




Minmax regret 1-sink location problems on dynamic flow path networks with parametric weights

Tetsuya Fujie¹ · Yuya Higashikawa¹ · Naoki Katoh¹ · Junichi Teruyama¹ · Yuki Tokuni¹ 

Accepted: 13 July 2024

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

Abstract

This paper addresses the minmax regret 1-sink location problem on a dynamic flow path network with parametric weights. A *dynamic flow path network* consists 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* where all the people evacuate quickly. In our model, each weight is represented by a linear function of a common parameter t , and the decision maker who determines the sink location does not know the value of t . We formulate the problem under such uncertainty as the *minmax regret problem*. Given t and sink location x , the cost 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 this cost and the optimal cost under t . The problem is to find the sink location minimizing the maximum regret over all t . For the problem, we propose an $O(n^4 2^{\alpha(n)} \alpha(n)^2 \log n)$ 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 $O(n^3 2^{\alpha(n)} \alpha(n) \log n)$.

Parts of this paper appeared in preliminary form in Proceedings of WALCOM2021 (Fujie et al. 2021).

✉ Yuki Tokuni
af23v006@guh.uhyogo.ac.jp

Tetsuya Fujie
fujie@gsis.u-hyogo.ac.jp

Yuya Higashikawa
higashikawa@gsis.u-hyogo.ac.jp

Naoki Katoh
naoki.katoh@gsis.u-hyogo.ac.jp

Junichi Teruyama
junichi.teruyama@gsis.u-hyogo.ac.jp

¹ University of Hyogo, Kobe, Japan

Keywords Minimax regret · Facility location problem · Polynomial time algorithm · Path network · Dynamic flow

1 Introduction

Recently, many parts of the world have been struck by disasters such as earthquakes, nuclear plant accidents, volcanic eruptions, and flooding, leading to the recognition of the urgent need for orderly evacuation planning. A powerful tool for evacuation planning is the *dynamic flow model* introduced by Ford and Fulkerson (1958), which represents the movement of people or goods over time in a network. In this model, we are given a graph with *source* vertices and *sink* vertices; each source vertex has a positive weight called a *supply*, each sink vertex has a positive weight called a *demand*, and each edge has positive length and capacity, with an edge capacity limiting the amount of supply that can enter the edge per unit time. One variant of the dynamic flow problem is the *quickest transshipment problem*, the objective of which is to send exactly the right amount of supply from sources to sinks while satisfying the demand constraints in the minimum overall time. Hoppe and Tardos (2000) provided a polynomial-time algorithm for this problem when the transit times are integral, but their algorithm is very complex. Therefore, finding a practical polynomial-time solution to this problem remains an open problem; see the survey by Skutella (2009) on dynamic flows.

Discussed herein is the related *sink location problem* (Belmonte et al. 2015; Benkoczi et al. 2018, 2020; Bhattacharya et al. 2017; Chen and Golin 2016, 2023; Higashikawa et al. 2014, 2015b; Mamada et al. 2006), the objective of which is to locate sinks in a given dynamic flow network so that all the supply is sent to the sinks as quickly as possible. As the criterion for optimal location, the following two can naturally be considered: (i) minimization of *evacuation completion time* or (ii) minimization of *aggregate evacuation time* (i.e., the *sum of evacuation times*), and we refer to the sink location problem with criterion (i) or (ii) as the CTSL problem or the ATSL problem, respectively. Several previous papers have studied CTSL problems (Belmonte et al. 2015; Bhattacharya et al. 2017; Chen and Golin 2016, 2023; Higashikawa et al. 2014, 2015b; Mamada et al. 2006), but for ATSL problems the results are scarce and only for path networks (Benkoczi et al. 2018, 2020; Higashikawa et al. 2015b).

