

# Minmax Regret 1-Sink Location Problems on Dynamic Flow Path Networks with Parametric Weights

TETSUYA FUJIE<sup>1</sup>   YUYA HIGASHIKAWA<sup>1</sup>   NAOKI KATOH<sup>1</sup>   JUNICHI TERUYAMA<sup>1</sup>  
YUKI TOKUNI<sup>2</sup>

<sup>1</sup>School of Social Information Science, University of Hyogo, Japan;  
{fujie, higashikawa, naoki.katoh, junichi.teruyama}@sis.u-hyogo.ac.jp

<sup>2</sup>School of Science and Technology, Kwansai Gakuin University, Japan;  
enj32048@kwansai.ac.jp

## Abstract

This paper addresses the minmax regret 1-sink location problem on dynamic flow path networks with parametric weights. We are given a *dynamic flow network* consisting of an undirected path with positive edge lengths, positive edge capacities, and nonnegative vertex weights. A path can be considered as a road, an edge length as the distance along the road and a vertex weight as the number of people at the site. An edge capacity limits the number of people that can enter the edge per unit time. We consider the problem of locating a *sink* in the network, to which all the people evacuate from the vertices as quickly as possible. In our model, each weight is represented by a linear function in a common parameter  $t$ , and the decision maker who determines the location of a sink does not know the value of  $t$ . We formulate the sink location problem under such uncertainty as the *minmax regret problem*. Given  $t$  and a sink location  $x$ , the cost of  $x$  under  $t$  is the sum of arrival times at  $x$  for all the people determined by  $t$ . The regret for  $x$  under  $t$  is the gap between the cost of  $x$  under  $t$  and the optimal cost under  $t$ . The task of the problem is formulated as the one to find a sink location that minimizes the maximum regret over all  $t$ . For the problem, we propose an  $O(n^4 2^{\alpha(n)} \alpha(n) \log n)$  time algorithm where  $n$  is the number of vertices in the network and  $\alpha(\cdot)$  is the inverse Ackermann function. Also for the special case in which every edge has the same capacity, we show that the complexity can be reduced to  $O(n^3 2^{\alpha(n)} \alpha(n) \log n)$ .

## 1 Introduction

Recently, many disasters, such as earthquakes, nuclear plant accidents, volcanic eruptions and flooding, have struck in many parts of the world, and it has been recognized that orderly evacuation planning is urgently needed. A powerful tool for evacuation planning is the *dynamic flow model* introduced by Ford and Fulkerson [13], which represents movement of commodities over time in a network. In this model, we are given a graph with *source* vertices and *sink* vertices. Each source vertex is associated with a positive weight, called a *supply*, each sink vertex is associated with a positive weight, called a *demand*, and each edge is associated with positive length and capacity. An edge capacity limits the amount of supply that can enter the edge per unit time. One variant of the dynamic flow problem is the *quickest transshipment problem*, of which the objective is to send exactly the right amount of supply out of sources into sinks with satisfying the demand constraints in the minimum overall time. Hoppe and Tardos [22] provided a polynomial time algorithm for this problem in the case where the transit times are integral. However, the complexity of their algorithm is very high. Finding a practical polynomial time solution to this problem is still open. A reader is referred to a recent survey by Skutella [27] on dynamic flows.

This paper discusses a related problem, called the *sink location problem* [4, 6, 7, 8, 11, 12, 20, 21, 26], of which the objective is to find a location of sinks in a given dynamic flow network so that all the supply is sent to the sinks as quickly as possible. For the optimality of location, the following two criteria can be naturally considered: the minimization of *evacuation completion time* and *aggregate evacuation time* (i.e., *sum of evacuation times*). We call the sink location problem

that requires finding a location of sinks on a dynamic flow network that minimizes the evacuation completion time (resp. the aggregate evacuation time) the CTSL problem (resp. the ATSL problem). Several papers have studied the CTSL problems [4, 8, 11, 12, 20, 21, 26]. On the other hand, for the ATSL problems, we have a few results only for path networks [6, 7, 21].

In order to model the evacuation behavior of people, it might be natural to treat each supply as a discrete quantity as in [22, 26]. Nevertheless, almost all the previous papers on sink location problems [4, 8, 11, 12, 20, 21] treat each supply as a continuous quantity since it is easier for mathematically handling the problems and the effect of such treatment is small enough to ignore when the number of people is large. Throughout the paper, we adopt the model with continuous supplies.

Although the above two criteria are reasonable, they may not be practical since the population distribution is assumed to be fixed. In a real situation, the number of people in an area may vary depending on the time, e.g., in an office area in a big city, there are many people during the daytime on weekdays while there are much less people on weekends or during the night time. In order to take such the uncertainty into account, Kouvelis and Yu [23] introduced the *minmax regret model*. In the *minmax regret sink location problems*, we are given a finite or infinite set  $S$  of *scenarios*, where each scenario gives a particular assignment of weights on all the vertices. Here, for a sink location  $x$  and a scenario  $s \in S$ , we denote the evacuation completion time or aggregate evacuation time by  $F(x, s)$ . Then, the problem can be understood as a 2-person Stackelberg game as follows. The first player picks a sink location  $x$  and the second player chooses a scenario  $s \in S$  that maximizes the regret defined as  $R(x, s) := F(x, s) - \min_x F(x, s)$ . The objective of the first player is to choose  $x$  that minimizes the maximum regret. Throughout the paper, we call the minmax regret sink location problem, where the regret is defined with the evacuation completion time (resp. the aggregate evacuation time), the MMR-CTSL problem (resp. the MMR-ATSL problem). The MMR-CTSL problems have been studied so far [3, 10, 14, 18, 20, 24, 25]. On the other hand, for the MMR-ATSL problems, we have few results [9, 19] although the problems are also important theoretically and practically.

As for how to define a set of scenarios, all of the previous studies on the minmax regret sink location problems adopt the model with *interval weights*, in which each vertex is given the weight as a real interval, and a scenario is defined by choosing an element of the Cartesian product of all the weight intervals over the vertices. One drawback of the minmax regret model with interval weights is that each weight can take an independent value, thus we consider some extreme scenarios which may not happen in real situations, e.g, a scenario where all the vertices have maximum weights or minimum weights. To incorporate the dependency among weights of all the vertices into account, we adopt the model with *parametric weights* (first introduced by Vairaktarakis and Kouvelis [28] for the minmax regret median problem), in which each vertex is given the weight as a linear function in a common parameter  $t$  on a real interval, and a scenario is just determined by choosing  $t$ . Note that considering a real situation, each weight function should be more complex, however, such a function can be approximated by a piecewise linear function. Thus superimposing all such piecewise linear functions, it turns out that for a sufficiently small subinterval of  $t$ , every weight function can be regarded as linear, and by solving multiple subproblems with linear weight functions, we can obtain the solution.

In this paper, we study the MMR-ATSL problem on dynamic flow path networks with parametric weights. Our main theorem is below.

**Theorem 1** (Main Results). *Suppose that we are given a dynamic flow path network of  $n$  vertices with parametric weights.*

- (i) *The MMR-ATSL problem can be solved in time  $O(n^4 2^{\alpha(n)} \alpha(n) \log n)$ , where  $\alpha(\cdot)$  is the inverse Ackermann function.*

(ii) When all the edge capacities are uniform, the MMR-ATSL problem can be solved in time  $O(n^3 2^{\alpha(n)} \alpha(n) \log n)$ .

Note that the MMR-ATSL problem with interval weights is studied by [9, 19], and only for the case with the uniform edge capacity, Higashikawa et al. [19] provide an  $O(n^3)$  time algorithm, which is improved to one running in  $O(n^2 \log^2 n)$  time by [9]. However, for the case with general edge capacities, no algorithm has been known so far. Therefore, our result implies that the problem becomes solvable in polynomial time by introducing parametric weights.

The rest of the paper is organized as follows. In Section 2, we give the notations and the fundamental properties that are used throughout the paper. In Section 3, we give the key lemmas and the algorithms that solves the problems, which concludes the paper.

## 2 Preliminaries

For two real values  $a, b$  with  $a < b$ , let  $[a, b] = \{t \in \mathbb{R} \mid a \leq t \leq b\}$ ,  $(a, b) = \{t \in \mathbb{R} \mid a < t < b\}$ , and  $(a, b] = \{t \in \mathbb{R} \mid a < t \leq b\}$ .

