European Journal of Operational Research 250 (2016) 360–366

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

Minimax regret 1-sink location problem with accessibility in dynamic general networks

Hongmei Li[a,b], Yinfeng Xu[b,c,*]

[a] School of Economic and Management, Northwest University, Xi'an 710069, China
[b] Business School, Sichuan University, Chengdu 610065, China
[c] State Key Lab for Manufacturing Systems Engineering, Xi'an 710049, China

ARTICLE INFO

Article history:
Received 9 January 2015
Accepted 12 September 2015
Available online 24 September 2015

Keywords:
Accessibility
Algorithm
General network
Location
Minimax regret

ABSTRACT

In this study, we consider the minimax regret 1-sink location problem with accessibility, where all of the weights should always evacuate to the nearest sink point along the shortest path in a dynamic general network with positive edge lengths and uniform edge capacity. Let $G = (V, E)$ be an undirected connected simple graph with n vertices and m edges. Each vertex has a weight that is not known exactly but the interval to which it belongs is given. A particular assignment of a weight to each vertex is called a scenario. The problem requires that a point x should be found as the sink on the graph and all the weights evacuate to x such that the maximum regret for all possible scenarios is minimized. We present an $O(m^2n^3)$ time algorithm to solve this problem.

1. Introduction

The minimax regret sink location problem was first proposed by Cheng et al. (2013) in the context of the Tohuku-Pacific Ocean Earthquake, which occurred in Japan on March 11, 2011. In general, the problem can be described as follows. We are given an undirected connected graph $G = (V, E)$, $|V| = n$ and $|E| = m$. Each vertex is associated with a vertex weight that is not known exactly but the interval to which it belongs is known. Each edge is associated with an edge length and an edge capacity. The edge length is the distance between the two endpoints of the edge and the edge capacity is the upper boundary for the number of weights that enter an edge per unit time. The objective is to find $k(k \geq 1)$ sink points on G such that all the weights are sent to one of the k sinks and the *maximum regret* of the *evacuation time* is minimized. In this case, the evacuation time usually refers to the maximum time, i.e., the time required by the last unit weight to complete evacuation.

Cheng et al. (2013) considered the minimax regret 1-sink location problem in a *dynamic path network* with a uniform edge capacity, where an $O(n\log^2 n)$ time algorithm was proposed for the problem. Later, Wang (2014) and Higashikawa et al. (2015) proposed two improved algorithms to address the problem, which each had a time complexity of $O(n\log n)$. It is noteworthy that the algorithm proposed by Wang (2014) further reduced the space complexity degree of previous algorithms. Li, Xu, and Ni (2014) considered the minimax regret vertex 2-sink location problem in dynamic path networks with a uniform edge capacity, where the sinks are limited to being located on the vertex of the path, and an $O(n^3\log n)$ time algorithm was proposed. For a general k, Ni, Xu, and Dong (2014) proposed an $O(n^{1+k}(\log n)^{1+\log k})$ time algorithm, whereas Arumugam, Augustine, Golin, and Strikanthan (2014) proposed two algorithms for this problem. The first algorithm proposed by Arumugam et al. (2014) is better for small values of k, where it runs in $O(kn^2(\log n)^k)$, whereas the second algorithm is better for larger values of k, where it runs in $O(kn^3\log n)$. It is noteworthy that Arumugam et al.'s algorithm is the first polynomial time algorithm for this problem. For other networks, Higashikawa, Golin, and Katoh (2014a) considered the minimax regret 1-sink location problem in dynamic tree networks with a uniform edge capacity and they proposed an $O(n^2\log^2 n)$ time algorithm. For the minimax regret 1-sink location problem in dynamic cycle networks with a uniform edge capacity, Xu and Li (2015) proposed an $O(n^3\log n)$ time algorithm. Bhattacharya and Kameda (2015) proposed several algorithms which improved the previous best algorithms for several minimax regret sink location problems, including an $O(n)$ time algorithm for the minimax regret 1-sink location problem in a dynamic path network, an $O(n\log n^4)$ time algorithm for the minimax regret 2-sink location problem in a dynamic path network, and an $O(n\log n)$ time algorithm for the minimax regret 1-sink location problem in a dynamic tree network.