To model the evacuation behavior of people, it would be natural to treat each supply as a discrete quantity as in Hoppe and Tardos (2000); Mamada et al. (2006). However, almost all previous papers on sink location problems (Belmonte et al. 2015; Bhattacharya et al. 2017; Chen and Golin 2016, 2023; Higashikawa et al. 2014, 2015b) have treated each supply as a continuous quantity, this being because doing so makes it easier to handle the problems mathematically, and also the effect of such treatment is small enough to ignore when the number of people is large. Therefore, here we likewise adopt the model with continuous supplies.

Although the above two criteria are reasonable, they may be impractical because the population distribution is assumed to be fixed. In a real situation, the number of people in an area may vary with time; for example, an office area in a large city contains many people during weekdays but far fewer people on weekends and during the night. To

account for such uncertainty, Kouvelis and Yu (1997) introduced the *minmax regret model*. In a *minmax regret sink location problem*, we are given a finite or infinite set S of *scenarios*, with each scenario giving 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 two-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. Here, we refer to the minmax regret sink location problem with the regret defined based on the evacuation completion time or the aggregate evacuation time as the MMR-CTSL problem or the MMR-ATSL problem, respectively. MMR-CTSL problems have been studied previously (Arumugam et al. 2019; Bhattacharya and Kameda 2015; Golin and Sandeep 2018; Higashikawa et al. 2015a, 2014; Li and Xu 2016; Li et al. 2016; Golin and Sandeep 2022), but there are few results for MMR-ATSL problems (Bhattacharya et al. 2018; Higashikawa et al. 2018) despite being important both theoretically and practically.

As for how to define a set of scenarios, all previous studies on minmax regret sink location problems adopted the model with *interval weights*, in which each vertex is given a 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. However, a drawback of the minmax regret model with interval weights is that each weight can take an independent value, thus we consider some extreme scenarios that may not happen in real situations, such as a scenario in which all the vertices have either maximum or minimum weights. To account for the dependency among the weights of all the vertices, we adopt the model with *parametric weights* (first introduced by Vairaktarakis and Kouvelis (1999) for the minmax regret median problem), in which each vertex is given a weight as a linear function in a common parameter t on a real interval, and a scenario is determined just by choosing t . Note that when considering a real situation, each weight function is likely to be more complex but can be approximated by a piecewise linear function. Thus, by 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 we can obtain the solution by solving multiple subproblems with linear weight functions.

Herein, we study the MMR-ATSL problem on a dynamic flow path network with parametric weights for the first time. For this problem with both general edge capacities and uniform edge capacity, we give polynomial time algorithms. Our main theorem is as follows.

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 $O(n^4 2^{\alpha(n)} \alpha(n)^2 \log n)$ time, where $\alpha(\cdot)$ is the inverse Ackermann function.*
- (ii) *When all the edge capacities are uniform, the MMR-ATSL problem can be solved in $O(n^3 2^{\alpha(n)} \alpha(n) \log n)$ time.*

Note that the MMR-ATSL problem with interval weights was studied by Bhattacharya et al. (2018), Higashikawa et al. (2018) but only for the case with uniform edge capacity.

Table 1 The current best results on MMR-CTSL problems

Edge capacity	Interval weight		Parametric weight	
	Uniform	General	Uniform	General
Path	$O(n)$ (Arumugam et al. 2019)	$O(n^4 \log n)$ (Golin and Sandeep (2022))	Open	Open
Tree	$O(n \log n)$ (Bhattacharya and Kameda 2015)	Open	Open	Open

Table 2 The current best results on MMR-ATSL problems

Edge capacity	Interval weight		Parametric weight	
	Uniform	General	Uniform	General
Path	$O(n^2 \log^2 n)$ (Bhattacharya et al. 2018)	Open	$O(n^3 2^{\alpha(n)} \alpha(n) \log n)$ (this paper)	$O(n^4 2^{\alpha(n)} \alpha(n)^2 \log n)$ (this paper)
Tree	Open	Open	Open	Open

Higashikawa et al. (2018) provided an $O(n^3)$ time algorithm, which Bhattacharya et al. (2018) improved to one running in $O(n^2 \log^2 n)$ time. However, no algorithm yet exists for the case with general edge capacities. Therefore, our result implies that the problem becomes solvable in polynomial time by introducing parametric weights. Tables 1 and 2 summarize the current best results and our results.