In our problem, we are given a real interval  $T = [t^-, t^+] \subset \mathbb{R}$  and a dynamic flow path network  $\mathcal{P} = (P, \mathbf{w}(t), \mathbf{c}, \mathbf{l}, \tau)$ , which consists of five elements:  $P = (V, E)$  is a path with vertex set  $V = \{v_i \mid 1 \leq i \leq n\}$  and edge set  $E = \{e_i = (v_i, v_{i+1}) \mid 1 \leq i \leq n-1\}$ ,  $\mathbf{w}(t)$  is a vector  $\langle w_1(t), \dots, w_n(t) \rangle$  of which component  $w_i(t)$  is a *weight function*  $w_i : T \rightarrow \mathbb{R}_{\geq 0}$  which is linear in a parameter  $t$  and nonnegative for any  $t \in T$ , a vector  $\mathbf{c} = \langle c_1, \dots, c_{n-1} \rangle$  consists of the *capacity*  $c_i$  of edge  $e_i$ , a vector  $\mathbf{l} = \langle \ell_1, \dots, \ell_{n-1} \rangle$  consists of the *length*  $\ell_i$  of edge  $e_i$ , and  $\tau$  is the *time* which the supply takes to move a unit distance on any edge. Let us explain how edge capacities and lengths affect the evacuation time. Consider an evacuation under fixed  $t \in T$ . Suppose that at time 0, the amount  $w$  of supply is at vertex  $v_{i+1}$  and going through edge  $e_i$  towards vertex  $v_i$ . The first fraction of supply from  $v_{i+1}$  can arrive at  $v_i$  at time  $\tau \ell_i$ . The edge capacity  $c_i$  represents the maximum amount of supply which can enter  $e_i$  in a unit time interval, so all the supply  $w$  can complete leaving  $v_{i+1}$  at time  $w/c_i$ . Therefore, all the supply  $w$  can complete arriving at  $v_i$  at time  $\tau \ell_i + w/c_i$ .

For any integers  $i, j$  with  $1 \leq i \leq j \leq n$ , we denote the sum of weights from  $v_i$  to  $v_j$  by  $W_{i,j}(t) = \sum_{h=i}^j w_h(t)$ . For the notation, we define  $W_{i,j}(t) = 0$  for  $i, j$  with  $i > j$ . For a vertex  $v_i \in V$ , we abuse  $v_i$  to denote the distance between  $v_1$  and  $v_i$ , i.e.,  $v_i = \sum_{j=1}^{i-1} \ell_j$ . For an edge  $e_i \in E$ , we abuse  $e_i$  to denote a real open interval  $(v_i, v_{i+1})$ . We also abuse  $P$  to denote a real closed interval  $[0, v_n]$ . If a real value  $x$  satisfies  $x \in (v_i, v_{i+1})$ ,  $x$  is said to be a point on edge  $e_i$  to which the distance from  $v_i$  is value  $x - v_i$ . Let  $C_{i,j}$  be the minimum capacity for all the edges from  $e_i$  to  $e_j$ , i.e.,  $C_{i,j} = \min\{c_h \mid i \leq h \leq j\}$ .

Note that we precompute values  $v_i$  and  $W_{1,i}(t)$  for all  $i$  in  $O(n)$  time, and then,  $W_{i,j}(t)$  for any  $i, j$  can be obtained in  $O(1)$  time as  $W_{i,j}(t) = W_{1,j}(t) - W_{1,i-1}(t)$ . In addition,  $C_{i,j}$  for any  $i, j$  can be obtained in  $O(1)$  time with  $O(n)$  preprocessing time, which is known as the *range minimum query* [2, 5].

### 2.1 Evacuation Completion Time on a Dynamic Flow Path Network

In this section, we see the details of evacuation phenomenon using a simple example, and eventually show the general formula of evacuation completion time on a path, first provided by Higashikawa [17]. W.l.o.g., an evacuation to a sink  $x$  follows the first-come first-served manner at each vertex, i.e., when a small fraction of supply arrives at a vertex  $v$  on its way to  $x$ , it has to wait for the departure if there already remains some supply waiting for leaving  $v$ .

Let us consider an example with  $|V| = 3$  where  $V = \{v_1, v_2, v_3\}$ ,  $E = \{e_1 = (v_1, v_2), e_2 = (v_2, v_3)\}$ . Assume that the sink  $x$  is located at  $v_1$ , and under a fixed parameter  $t \in T$ , the amount of supply at  $v_i$  is  $w_i$  for  $i = 2, 3$ .

All the supply  $w_1$  at  $v_1$  immediately completes its evacuation at time 0 and we send all the supply  $w_2$  and  $w_3$  to  $v_1$  as quickly as possible. Let us focus on how the supply of  $v_3$  moves to  $v_1$ . First, the foremost fraction of supply from  $v_3$  arrives at  $v_2$  at time  $\tau\ell_2$ , and all the supply  $w_3$  completes leaving  $v_3$  at time  $w_3/c_2$ , i.e., it completes arriving at  $v_2$  at time  $\tau\ell_2 + w_3/c_2$ . Suppose that at time  $\tau\ell_2 + w_3/c_2$ , the amount  $w'(\geq 0)$  of supply remains at  $v_2$ . From then on, the time required to send all the supply  $w'$  to  $v_1$  is  $\tau\ell_1 + w'/c_1$ . Thus, the evacuation completion time is expressed as

$$\tau(\ell_1 + \ell_2) + \frac{w_3}{c_2} + \frac{w'}{c_1}. \quad (1)$$

We observe what value  $w'$  takes in the following cases.

**Case 1: It holds  $c_1 \geq c_2$ .** In this case, the amount of supply at  $v_2$  should be non-increasing, because the amount  $c_1$  of supply leaves  $v_2$  and the amount at most  $c_2$  of supply arrives at  $v_2$  per unit time. Let us consider the following two situations at time  $\tau\ell_2 + w_3/c_2$ : When all the supply  $w_3$  completes arriving at  $v_2$ , there remains no supply at  $v_2$ , that is,  $w' = 0$  holds or not. If  $w' = 0$  holds, then substituting it into (1), the evacuation completion time is expressed as

$$\tau(\ell_1 + \ell_2) + \frac{w_3}{c_2}. \quad (2)$$

Otherwise, that is  $w' > 0$  holds, there remains a certain amount of supply at  $v_2$  even at time  $\tau\ell_2$  since the amount of supply at  $v_2$  is non-increasing. Thus at time  $\tau\ell_2$ , the amount  $w_2 - c_1\tau\ell_2$  of supply remains at  $v_2$ . From time  $\tau\ell_2$  to time  $\tau\ell_2 + w_3/c_2$ , the amount of supply waiting at  $v_2$  decreases by  $c_1 - c_2$  per unit time. Then, we have

$$w' = w_2 - c_1(\tau\ell_2) - (c_1 - c_2) \cdot \frac{w_3}{c_2} = w_2 + w_3 - c_1\tau\ell_2 - \frac{c_1w_3}{c_2}.$$

Thus, the evacuation completion time is expressed as

$$\tau(\ell_1 + \ell_2) + \frac{w_3}{c_2} + \frac{w_2 + w_3 - c_1\tau\ell_2 - c_1w_3/c_2}{c_1} = \tau\ell_1 + \frac{w_2 + w_3}{c_1}. \quad (3)$$

**Case 2: It holds  $c_1 < c_2$ .** In this case, the amount of supply waiting at  $v_2$  increases by  $c_2 - c_1$  per unit time from time  $\tau\ell_2$  (when the foremost supply from  $v_3$  arrives at  $v_2$ ) to time  $\tau\ell_2 + w_3/c_2$  (when the supply from  $v_3$  completes to arrive at  $v_2$ ). Let us consider the following two situations at time  $\tau\ell_2$ . When the foremost supply from  $v_3$  arrives at  $v_2$ , there remains no supply at  $v_2$  or not.

If there remains no supply at  $v_2$  at time  $\tau\ell_2$ , then it holds  $w' = (c_2 - c_1)(w_3/c_2) = w_3 - c_1w_3/c_2$  in (1). Thus, the evacuation completion time is expressed as

$$\tau(\ell_1 + \ell_2) + \frac{w_3}{c_2} + \frac{w_3 - c_1w_3/c_2}{c_1} = \tau(\ell_1 + \ell_2) + \frac{w_3}{c_1}. \quad (4)$$

Otherwise, the situation is similar to the latter case of Case 1. The difference is that the amount of supply waiting at  $v_2$  increases by  $c_2 - c_1$  per unit time during from time  $\tau\ell_2$  to time  $\tau\ell_2 + w_3/c_2$ , while in Case 1, it decreases by  $c_1 - c_2$  per unit time. For this case, the evacuation completion time is given by formula (3).