When the exact weight of each vertex is known, the objective of the sink location problem is to find the sink location in the given network that minimizes the evacuation time. For path networks,

* Corresponding author at: Business School, Sichuan University, Chengdu 610065, China. Tel./fax: +86 28 85417993.
E-mail address: yfxu@scu.edu.cn, yfxu@mail.xjtu.edu.cn (Y. Xu).

http://dx.doi.org/10.1016/j.ejor.2015.09.022
0377-2217/

Higashikawa, Golin, and Katoh (2014b) solved the k-sink location problem in dynamic path networks with a uniform edge capacity in $O(kn\log n)$ time for a general k and proposed an improved algorithm with a time complexity of $O(kn)$ (Higashikawa et al., 2014b). Tree networks have been addressed in several studies. Mamada, Makino, and Fujishige (2002) considered the vertex 1-sink location problem in a dynamic tree network with general capacities while an $O(n\log^2 n)$ time algorithm for this problem was proposed by Mamada, Uno, Makino, and Fujishige (2006). Higashikawa et al. (2014a) described an $O(n\log n)$ time algorithm for the problem when the capacities are uniform. In addition, it should be mentioned that the studies of Higashikawa et al. (2014b) and Higashikawa, Golin, and Katoh (2014c) also considered a situation where the evacuation time was denoted as the total time instead of the maximum time.

If there is only one sink and the route to the sink is unique for any vertex, it is obvious that the weights on a vertex will evacuate to the sink along that unique route. If there is more than one sink or the route from a vertex to a sink is not unique, then we need to make a decision such that the *evacuation solution* (which includes the objective sink and the corresponding evacuation path) is defined for every vertex. In previous related studies (Arumugam et al., 2014; Bhattacharya & Kameda, 2015; Higashikawa et al., 2014b; 2014c; Li et al., 2014; Ni et al., 2014; Xu & Li, 2015), the *division point* is set to the optimal value, i.e., the weight of any vertex will evacuate to the sink(s) with the *optimal evacuation solution*, which allows the weights to evacuate to more distant sinks rather than the nearest and/or complete evacuation along a path that is not the shortest path. It is assumed that the weights on a vertex should evacuate to the same sink and we also retain this assumption in this paper. Thus, the optimal evacuation solution is the solution with the minimum evacuation time for the weights between any two neighboring sink points. However, when an emergency, such as an earthquake, occurs, people do not behave as expected during the evacuation process and they will usually evacuate to the nearest shelter via the shortest path (known as the *intuitive evacuation solution*) rather than evacuating according to the optimal evacuation solution. Indeed, the optimal evacuation solution might not be known when an emergency occurs.

Thus, in this paper, we consider the minimax regret sink location problem with accessibility where all of the weights are assumed to evacuate to the nearest sink via the shortest path. We consider this problem in a general network $G = (V, E)$ where the number of sink points is 1. Throughout this study, we assume that each edge has a uniform edge capacity. As usual, and without loss of generality, we also assume that the undirected connected graph G contains no loops or multiple edges, i.e., the graph G is an undirected connected simple graph. Given a dynamic general network G, the weight of every vertex is known as an interval. A specific assignment of a weight to each vertex is called a *scenario*. The problem involves finding a point x on G as the sink point such that the maximum regret of the evacuation time should be as small as possible. The remainder of this paper is organized as follows. In Section 2, we provide some useful definitions and properties. In Section 3, we present an $O(m^2n^3)$ time algorithm for the minimax regret 1-sink location problem in a dynamic general network, and we present our conclusions in Section 4.

2. Definitions and preliminaries

2.1. Definitions

Let $G = (V, E)$ be an undirected connected simple graph where $V = \{v_1, \ldots, v_n\}$. A positive number $l(e)$ (called the length of e) and a positive number c (called the uniform edge capacity) are associated with each of the $|E| = m$ edges, and an interval $W(v_i) = \left[\underline{w_i}, \overline{w_i}\right]$ with $0 < \underline{w_i} \le \overline{w_i}$ (called the weight interval of v_i) is associated with each of the $|V| = n$ vertices. Let $\mathcal{S}$ denote the Cartesian product of all $W(v_i)$ for $1 \le i \le n$. When a scenario $s \in \mathcal{S}$ is given, we use the notation $w_i(s)$ to denote the weight of each vertex $v_i \in V$ under the scenario s. Let τ be a constant that represents the time required to traverse the unit distance of each unit weight. We also use a notation G to denote the set of all points on G. The distance between any two points x and y on G, which is represented by $d_{x,y}$ (note that $d_{y,x} = d_{x,y}$), is the length of the shortest path on G from x to y. We assume that the distance between any two vertices is known. In fact, the distance matrix (which gives the distance between any two vertices) of a general graph can be obtained in $O(n^3)$ time with the Floyd–Warshall algorithm. Of course, there are algorithms which are better than $O(n^3)$ time and for more details please refer to Williams (2014). Since the time required to determine the distance matrix has no effect on the time complexity of our algorithm, in this paper we just assume the distance matrix is known. Because G is connected, $m \ge n - 1$ holds. When $m = n - 1$, G is a tree, and thus we assume that $m \ge n$ in this study.

