Theoretical Computer Science 607 (2015) 2–15

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Multiple sink location problems in dynamic path networks ☆

Yuya Higashikawa [a,*,1], Mordecai J. Golin [b], Naoki Katoh [a,2]

[a] Department of Architecture and Architectural Engineering, Kyoto University, Japan
[b] Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong

ARTICLE INFO

Article history:
Received 1 October 2014
Received in revised form 1 May 2015
Accepted 4 May 2015
Available online 6 June 2015

Keywords:
Sink location
Dynamic network
Evacuation planning

ABSTRACT

This paper considers the k-sink location problem in dynamic path networks. A dynamic path network consists of an undirected path with positive edge lengths, uniform edge capacity, and positive vertex supplies. A path can be considered as a road, edge lengths as the distance along a road segment and vertex supplies as the number of people at a location. The edges all have a fixed common *capacity*, which limits the number of people that can enter that edge in a unit of time. The problem is to find the optimal location of k sinks (exits) on the path such that each evacuee is sent to one of the k sinks. The existence of capacities causes congestion, which can slow evacuation down in unexpected ways.

Let $\boldsymbol{x}$ be a vector denoting the location of the k sinks. The optimal evacuation policy for $\boldsymbol{x}$ is $(k-1)$-dimensional vector $\boldsymbol{d}$, called $(k-1)$-*divider*. Each component of $\boldsymbol{d}$ corresponds to a boundary dividing all evacuees between adjacent two sinks into two groups, i.e., all supplies to the right of the boundary evacuate to the right sink and all the others to the left sink. In this paper, we consider *optimality* defined by two different criteria, the minimax criterion and the minisum one. We prove that the minimax problem can be solved in $O(kn)$ time and the minisum problem in $O(n^2 \cdot \min\{\sqrt{k \log n} + \log n, 2^{\sqrt{\log k \log \log n}}\})$ time, where n is the number of vertices in the given network.

1. Introduction

The Tohoku-Pacific Ocean Earthquake occurred on March 11, 2011. Many people in Japan failed to evacuate from the coast and lost their lives to the resultant tsunamis. From the viewpoint of city planning it has now become extremely important to establish effective evacuation planning for large scale disasters. In particular, arrangements for designating evacuation buildings in large Japanese cities near the coast where people can go in case of tsunami warnings have become urgent issues. To determine appropriate tsunami evacuation buildings, we need to consider both where the evacuation buildings should be located and who will evacuate to which building, i.e., how to partition a large area into small regions so that everyone in a particular region evacuates to a designated building.

Modeled as a graph, this problem leads to some very interesting algorithmic questions. This paper focuses on the problem of locating the evacuation buildings in the simplest case of the region being restricted to a single path (road).

☆ A preliminary version of this paper appeared in the proceedings of AAIM 2014 [5].

* Corresponding author.
E-mail address: as.higashikawa@archi.kyoto-u.ac.jp (Y. Higashikawa).

1 Supported by JSPS Grant-in-Aid for JSPS Fellows (26·4042).
2 Supported by JSPS Grant-in-Aid for Scientific Research (A) (25240004).

http://dx.doi.org/10.1016/j.tcs.2015.05.053
0304-3975/

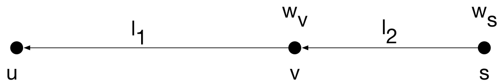

Fig. 1. Illustration of evacuation on a two edge path. All supplies on vertices v and s evacuate to a vertex u. The problem starts with w_v units of supply at vertex v and w_s units of supply at vertex s. All edges have capacity c.

The problem can be modeled in the *dynamic* setting in graph networks, which was first introduced by Ford et al. [2] in the context of maximum dynamic flows, studying how long it would take to move flow across a network. In a graph network under the dynamic setting, each vertex has an associated supply and each edge has both a length and a capacity which limits the rate of the flow into the edge per unit time. Such networks are called *dynamic networks*.