In summary of formulae (2)–(4), the evacuation completion time for a dynamic flow path network with three vertices is given by the following formula:

$$\max \left\{ \tau\ell_1 + \frac{w_2 + w_3}{c_1}, \tau(\ell_1 + \ell_2) + \frac{w_3}{\min\{c_1, c_2\}} \right\}. \quad (5)$$

Let us turn to the case with  $n$  vertices, that is,  $V = \{v_i \mid 1 \leq i \leq n\}$ . When the sink is located at  $v_1$  and a parameter  $t \in T$  is fixed, generalizing formula (5), the evacuation completion time is

given by the following formula, which is provided by Higashikawa [17]:

$$\max_{2 \leq i \leq n} \left\{ \tau \sum_{j=1}^{i-1} \ell_j + \frac{\sum_{j=i}^n w_j(t)}{\min_{1 \leq j \leq i-1} c_j} \right\} = \max_{2 \leq i \leq n} \left\{ \tau v_i + \frac{W_{i,n}(t)}{C_{1,i}} \right\}. \quad (6)$$

An interesting observation is that each  $\tau v_i + W_{i,n}(t)/C_{1,i}$  in (6) is equivalent to the evacuation completion time for the transformed input so that only  $v_i$  is given supply  $W_{i,n}(t)$  and all the others are given zero supply.

Let us give explicit formula of the evacuation completion time for fixed  $x \in P$  and parameter  $t \in T$ . Suppose that a sink  $x$  is on edge  $e_i = (v_i, v_{i+1})$ . In this case, all the supply on the right side (i.e., at  $v_{i+1}, \dots, v_n$ ) will flow left to sink  $x$  and all the supply on the left side (i.e., at  $v_1, \dots, v_i$ ) will flow right to sink  $x$ . First, we consider the evacuation for the supply on the right side of  $x$ . Supply on the path is viewed as a continuous value, and we regard that all the supply on the right side of  $x$  is mapped to the interval  $(0, W_{i+1,n}(t)]$ . The value  $z$  satisfying  $z \in (W_{i+1,j-1}(t), W_{i+1,j}(t)]$  with  $i+1 \leq j \leq n$  represents all the supply at vertices  $v_{i+1}, v_{i+2}, \dots, v_{j-1}$  plus partial supply of  $z - W_{i+1,j-1}(t)$  at  $v_j$ . Let  $\theta_R^{e_i}(x, t, z)$  denote the time at which the first  $z$  amount of supply on the right side of  $x$  (i.e.,  $v_{i+1}, v_{i+2}, \dots, v_n$ ) completes its evacuation to sink  $x$ . Modifying formula (6),  $\theta_R^{e_i}(x, t, z)$  is given by the following formula: For  $z \in (W_{i+1,j-1}(t), W_{i+1,j}(t)]$  with  $i+1 \leq j \leq n$ ,

$$\theta_R^{e_i}(x, t, z) = \max_{i+1 \leq h \leq j} \left\{ \tau(v_h - x) + \frac{z - W_{i+1,h-1}(t)}{C_{i,h}} \right\}. \quad (7)$$

In a symmetric manner, we consider the evacuation for the supply on the left side of  $x$  (i.e.,  $v_1, \dots, v_i$ ). The value  $z$  satisfying  $z \in (W_{j+1,i}(t), W_{j,i}(t)]$  with  $1 \leq j \leq i$  represents all the supply at vertices  $v_i, v_{i-1}, \dots, v_{j+1}$  plus partial supply of  $z - W_{j+1,i}(t)$  at  $v_j$ . Let  $\theta_L^{e_i}(x, t, z)$  denote the time at which the first  $z$  amount of supply on the left side of  $x$  completes its evacuation to sink  $x$ , which is given by the following formula: For  $z \in (W_{j+1,i}(t), W_{j,i}(t)]$  with  $1 \leq j \leq i$ ,

$$\theta_L^{e_i}(x, t, z) = \max_{j \leq h \leq i} \left\{ \tau(x - v_h) + \frac{z - W_{h+1,i}(t)}{C_{h,i}} \right\}. \quad (8)$$

Let us turn to the case that sink  $x$  is at a vertex  $v_i \in V$ . We confirm that the evacuation times when the amount  $z$  of supply originating from the right side of and the left side of  $v_i$  to sink  $v_i$  are given by  $\theta_R^{e_i}(v_i, t, z)$  and  $\theta_L^{e_{i-1}}(v_i, t, z)$ , respectively.

## 2.2 Aggregate Evacuation Time

Let  $\Phi(x, t)$  be the aggregate evacuation time (i.e., sum of evacuation time) when a sink is at a point  $x \in P$  and the weight functions are fixed by a parameter  $t \in T$ . For a point  $x$  on edge  $e_i$  and a parameter  $t \in T$ , the aggregate evacuation time  $\Phi(x, t)$  is defined by the integrals of the evacuation completion times  $\theta_L^{e_i}(x, t, z)$  over  $z \in [0, W_{1,i}(t)]$  and  $\theta_R^{e_i}(x, t, z)$  over  $z \in [0, W_{i+1,n}(t)]$ , i.e.,

$$\Phi(x, t) = \int_0^{W_{1,i}(t)} \theta_L^{e_i}(x, t, z) dz + \int_0^{W_{i+1,n}(t)} \theta_R^{e_i}(x, t, z) dz. \quad (9)$$

In a similar way, if a sink  $x$  is at vertex  $v_i$ , then  $\Phi(v_i, t)$  is given by

$$\Phi(v_i, t) = \int_0^{W_{1,i-1}(t)} \theta_L^{e_{i-1}}(v_i, t, z) dz + \int_0^{W_{i+1,n}(t)} \theta_R^{e_i}(v_i, t, z) dz. \quad (10)$$

### 2.3 Minmax Regret Formulation

We denote by  $\text{Opt}(t)$  the minimum aggregate evacuation time with respect to a parameter  $t \in T$ . Higashikawa et al. [21] and Benkoczi et al. [7] showed that for the minsum  $k$ -sink location problems, there exists an optimal  $k$ -sink such that all the  $k$  sinks are at vertices. This implies that we have

$$\text{Opt}(t) = \min_{x \in V} \Phi(x, t) \quad (11)$$

for any  $t \in T$ . For a point  $x \in P$  and a value  $t \in T$ , a *regret*  $R(x, t)$  with regard to  $x$  and  $t$  is a gap between  $\Phi(x, t)$  and  $\text{Opt}(t)$  that is defined as

$$R(x, t) = \Phi(x, t) - \text{Opt}(t). \quad (12)$$

The *maximum regret* for a sink  $x \in P$ , denoted by  $MR(x)$ , is the maximum value of  $R(x, t)$  with respect to  $t \in T$ . Thus,  $MR(x)$  is defined as

$$MR(x) = \max_{t \in T} R(x, t). \quad (13)$$

Given a dynamic flow path network  $\mathcal{P}$  and a real interval  $T$ , the problem 1-MMR-AT-PW is defined as follows:

$$\text{minimize } MR(x) \text{ subject to } x \in P \quad (14)$$

Let  $x^*$  denote an optimal solution of (14).

### 2.4 Piecewise Functions and Upper/Lower Envelopes

A function  $f : X(\subset \mathbb{R}) \rightarrow \mathbb{R}$  is called a *piecewise polynomial function* if and only if real interval  $X$  can be partitioned into subintervals  $X_1, X_2, \dots, X_m$  so that  $f$  forms as a polynomial  $f_i$  on each  $X_i$ . We denote such a piecewise polynomial function  $f$  by  $f = \langle (f_1, X_1), \dots, (f_m, X_m) \rangle$ , or simply  $f = \langle (f_i, X_i) \rangle$ . We assume that such partition into subintervals are maximal in a sense that for any  $i$  and  $i + 1$   $f_i \neq f_{i+1}$ . We call each pair  $(f_i, X_i)$  a *piece* of  $f$ , and an endpoint of the closure of  $X_i$  a *breakpoint* of  $f$ . A piecewise polynomial function  $f = \langle (f_i, X_i) \rangle$  is called a *piecewise polynomial function of degree at most two* if and only if each  $f_i$  is quadratic or linear. We confirm the following property about the sum of piecewise polynomial functions.