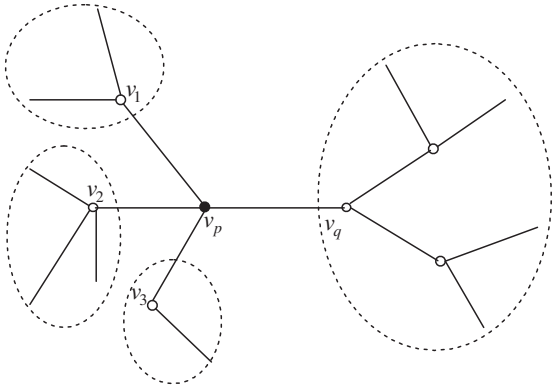

Fig. 1. Illustration of $T(x, s)$ where $x = v_p$.

Next, we define the time required for the evacuation of all weights to $x \in G$ under $s \in \mathcal{S}$, i.e., $T(x, s)$. Because there is only one sink and all of the weights are assumed to complete their evacuation along the shortest path, then the evacuation solution for every vertex is provided as long as the sink point is given. In other words, with a given sink point x, the general graph G can be viewed as a tree graph $G(x)$ rooted at x, based on which the evacuation paths for all of the vertices are represented. For any point x, in $G(x)$ we define the vertices adjacent to x as neighbors of x, which are denoted as N_x. Thus, for a vertex v_i, $N_{v_i} = \left\{v_j \mid e = \left(v_i, v_j\right) \text{ exists in } G(x)\right\}$, and for a point $x \in e$, if we assume that $e = (v_p, v_q)$, $x \ne v_p$, and $x \ne v_q$, then $N_x = \{v_p, v_q\}$. Note that when we state that $x \in e = (v_p, v_q)$, we always mean the points on e except for the two endpoints v_p and v_q, unless indicated otherwise. With a given sink x, for any vertex $v_i \in G$, the evacuation path of v_i must contain the vertex $v_j \in N_x$. Let $G(x, v_i)$ be a subtree of $G(x)$ rooted at v_i and $T^{v_i}(x, s)$ is the evacuation time required for all of the weights on $G(x, v_i)$ to complete their evacuation to x under scenario s. Then, we have

$$T(x, s) = \max_{v_i \in N_x} \left\{T^{v_i}(x, s)\right\}. \tag{1}$$

For example, as shown in Fig. 1, $T(v_p, s) = \max\left\{T^{v_1}(v_p, s), T^{v_2}(v_p, s), T^{v_3}(v_p, s), T^{v_q}(v_p, s)\right\}$. If $x \in e = (v_p, v_q)$, then $T(x, s) = \max\left\{T^{v_p}(x, s), T^{v_q}(x, s)\right\}$. We assume that $v_i \in N_x$, $T^{v_i}(x, s)$ can be obtained by transiting $G(x, v_i)$ to a path (Kamiyama, Katoh, & Takizawa, 2006). Suppose that n' vertices in $G(x, v_i)$ are denoted as $u_1, u_2, \ldots, u_{n'} (= v_i)$ such that $d_{x,u_i} \le d_{x,u_{i-1}}$ for $2 \le i \le n'$. Kamiyama et al. (2006) proved that $T^{v_i}(x, s)$ does not change if x and u_i for $1 \le i \le n'$ are relocated on a line with the same capacity such that d_{x,u_i} remains unchanged. Thus, $T^{v_i}(x, s)$ can be represented as follows:

$$T^{v_i}(x, s) = \max_{1 \le j \le n'} \left\{\tau \cdot d_{x,u_j} + \frac{\sum_{1 \le l \le j} w_l(s)}{c}\right\}. \tag{2}$$