Dynamic networks can be considered in both discrete and continuous models. In the discrete model, all values are integers. Each supply can be regarded as a set of evacuees, and edge capacity is defined as the maximum number of evacuees who can enter an edge per unit time. In the continuous model, each input value is given as a real number. Supply can be regarded as fluid, and edge capacity is defined as the maximum amount of supply which can enter an edge per unit time.

Unlike in the dynamic max flow problem which can split flow along different paths, our models assume that all supply from a vertex gets sent along the same path (and therefore all go to the same sink). This is because in the evacuation context, all people at a location should follow the same evacuation plan.

Also, unlike in static problems, the time required to move supply from one vertex to a sink can be increased due to congestion caused by the capacity constraints, which require supplies to wait at vertices until supplies preceding them have left.

The k-sink location problem in dynamic networks is defined as the problem of finding the optimal location of k sinks in a given network (along with the paths that supplies at each vertex will follow) so that all supplies of all vertices are sent to one of the k sinks in the shortest time. The definition of *shortest* in the above is slightly ambiguous. There are two natural definitions: *maximum cost criterion* and *total cost criterion*.

We illustrate these definitions in the 1-sink location problem and give formal definitions for general k-sink problem later. Let x be the location of the single sink. In the discrete version the cost of x for a particular evacuee is defined as the minimum time required to send that evacuee to x. Note that due to congestion this cost can depend upon the paths taken by supplies from other vertices. The *maximum cost* is just the maximum evacuation time over all evacuees. The *total cost* is the sum of all of the evacuees evacuation times.

In the continuous model, we define a *unit* as an infinitesimally small portion of supply with evacuation cost being defined on each unit. The cost of x for a unit is defined as the minimum time required to send the unit to x. The *maximum cost* and *total cost* are then naturally the maximum cost and sum of costs over all units.

We remark that our minimax (resp. minisum) sink location problem are generalizations of the NP-Hard unweighted k*-center* (resp. k*-median*) problem in static networks [8]. More specifically, if each edge capacity is sufficiently large, our problems reduce to the k-center or k-median problem, respectively.

Recently, several researchers have studied the minimax sink location problem in dynamic networks. Mamada et al. [7] studied the minimax 1-sink location problem in dynamic tree networks assuming that the sink must be located at a vertex, and proposed an $O(n\log^2 n)$ time algorithm. Higashikawa et al. [4] also studied the same problem as [7] assuming that edge capacity is uniform and the sink can be located at any point in a network, and proposed an $O(n\log n)$ time algorithm. However, the minisum sink location problem in dynamic networks has never been studied so far.

In this paper, we study the k-sink location problems in dynamic path networks in the continuous model assuming that edge capacities are uniform, i.e., all edge capacities are identical. We also allow the sinks to be located at any point in a network, i.e., they can lie on an edge and do not have to be one of the path vertices. We prove that the minimax problem can be solved in $O(kn)$ time and the minisum problem can be solved in $O(n^2 \cdot \min\{\sqrt{k\log n}+\log n, 2^{\sqrt{\log k \log\log n}}\})$ time. Note that this study is the first one which treats the minisum sink location problem in dynamic networks and also gives an exact algorithm for the minimax k-sink location problem in dynamic networks.

2. Minimax k-sink location problem

2.1. Preliminaries

Model definition: Let $P=(V,E)$ be an undirected path with ordered vertices $V=\{v_1,v_2,\dots,v_n\}$ and edges $E=\{e_1,e_2,\dots,e_{n-1}\}$ where $e_i=(v_i,v_{i+1})$ for $1\le i\le n-1$. Let $\mathcal{N}=(P,l,w,c,\tau)$ be a dynamic network with the underlying path graph P; l is a function that associates each edge e_i with positive length l_i, w is a function that associates each vertex v_i with positive weight w_i, amount of supply at v_i; c is the capacity, a positive constant representing the amount of supply which can enter an edge per unit time; τ is a positive constant representing the time required for a flow to travel distance one. We call such networks with path structures *dynamic path networks*.

As a simple illustration, let us consider Fig. 1. All supplies on vertices v and s are both evacuating to a vertex u. It takes time τl_1 for a unit of supply to cross the edge (u,v) and τl_2 to cross the edge (v,s). Since the amount of supply which