**Proposition 1.** *Let  $m$  and  $m'$  be positive integers, and  $f, g : X(\subset \mathbb{R}) \rightarrow \mathbb{R}$  be piecewise polynomial functions of degree at most two with  $m$  and  $m'$  pieces, respectively. Then, a function  $h = f + g$  is a piecewise polynomial function of degree at most two with at most  $m + m'$  pieces. Moreover, given  $f = \langle (f_i, X_i) \rangle$  and  $g = \langle (g_j, X'_j) \rangle$ , we can obtain  $h = f + g = \langle (h_j, X''_j) \rangle$  in  $O(m + m')$  time.*

Let  $\mathcal{F} = \{f_1(y), \dots, f_m(y)\}$  be a family of  $m$  polynomial functions where  $f_i : Y_i(\subset \mathbb{R}) \rightarrow \mathbb{R}$  and  $Y$  denote the union of  $Y_i$ , that is,  $Y = \cup_{i=1}^m Y_i$ . An *upper envelope*  $\mathcal{U}_{\mathcal{F}}(y)$  and a *lower envelope*  $\mathcal{L}_{\mathcal{F}}(y)$  of  $\mathcal{F}$  are functions from  $Y$  to  $\mathbb{R}$  defined as follows:

$$\mathcal{U}_{\mathcal{F}}(y) = \max_{i=1, \dots, m} f_i(y), \quad \mathcal{L}_{\mathcal{F}}(y) = \min_{i=1, \dots, m} f_i(y), \quad (15)$$

where the maximum and the minimum are taken over those functions that are defined at  $y$ , respectively. For an upper envelope  $\mathcal{U}_{\mathcal{F}}(y)$  of  $\mathcal{F}$ , there exist an integer sequence  $U_{\mathcal{F}} = \langle u_1, \dots, u_k \rangle$  and subintervals  $I_1, \dots, I_k$  of  $Y$  such that  $\mathcal{U}_{\mathcal{F}}(y) = \langle (f_{u_1}(y), I_1), \dots, (f_{u_k}(y), I_k) \rangle$  holds. That is, an upper envelope  $\mathcal{U}_{\mathcal{F}}(y)$  can be represented as a piecewise polynomial function. We call the above sequence  $U_{\mathcal{F}}$  the *upper-envelope sequence* of  $\mathcal{U}_{\mathcal{F}}(y)$ .

In our algorithm, we compute the upper/lower envelopes of partially defined, univariate polynomial functions. The following result is useful for this operation.

**Theorem 2** ([15, 16, 1]). *Let  $\mathcal{F}$  be a family of  $n$  partially defined, polynomial functions of degree at most two. Then,  $\mathcal{U}_{\mathcal{F}}$  and  $\mathcal{L}_{\mathcal{F}}$  consist of  $O(n2^{\alpha(n)})$  pieces and one can obtain them in time  $O(n\alpha(n)\log n)$ , where  $\alpha(n)$  is the inverse Ackermann function. Moreover, if  $\mathcal{F}$  a family of  $n$  line segments, then  $\mathcal{U}_{\mathcal{F}}$  and  $\mathcal{L}_{\mathcal{F}}$  consist of  $O(n)$  pieces and one can obtain them in time  $O(n\log n)$ .*

Note that the number of pieces and the computation time for the upper/lower envelopes are involved with the maximum length of Davenport–Schinzel sequences. See [15] for the details. For a family  $\mathcal{F}$  of functions, if we say that we *obtain* envelopes  $\mathcal{U}_{\mathcal{F}}(y)$  or  $\mathcal{L}_{\mathcal{F}}(y)$ , then we obtain the information of all pieces  $(f_{u_i}(y), I_i)$ .

## 2.5 Property of the Inverse Ackermann Function

The Ackerman function is defined as follows:

$$A(n, m) = \begin{cases} m + 1 & \text{if } n = 0, \\ A(n - 1, 1) & \text{if } n > 0, m = 0, \\ A(n - 1, A(n, m - 1)) & \text{otherwise.} \end{cases}$$

The inverse Ackermann function  $\alpha(n)$  is defined as

$$\alpha(n) = \min\{k \in \mathbb{N}_0 \mid n \leq A(k, k)\}.$$

We show that the following inequality:

**Property 1.**

$$\alpha(n^3) \leq \alpha(n) + 1 \tag{16}$$

holds for any positive integer  $n$  with  $n \geq 8$ .

Let us suppose that  $n \geq 8$  and  $k = \alpha(n) \geq 3$  since  $A(2, 2) = 7$  holds. The inequality (16) is equivalent to the inequality

$$n^3 \leq A(k + 1, k + 1).$$

By the definition and the monotonicity of the Ackermann function, we have

$$A(k + 1, k + 1) = A(k, A(k + 1, k)) > A(k, A(k, k)) \geq 2^{A(k, k)} \geq (A(k, k))^3.$$

The last two inequalities are led by the fact that

$$A(k, m) \geq A(3, m) = 2^{m+3} - 3 \geq m^3$$

holds for any  $k \geq 3$  and any positive integer  $m$ . Since  $n \leq A(k, k)$  holds, we have  $n^3 \leq (A(k, k))^3 \leq A(k + 1, k + 1)$ . Thus, the proof completes.

## 3 Algorithms

The main task of the algorithm is to compute the following  $O(n)$  values,  $MR(v)$  for all  $v \in V$  and  $\min\{MR(x) \mid x \in e\}$  for all  $e \in E$ . Once we compute these values, we immediately obtain the solution of the problem by choosing the minimum one among them in  $O(n)$  time.

Let us focus on computing  $\min\{MR(x) \mid x \in e\}$  for each  $e \in E$ . (Note that we can compute  $MR(v)$  for  $v \in V$  in a similar manner.) Recall the definition of the maximum regret for  $x$ ,  $MR(x) = \max\{R(x, t) \mid t \in T\}$ . A main difficulty lies in evaluating  $R(x, t)$  over  $t \in T$  even for a fixed  $x$  since we need treat an infinite set  $T$ . Furthermore, we are also required to find an optimal location among an infinite set  $e$ . To tackle with this issue, our key idea is to partition the problem

into a polynomial number of subproblems as follows: We partition interval  $T$  into a polynomial number of subintervals  $T_1, \dots, T_m$  so that  $R(x, t)$  is represented as a (single) polynomial function in  $x$  and  $t$  on  $\{x \in e\} \times T_j$  for each  $j = 1, \dots, m$ . For each  $T_j$ , we compute the maximum regret for  $x \in e$  over  $T_j$  denoted by  $G_j(x) = \max\{R(x, t) \mid t \in T_j\}$ . An explicit form of  $G_j(x)$  will be given in Sec. 3.1. We then obtain  $MR(x)$  for  $x \in e$  as the upper envelope of functions  $G_1(x), \dots, G_m(x)$  and find the minimum value of  $MR(x)$  for  $x \in e$  by elementary calculation.

In the rest of the paper, we mainly show that for each  $e$  or  $v$ , there exists a partition of  $T$  with a polynomial number of subintervals such that the regret  $R(x, t)$  is a polynomial function of degree at most two on each subinterval.

### 3.1 Key Lemmas

To understand  $R(x, t)$ , we observe function  $\Phi(x, t)$ . We give some other notations. Let  $f_{\mathbf{R}}^{e_i, j}(t, z)$  and  $f_{\mathbf{L}}^{e_i, j}(t, z)$  denote functions obtained by removing terms containing  $x$  from formulae (7) and (8). Formally, for  $1 \leq i < j \leq n$ , let function  $f_{\mathbf{R}}^{e_i, j}(t, z)$  be defined on  $t \in T$  and  $z \in (W_{i+1, j-1}(t), W_{i+1, n}(t))$  as

$$f_{\mathbf{R}}^{e_i, j}(t, z) = \tau v_j + \frac{z - W_{i+1, j-1}(t)}{C_{i, j}}, \quad (17)$$

and for  $1 \leq j < i \leq n$ , let function  $f_{\mathbf{L}}^{e_i, j}(t, z)$  be defined on  $t \in T$  and  $z \in (W_{j+1, i}(t), W_{1, i}(t))$  as

$$f_{\mathbf{L}}^{e_i, j}(t, z) = -\tau v_j + \frac{z - W_{j+1, i}(t)}{C_{j, i}}. \quad (18)$$

