_1-metric and the Euclidean metric. Aichholzer et al. [3] introduced the city metric induced by the L_1-metric and a highway network that consists of a number of axis-parallel line segments. They gave an $O(m \log m + c^2 \log c)$ time algorithm using $O(m+c)$ space for constructing the Voronoi diagram, where c is the number of nodes on the highway network. Using the diagram, a quickest paths query from the query point to a site can be answered in $O(\log(m+c)+r)$ time, where r is the complexity of the reported path. Görke et al. [14] improved the results by Aichholzer et al. [3]. They gave an $O((c+m)\log(c+m))$ time algorithm for the construction of the Voronoi diagram using a wavefront expansion. Bae et al. [7] later generalized these results.

This paper is organized as follows. Next we review three fundamental properties of an optimal path among a set of roads [13]. Then, in Section 3, we show how we can use these properties to build a graph that models the transportation network, a source point s, a destination point t, and contains a quickest path between s and t. In Section 4 we consider the query version of the problem. That is, preprocess the input such that an approximate quickest path cost query between two query points s and t can be answered efficiently. In Section 5 we consider the nearest neighbor query and its variants. Finally, we conclude with some remarks and open problems.