In addition, let  $F_{\mathbf{L}}^{e_i}(t)$  and  $F_{\mathbf{R}}^{e_i}(t)$  denote univariate functions defined as

$$F_{\mathbf{L}}^{e_i}(t) = \int_0^{W_{1, i}(t)} f_{\mathbf{L}}^{e_i}(t, z) dz, \quad F_{\mathbf{R}}^{e_i}(t) = \int_0^{W_{i+1, n}(t)} f_{\mathbf{R}}^{e_i}(t, z) dz, \quad (19)$$

where  $f_{\mathbf{L}}^{e_i}(t, z)$  and  $f_{\mathbf{R}}^{e_i}(t, z)$  denote functions defined as

$$f_{\mathbf{L}}^{e_i}(t, z) = \max_{1 \leq j \leq i} \left\{ f_{\mathbf{L}}^{e_i, j}(t, z) \right\}, \quad f_{\mathbf{R}}^{e_i}(t, z) = \max_{i+1 \leq j \leq n} \left\{ f_{\mathbf{R}}^{e_i, j}(t, z) \right\}.$$

Recall the definition of the aggregate evacuation time  $\Phi(x, t)$  shown in (9). We observe that for  $x \in e_i$ ,  $\Phi(x, t)$  can be represented as

$$\begin{aligned} \Phi(x, t) &= (W_{1, i}(t) - W_{i+1, n}(t))\tau x + \int_0^{W_{1, i}(t)} f_{\mathbf{L}}^{e_i}(t, z) dz + \int_0^{W_{i+1, n}(t)} f_{\mathbf{R}}^{e_i}(t, z) dz \\ &= (W_{1, i}(t) - W_{i+1, n}(t))\tau x + F_{\mathbf{L}}^{e_i}(t) + F_{\mathbf{R}}^{e_i}(t). \end{aligned} \quad (20)$$

In a similar manner, by the definition of (10) and formula (19), we have

$$\Phi(v_i, t) = (W_{1, i-1}(t) - W_{i+1, n}(t))\tau v_i + F_{\mathbf{L}}^{e_i-1}(t) + F_{\mathbf{R}}^{e_i}(t). \quad (21)$$

Let us focus on function  $F_{\mathbf{R}}^e(t)$ . As  $t$  increases, while the upper-envelope sequence of  $f_{\mathbf{R}}^e(t, z)$  w.r.t.  $z$  remains the same, function  $F_{\mathbf{R}}^e(t)$  is represented as the same polynomial, whose degree is at most two by formulae (17), (18) and (19). In other words, a breakpoint of  $F_{\mathbf{R}}^e(t)$  corresponds to the value  $t$  such that the upper-envelope sequence of  $f_{\mathbf{R}}^e(t, z)$  w.r.t.  $z$  changes. We notice that such a change happens only when three functions  $f_{\mathbf{R}}^{e, h}(t, z)$ ,  $f_{\mathbf{R}}^{e, i}(t, z)$  and  $f_{\mathbf{R}}^{e, j}(t, z)$  intersect each other, which can happen at most once. This implies that  $F_{\mathbf{R}}^e(t)$  consists of  $O(n^3)$  breakpoints, that is, it is a piecewise polynomial function of degree at most two with  $O(n^3)$  pieces. The following lemma shows that the number of pieces is actually  $O(n^2)$ .

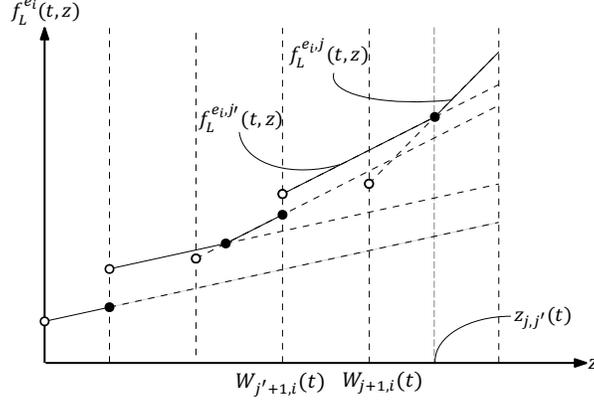


Figure 1: Graph of  $f_L^{e_i}(t, z)$  w.r.t.  $z$  with a focus on  $f_L^{e_i, j}(t, z)$ ,  $f_L^{e_i, j'}(t, z)$  and  $z_{j, j'}(t)$  with  $j < j'$ .

**Lemma 1.** For each  $e \in E$ ,  $F_L^e(t)$  and  $F_R^e(t)$  are piecewise polynomial functions of degree at most two with  $O(n^2)$  pieces, and can be computed in  $O(n^3 \log n)$  time. Especially, when all the edge capacities are uniform, the numbers of pieces of them are  $O(n)$ , and can be computed in  $O(n^2 \log n)$  time.

*Proof.* Let us suppose that  $x$  is on  $e_i \in E$ . We will show that  $F_L^{e_i}(t)$  is a piecewise polynomial functions of degree at most two with  $O(n^2)$  pieces, and when all the edge capacities are uniform, the numbers of pieces of them are  $O(n)$ . For  $F_R^{e_i}(t)$ , we can see the same properties in a symmetric manner.

To show the above statement, we give the following properties of  $F_L^{e_i}(t)$ :

- (i) The number of pieces of  $F_L^{e_i}(t)$  is  $O(n^2)$ . Especially, when all the edge capacities are uniform, it is  $O(n)$ .
- (ii) Function of each piece of  $F_L^{e_i}(t)$  is a quadratic function in  $t$ .

First, we show that property (i) holds. Recall that  $f_L^{e_i}(t, z)$  is the upper envelope of  $\{f_L^{e_i, j}(t, z) \mid 1 \leq j \leq i\}$  w.r.t.  $z \in (0, W_{1, i}(t)]$ . For integers  $j, j'$  with  $1 \leq j < j' \leq i$ , let  $z_{j, j'}(t)$  be a function in  $t$  defined as follows: If  $C_{j, i} = C_{j', i}$ ,

$$z_{j, j'}(t) = \begin{cases} W_{j+1, i}(t) & \text{if } f_L^{e_i, j'}(t, W_{j+1, i}(t)) \leq f_L^{e_i, j}(t, W_{j+1, i}(t)), \\ W_{1, i}(t) & \text{if } f_L^{e_i, j'}(t, W_{j+1, i}(t)) > f_L^{e_i, j}(t, W_{j+1, i}(t)), \end{cases} \quad (22)$$

otherwise, i.e., if  $C_{j, i} < C_{j', i}$ ,

$$z_{j, j'}(t) = \begin{cases} W_{j+1, i}(t) & \text{if } f_L^{e_i, j'}(t, W_{j+1, i}(t)) \leq f_L^{e_i, j}(t, W_{j+1, i}(t)), \\ \min\{z_{j, j'}^*(t), W_{1, i}(t)\} & \text{if } f_L^{e_i, j'}(t, W_{j+1, i}(t)) > f_L^{e_i, j}(t, W_{j+1, i}(t)), \end{cases} \quad (23)$$

where  $z_{j, j'}^*(t)$  is the solution for  $z$  of equation  $f_L^{e_i, j}(t, z) = f_L^{e_i, j'}(t, z)$  (See Figure 1). Note that inequality conditions in (22) and (23) can be solved for  $t$  as  $[t \leq t_{j, j'}; t > t_{j, j'}]$  or  $[t \geq t_{j, j'}; t < t_{j, j'}]$ , where  $t_{j, j'}$  is the solution for  $t$  of equation  $f_L^{e_i, j'}(t, W_{j+1, i}(t)) = f_L^{e_i, j}(t, W_{j+1, i}(t))$ .

For any  $t \in T$ , let  $U(t)$  be the upper-envelope sequence of  $f_L^{e_i}(t, z)$  in  $z$ . Now, for some  $t'$ , suppose  $U(t') = \langle u_1, \dots, u_k \rangle$ . Note that  $u_1 > \dots > u_k$  holds. Then, the  $h$ -th smallest breakpoint of  $f_L^{e_i}(t', z)$  for  $z \in (0, W_i(t'))$  is  $z_{u_{h+1}, u_h}(t')$ . We notice that if  $t$  changes from  $t'$  in condition that  $U(t)$  and formula of each  $z_{u_{h+1}, u_h}(t)$  remain the same, formula of  $F_L^{e_i}(t)$  also remains the same. In other words, a breakpoint of  $F_L^{e_i}(t)$  corresponds to when  $U(t)$  or formula of some  $z_{u_{h+1}, u_h}(t)$  changes. Consider the following three cases.

[Case 1]: When  $t$  comes to some value  $t_1$ , formula of some  $z_{u_{h+1},u_h}(t)$  changes.

[Case 2]: Just after  $t$  comes to some value  $t_2$ , i.e., when  $t = t_2 + \epsilon$ ,  $f_L^{e_i,j}(t_2 + \epsilon, z)$  of some  $j$  appears as a part of  $f_L^{e_i}(t_2 + \epsilon, z)$  so that  $U(t_2 + \epsilon) = \langle u_1, \dots, u_h, j, u_{h+1}, \dots, u_k \rangle$ .

[Case 3]: When  $t$  comes to some value  $t_3$ ,  $f_L^{e_i, u_h}(t_3, z)$  disappears from  $f_L^{e_i}(t_3, z)$ , i.e.,  $U(t_3) = \langle u_1, \dots, u_{h-1}, u_{h+1}, \dots, u_k \rangle$ .

We observe that each of the above  $t_1, t_2$  and  $t_3$  is a breakpoint of

$$z_{u_{h+1}}(t) := \max_{j'} \{z_{u_{h+1},j'}(t) \mid u_{h+1} + 1 \leq j' \leq i\}. \quad (24)$$

Therefore, the number of breakpoints of  $F_L^{e_i}(t)$  is at most the number of all the breakpoints of

$$z_j(t) := \max_{j'} \{z_{j,j'}(t) \mid j + 1 \leq j' \leq i\} \quad \forall j \text{ with } 1 \leq j \leq i - 1, \quad (25)$$

which is  $O(n^2)$  since the number of breakpoints of  $z_j(t)$  for any  $j$  is  $O(n)$ . Especially, when all the edge capacities are uniform, according to (22) and (25), we have

$$z_j(t) = \begin{cases} W_{j+1,i}(t) & \text{if } f_L^{e_i,j'}(t, W_{j+1,i}(t)) \leq f_L^{e_i,j}(t, W_{j+1,i}(t)) \\ & \forall j' \text{ with } j + 1 \leq j' \leq i, \\ W_{1,i}(t) & \text{otherwise.} \end{cases} \quad (26)$$

Thus, the number of breakpoints of  $z_j(t)$  for any  $j$  is at most two, which means that the number of breakpoints of  $F_L^{e_i}(t)$  is  $O(n)$ . This completes the proof for property (i).

We next show that property (ii) holds. Let  $(H(t), T')$  be a piece of  $F_L^{e_i}(t)$  such that for  $t \in T'$ ,  $U(t) = \langle u_1, \dots, u_k \rangle$  and formula of each  $z_{u_{h+1},u_h}(t)$  remains the same. We then have

$$H(t) = \sum_{h=1}^k \int_{z_{u_h, u_{h-1}}(t)}^{z_{u_{h+1}, u_h}(t)} f_L^{e_i, u_h}(t, z) dz, \quad (27)$$

where  $z_{u_1, u_0}(t) = 0$  and  $z_{u_{k+1}, u_k}(t) = W_{1,i}(t)$ . By formula (18), it holds for  $h$  with  $1 \leq h \leq k$  that

$$f_L^{e_i, u_h}(t, z) = \rho_1^h z + \rho_2^h t + \rho_3^h, \quad (28)$$

where  $\rho_1^h, \rho_2^h, \rho_3^h$  are some constants. Substituting (28) into (27), we obtain

$$H(t) = \sum_{h=1}^k \left\{ \frac{\rho_1^h}{2} (z_{u_{h+1}, u_h}(t)^2 - z_{u_h, u_{h-1}}(t)^2) + (\rho_2^h t + \rho_3^h) (z_{u_{h+1}, u_h}(t) - z_{u_h, u_{h-1}}(t)) \right\}, \quad (29)$$

which is quadratic in  $t$  since  $z_{u_{h+1}, u_h}(t)$  and  $z_{u_h, u_{h-1}}(t)$  are linear in  $t$ . This completes the proof for property (ii).

The rest of the proof is to give an algorithm for obtaining  $F_L^{e_i}(t) = \langle \langle F_{L,h}^{e_i}(t), T_{L,h} \rangle \rangle$ . Let  $m$  denote the number of pieces of  $F_L^{e_i}(t)$ . Note that one can obtain  $F_R^{e_i}(t)$  in a similar manner. Our algorithm consists of the following two steps:

**Step. 1: Obtain breakpoints of  $F_L^{e_i}(t)$ .** For each  $j$  with  $1 \leq j \leq i$ , we find all the breakpoints of an upper envelope  $z_j(t)$ . Since  $z_j(t)$  consists of partially defined linear functions, we can apply Theorem 2. Therefore, this operation requires  $O(n \log n)$  time for each  $j$ . Thus, Step 1 requires  $O(n^2 \log n)$  time.

**Step. 2: Obtain polynomial functions of all the pieces of  $F_L^{e_i}(t)$ .** Let  $T'$  be an interval of a piece of  $F_L^{e_i}(t)$ . Picking up a parameter  $t \in T'$ , we obtain an upper-envelope sequence  $U(t) = \langle u_1, \dots, u_k \rangle$  in  $O(n \log n)$  time by Theorem 2. We obtain polynomial of degree at most two in  $O(n)$  time by evaluating formula (29). Since the number of pieces of  $F_L^{e_i}(t)$  is at most  $m$ , Step 2 requires  $O(nm \log n)$  time.

In total, our algorithm requires  $O(n(n+m) \log n)$  time, which completes the proof of our main theorem because  $m = O(n^2)$ , and for the case with uniform edge capacity,  $m = O(n)$ .  $\square$

Let  $N_F$  denote the maximum number of pieces of  $F_L^e(t)$  and  $F_R^e(t)$  over  $e \in E$ . Then we have  $N_F = O(n^2)$ , and for the case with uniform edge capacity,  $N_F = O(n)$ . Next, we consider  $\text{Opt}(t) = \min\{\Phi(x, t) \mid x \in V\}$ , which is the lower envelope of a family of  $n$  functions  $\Phi(v_i, t)$  in  $t$ . Theorem 2 and Lemma 1 imply the following lemma.

**Lemma 2.** *Opt(t) is a piecewise polynomial function of degree at most two with  $O(nN_F 2^{\alpha(n)})$  pieces, and can be obtained in  $O(nN_F \alpha(n) \log n)$  time if functions  $F_L^e(t)$  and  $F_R^e(t)$  for all  $e \in E$  are available.*

*Proof.* Formula (11) implies that  $\text{Opt}(t)$  is the lower envelope of a family of  $n$  functions  $\{\Phi(v, t) \mid v \in V\}$ . Recall that for  $v_i \in V$ , we have

$$\Phi(v_i, t) = (W_{1,i-1}(t) - W_{i+1,n}(t))\tau v_i + F_L^{e_i-1}(t) + F_R^{e_i}(t).$$

By Lemma 1, since  $F_L^e(t)$  and  $F_R^e(t)$  for any  $e \in E$  is a piecewise polynomial function of degree at most two with at most  $N_F$  pieces,  $\text{Opt}(t)$  is the lower envelope of  $O(nN_F)$  partially defined polynomial functions of degree at most two.

Theorem 2 implies that  $\text{Opt}(t)$  is a piecewise polynomial function of degree at most two with at most  $O(nN_F 2^{\alpha(nN_F)}) = O(nN_F 2^{\alpha(n)})$  pieces and can be obtained in  $O(nN_F \alpha(nN_F) \log(nN_F)) = O(nN_F \alpha(n) \log n)$  time, where we used the facts that  $N_F = O(n^3)$  and Property 1. It completes the proof.  $\square$

Let  $N_{\text{Opt}}$  denote the number of pieces of  $\text{Opt}(t)$ . Then we have  $N_{\text{Opt}} = O(nN_F 2^{\alpha(n)})$ .

Let us consider  $R(x, t)$  in the case that sink  $x$  is on an edge  $e_i \in E$ . Substituting formula (20) for (12), we have

$$R(x, t) = \Phi(x, t) - \text{Opt}(t) = (W_{1,i}(t) - W_{i+1,n}(t))\tau x + F_L^{e_i}(t) + F_R^{e_i}(t) - \text{Opt}(t).$$

By Proposition 1,  $F_L^{e_i}(t) + F_R^{e_i}(t) - \text{Opt}(t)$  is a piecewise polynomial function of degree at most two with at most  $2N_F + N_{\text{Opt}} = O(N_{\text{Opt}})$  pieces. Let  $N_{e_i}$  be the number of pieces of  $F_L^{e_i}(t) + F_R^{e_i}(t) - \text{Opt}(t)$  and  $T_j^{e_i}$  be the interval of the  $j$ -th piece (from the left) of  $F_L^{e_i}(t) + F_R^{e_i}(t) - \text{Opt}(t)$ . Thus,  $R(x, t)$  is represented as a (single) polynomial function in  $x$  and  $t$  on  $\{x \in e\} \times T_j$  for each  $T_j$ . For each integer  $j$  with  $1 \leq j \leq N_{e_i}$ , let  $G_j^{e_i}(x)$  be a function defined as

$$G_j^{e_i}(x) = \max\{R(x, t) \mid t \in T_j^{e_i}\}. \quad (30)$$

We then have the following lemma.

**Lemma 3.** *For each  $e_i \in E$  and  $j$  with  $1 \leq j \leq N_{e_i}$ ,  $G_j^{e_i}(x)$  is a piecewise polynomial function of degree at most two with at most three pieces, and can be obtained in constant time if functions  $F_L^{e_i}(t)$ ,  $F_R^{e_i}(t)$  and  $\text{Opt}(t)$  are available.*

*Proof.* Recall that we have

$$R(x, t) = \Phi(x, t) - \text{Opt}(t) = (W_{1,i}(t) - W_{i+1,n}(t))\tau x + F_L^{e_i}(t) + F_R^{e_i}(t) - \text{Opt}(t).$$

Because  $W_{1,i}(t)$ ,  $W_{i+1,n}(t)$  are linear in  $t$  and  $F_L^{e_i}(t) + F_R^{e_i}(t) - \text{Opt}(t)$  are polynomial function of degree at most two on  $t \in T_j$ , we can represent  $R(x, t)$  on  $\{x \in e_i\} \times T_j$  with five real constants  $\beta_k$  ( $k = 1, \dots, 5$ ) as

$$R(x, t) = \beta_1 t^2 + \beta_2 x t + \beta_3 t + \beta_4 x + \beta_5.$$

Let us observe the explicit form of the maximum value  $G_j(x)$  of  $R(x, t)$  for  $x \in e_i$  over  $t \in T_j$ . Let  $T_j = [t_{j-1}, t_j]$ . If  $\beta_1 \geq 0$ ,  $R(x, t)$  takes the maximum value when  $t = t_{j-1}$  or  $t = t_j$  for any  $x$ . Thus, we have  $G_j(x) = \max\{R(x, t_{j-1}), R(x, t_j)\}$  that is a piecewise linear function in  $x$  with at

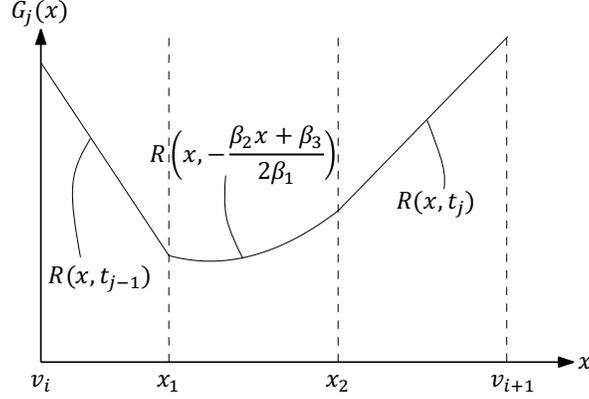


Figure 2: Graph of  $G_j(x)$  in the case of  $v_i < x_1 < x_2 < v_{i+1}$ .

most two pieces, since both  $R(x, t_{j-1})$  and  $R(x, t_j)$  are linear in  $x$ . Let us consider the other case that  $\beta_1 < 0$  holds. When the axis of symmetry of  $R(x, t)$  w.r.t.  $t$ , i.e.,  $t = -(\beta_2 x + \beta_3)/(2\beta_1)$ , is contained in  $T_j$ , it holds  $G_j(x) = R(x, -(\beta_2 x + \beta_3)/(2\beta_1))$ . We thus have

$$G_j(x) = \begin{cases} R(x, t_{j-1}) & \text{if } -\frac{\beta_2 x + \beta_3}{2\beta_1} < t_{j-1}, \\ R\left(x, -\frac{\beta_2 x + \beta_3}{2\beta_1}\right) & \text{if } t_{j-1} \leq -\frac{\beta_2 x + \beta_3}{2\beta_1} \leq t_j, \\ R(x, t_j) & \text{if } -\frac{\beta_2 x + \beta_3}{2\beta_1} > t_j. \end{cases} \quad (31)$$

Note that inequality conditions in (31) can be solved for  $x$  as  $[x < x_1; x_1 \leq x \leq x_2; x > x_2]$  or  $[x > x_1; x_1 \geq x \geq x_2; x < x_2]$ , where  $x_1$  is the solution for  $x$  of equation  $-(\beta_2 x + \beta_3)/(2\beta_1) = t_{j-1}$  and  $x_2$  is the solution for  $x$  of equation  $-(\beta_2 x + \beta_3)/(2\beta_1) = t_j$  (See Figure 2). Therefore,  $G_j(x)$  is a piecewise polynomial function with at most three polynomial of degree at most two because  $R\left(x, -\frac{\beta_2 x + \beta_3}{2\beta_1}\right)$  is a quadratic function in  $x$ .  $\square$

Recalling the definition of  $MR(x)$ , it holds that for  $x \in e$ ,

$$MR(x) = \max\{R(x, t) \mid t \in T\} = \max\{G_j^e(x) \mid 1 \leq j \leq N_e\},$$

that is,  $MR(x)$  is the upper envelope of functions  $G_1^e(x), \dots, G_{N_e}^e(x)$ . Applying Theorem 2, we have the following lemma.

**Lemma 4.** *For each  $e \in E$ , there exists an algorithm that finds a location that minimizes  $MR(x)$  under the restriction with  $x \in e$  in  $O(N_{\text{Opt}}\alpha(n) \log n)$  time if functions  $F_L^e(t)$ ,  $F_R^e(t)$  and  $\text{Opt}(t)$  are available.*

*Proof.* We give how to find a location  $x^{*,e}$  that minimizes  $MR(x)$  over  $x \in e$ . Since functions  $F_L^e(t)$ ,  $F_R^e(t)$  and  $\text{Opt}(t)$  are available, we can apply Lemma 3 and then compute the explicit forms of functions  $G_j^e(x)$  for all  $j$  with  $1 \leq j \leq N_e$  which are obtained in  $O(N_e) = O(N_{\text{Opt}})$  time. Function  $MR(x)$  for  $x \in e$  is the upper envelope of functions  $G_1^e(x), \dots, G_{N_e}^e(x)$ . Since  $G_j^e(x)$  is a piecewise polynomial function of degree at most two with at most three pieces,  $MR(x)$  is the upper envelope of at most  $3N_e = O(N_{\text{Opt}})$  functions of degree at most two. Theorem 2 implies that  $MR(x)$  consists of  $O(N_{\text{Opt}}2^{\alpha(N_{\text{Opt}})}) = O(N_{\text{Opt}}2^{\alpha(n)})$  pieces and can be obtained in  $O(N_{\text{Opt}}\alpha(N_{\text{Opt}}) \log N_{\text{Opt}}) = O(N_{\text{Opt}}\alpha(n) \log n)$  time, where we used the fact that the inverse Ackermann function satisfies that  $\alpha(n^3) \leq \alpha(n) + 1$  holds for any positive integer  $n$  with  $n \geq 8$ .

For each piece, compute a point which minimizes  $MR(x)$  in constant time, and among the obtained values, choose the minimum one as  $x^{*,e}$ . Summarizing the above argument, these operations take  $O(N_{\text{Opt}}\alpha(n) \log n)$  time.  $\square$

Note that the small modification for the algorithm of Lemma 4 leads that we can also compute  $MR(v)$  for all  $v \in V$  in  $O(N_{\text{Opt}})$  time.

**Lemma 5.** *For each  $v \in V$ , there exists an algorithm that computes  $MR(v)$  in  $O(N_{\text{Opt}})$  time if functions  $F_L^e(t)$ ,  $F_R^e(t)$  and  $\text{Opt}(t)$  are available for all  $e \in E$ .*

*Proof.* We show that it takes to obtain  $MR(v_i)$  in  $O(N_{\text{Opt}})$  time for each  $v_i \in V$ . Substituting formula (21) for (12), we have

$$R(v_i, t) = (W_{i-1}(t) + W_i(t) - W_n(t))\tau v_i + F_L^{e_{i-1}}(t) + F_R^{e_i}(t) - \text{Opt}(t).$$

By Proposition 1,  $F_L^{e_{i-1}}(t) + F_R^{e_i}(t) - \text{Opt}(t)$  is a piecewise polynomial function of degree at most two with at most  $O(N_{\text{Opt}})$  pieces. Let  $N_{v_i}$  be the number of pieces of  $F_L^{e_{i-1}}(t) + F_R^{e_i}(t) - \text{Opt}(t)$  and  $T_j^{v_i}$  be the interval of the  $j$ -th piece (from the left) of  $F_L^{e_{i-1}}(t) + F_R^{e_i}(t) - \text{Opt}(t)$ . Thus,  $R(v_i, t)$  is represented as a polynomial function of degree at most two on  $t \in T_j^{v_i}$  for each  $T_j^{v_i}$ . For each integer  $j$  with  $1 \leq j \leq N_{v_i}$ , by elementary calculation, we obtain the maximum value  $G_j^{v_i}$  of  $R(v_i, t)$  over  $t \in T_j$  in  $O(1)$  time. By choosing the maximum values among  $G_1^{v_i}, \dots, G_{N_{v_i}}^{v_i}$ , we can obtain  $MR(v_i)$  in  $O(N_{v_i}) = O(N_{\text{Opt}})$  time.  $\square$

### 3.2 Algorithms and Time Analyses

Let us give an algorithm that finds a sink location that minimizes the maximal regret and the analysis of the running time of each step.

First, we obtain  $F_L^e(t)$  and  $F_R^e(t)$  for all  $e \in E$  and obtain function  $\text{Opt}(t)$  as a preprocess. Applying Lemmas 1 and 2, we take  $O(n^2 N_F \log n)$  time for these operations. Next, we compute  $x^{*,e} = \arg \min\{MR(x) \mid x \in e\}$  for all  $e \in E$  in  $O(n N_{\text{Opt}} \alpha(n) \log n)$  time by applying Lemma 4. Then, we also compute  $MR(v)$  for all  $v \in V$  in  $O(n N_{\text{Opt}})$  time by applying Lemma 5. Finally, we find an optimal sink location  $x^*$  in  $O(n)$  time by evaluating the values  $MR(x)$  for  $x \in \{x^{*,e}\} \cup V$ .

Since we have  $N_{\text{Opt}} = O(n N_F 2^{\alpha(n)})$ , the bottleneck of our algorithm is to compute  $x^{*,e}$  for all  $e \in E$ . Thus, we see that the algorithm runs in  $O(n^2 N_F 2^{\alpha(n)} \alpha(n) \log n)$  time, which completes the proof of our main theorem because  $N_F = O(n^2)$ , and for the case with uniform edge capacity,  $N_F = O(n)$ .

## References

- [1] Agarwal, P.K., Sharir, M., Shor, P.: Sharp upper and lower bounds on the length of general davenport-schinzel sequences. *J. Comb. Theory, Ser. A* **52**(2), 228–274 (1989)
- [2] Alstrup, S., Gavoille, C., Kaplan, H., Rauhe, T.: Nearest common ancestors: a survey and a new distributed algorithm. In: *Proc. of the 14th annual ACM symposium on Parallel algorithms and architectures (SPAA 2002)*. pp. 258–264 (2002)
- [3] Arumugam, G.P., Augustine, J., Golin, M.J., Srikanthan, P.: Minmax regret  $k$ -sink location on a dynamic path network with uniform capacities. *Algorithmica* **81**(9), 3534–3585 (2019)
- [4] Belmonte, R., Higashikawa, Y., Katoh, N., Okamoto, Y.: Polynomial-time approximability of the  $k$ -sink location problem. *CoRR* **abs/1503.02835** (2015)
- [5] Bender, M.A., Farach-Colton, M.: The LCA problem revisited. In: *Proc. of the 4th Latin American Symposium on Theoretical Informatics (LATIN 2000)*. pp. 88–94 (2000)
- [6] Benkoczi, R., Bhattacharya, B., Higashikawa, Y., Kameda, T., Katoh, N.: Minsum  $k$ -sink problem on dynamic flow path networks. In: *Proc. of the 29th International Workshop on Combinatorial Algorithms (IWOCA 2018)*. pp. 78–89 (2018)

- [7] Benkoczi, R., Bhattacharya, B., Higashikawa, Y., Kameda, T., Katoh, N.: Minsum  $k$ -sink problem on path networks. *Theor. Comput. Sci.* **806**, 388–401 (2020)
- [8] Bhattacharya, B., Golin, M.J., Higashikawa, Y., Kameda, T., Katoh, N.: Improved algorithms for computing  $k$ -sink on dynamic flow path networks. In: *Proc. of the 15th Workshop on Algorithms and Data Structures (WADS 2017)*. pp. 133–144 (2017)
- [9] Bhattacharya, B., Higashikawa, Y., Kameda, T., Katoh, N.: An  $O(n^2 \log^2 n)$  time algorithm for minmax regret minsum sink on path networks. In: *Proc. of the 29th International Symposium on Algorithms and Computation (ISAAC 2018)* (2018)
- [10] Bhattacharya, B., Kameda, T.: Improved algorithms for computing minmax regret sinks on dynamic path and tree networks. *Theor. Comput. Sci.* **607**, 411–425 (2015)
- [11] Chen, D., Golin, M.J.: Sink evacuation on trees with dynamic confluent flows. In: *27th International Symposium on Algorithms and Computation (ISAAC 2016)* (2016)
- [12] Chen, D., Golin, M.J.: Minmax centered  $k$ -partitioning of trees and applications to sink evacuation with dynamic confluent flows. *CoRR* **abs/1803.09289** (2018)
- [13] Ford, L.R., Fulkerson, D.R.: Constructing maximal dynamic flows from static flows. *Operations research* **6**(3), 419–433 (1958)
- [14] Golin, M.J., Sandeep, S.: Minmax-regret  $k$ -sink location on a dynamic tree network with uniform capacities. *CoRR* **abs/1806.03814** (2018)
- [15] Hart, S., Sharir, M.: Nonlinearity of Davenport–Schinzel sequences and of generalized path compression schemes. *Combinatorica* **6**(2), 151–177 (1986)
- [16] Hershberger, J.: Finding the upper envelope of  $n$  line segments in  $O(n \log n)$  time. *Information Processing Letters* **33**(4), 169–174 (1989)
- [17] Higashikawa, Y.: Studies on the space exploration and the sink location under incomplete information towards applications to evacuation planning. PhD thesis, Kyoto University, Japan (2014)
- [18] Higashikawa, Y., Augustine, J., Cheng, S.W., Golin, M.J., Katoh, N., Ni, G., Su, B., Xu, Y.: Minimax regret 1-sink location problem in dynamic path networks. *Theor. Comput. Sci.* **588**, 24–36 (2015)
- [19] Higashikawa, Y., Cheng, S.W., Kameda, T., Katoh, N., Saburi, S.: Minimax regret 1-median problem in dynamic path networks. *Theory Comput. Syst.* **62**(6), 1392–1408 (2018)
- [20] Higashikawa, Y., Golin, M.J., Katoh, N.: Minimax regret sink location problem in dynamic tree networks with uniform capacity. *J. Graph Algorithms Appl.* **18**(4), 539–555 (2014)
- [21] Higashikawa, Y., Golin, M.J., Katoh, N.: Multiple sink location problems in dynamic path networks. *Theor. Comput. Sci.* **607**, 2–15 (2015)
- [22] Hoppe, B., Tardos, E.: The quickest transshipment problem. *Mathematics of Operations Research* **25**(1), 36–62 (2000)
- [23] Kouvelis, P., Yu, G.: *Robust Discrete Optimization and its Applications*. Kluwer Academic Publishers, London (1997)

- [24] Li, H., Xu, Y.: Minimax regret 1-sink location problem with accessibility in dynamic general networks. *Eur. J. Oper. Res.* **250**(2), 360–366 (2016)
- [25] Li, H., Xu, Y., Ni, G.: Minimax regret vertex 2-sink location problem in dynamic path networks. *Journal of Combinatorial Optimization* **31**(1), 79–94 (2016)
- [26] Mamada, S., Uno, T., Makino, K., Fujishige, S.: An  $O(n \log^2 n)$  algorithm for a sink location problem in dynamic tree networks. *Discret. Appl. Math.* **154**, 2387–2401 (2006)
- [27] Skutella, M.: An introduction to network flows over time. In: *Research Trends in Combinatorial Optimization*, pp. 451–482. Springer (2009)
- [28] Vairaktarakis, G.L., Kouvelis, P.: Incorporation dynamic aspects and uncertainty in 1-median location problems. *Naval Research Logistics (NRL)* **46**(2), 147–168 (1999)