The rest of this paper is organized as follows. In Sect. 2, we give the notation and fundamental properties that are used throughout the paper. In Sect. 3, we give the key lemmas and the algorithm for solving the problem, 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 $\mathcal{P} = (P, \mathbf{w}(t), \mathbf{c}, \mathbf{l}, \tau)$ that 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 $E = \{e_i = (v_i, v_{i+1}) \mid 1 \leq i \leq n - 1\}$. A vector $\mathbf{w}(t) = \langle w_1(t), \dots, w_n(t) \rangle$ consists of the *weight function* $w_i : T \rightarrow \mathbb{R}_{\geq 0}$ that 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* ℓ_i of edge e_i , and τ is the *time* that 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, supply of amount w is at vertex v_{i+1} and going through edge e_i toward 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 that 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 use v_i to denote the distance between v_1 and v_i , that is, $v_i = \sum_{j=1}^{i-1} \ell_j$. For an edge $e_i \in E$, we use e_i to denote a real open interval (v_i, v_{i+1}) . We also use P to denote a real closed interval $[0, v_n]$. If a real value x satisfies $x \in (v_i, v_{i+1})$, then x is said to be a point on edge e_i to which the distance from v_i is $x - v_i$. Let $C_{i,j}$ be the minimum capacity for all the edges from e_i to e_j , that is, $C_{i,j} = \min\{c_h \mid i \leq h \leq j\}$.

Note that we precompute the values of 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* (Alstrup et al. 2002; Bender and Farach-Colton 2000).

2.1 Evacuation completion time on a dynamic flow path network

In this section, we give a formula for the evacuation completion time for fixed $x \in P$ and $t \in T$. Suppose that 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 all the supply on the right side of x as being 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 the 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 . Higashikawa (2014) gives the following formula for $\theta_R^{e_i}(x, t, z)$; For $z \in (W_{i+1,j-1}(t), W_{i+1,j}(t)]$ with $i + 1 \leq j \leq n$, we have

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

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 the 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 . Higashikawa (2014) gives the following formula for $\theta_L^{e_i}(x, t, z)$; For $z \in (W_{j+1,i}(t), W_{j,i}(t)]$ with $1 \leq j \leq i$, we have

$$\theta_L^{e_i}(x, t, z) = \max_{j \leq h \leq i} \left\{ \tau(x - v_h) + \frac{z - W_{h+1,i}(t)}{C_{h,i}} \right\}. \tag{2}$$

Let us turn to the case in which sink x is at a vertex $v_i \in V$. We confirm that the evacuation times when supply of amount z originates from the right and left sides 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., the sum of evacuation time) when the sink is at a point $x \in P$ and the weight functions are fixed by the parameter $t \in T$. For point x on edge e_i and 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 $z \in [0, W_{i+1,n}(t)]$, that is,

$$\Phi(x, t) = \int_0^{W_{1,i}(t)} \theta_L^{e_i}(x, t, z) dz + \int_0^{W_{i+1,n}(t)} \theta_R^{e_i}(x, t, z) dz. \tag{3}$$

Similarly, if sink x is at vertex v_i , then $\Phi(v_i, t)$ is given by

$$\Phi(v_i, t) = \int_0^{W_{1,i-1}(t)} \theta_L^{e_{i-1}}(v_i, t, z) dz + \int_0^{W_{i+1,n}(t)} \theta_R^{e_i}(v_i, t, z) dz. \tag{4}$$

2.3 Minmax regret formulation

We denote by $\text{Opt}(t)$ the minimum aggregate evacuation time with respect to parameter $t \in T$. Higashikawa et al. (2015b) and Benkoczi et al. (2020) showed that for the minsum k -sink location problem, there exists an optimal k -sink such that all the k sinks are at vertices. This implies that we have

$$\text{Opt}(t) = \min_{x \in V} \Phi(x, t) \tag{5}$$

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 the gap between $\Phi(x, t)$ and $\text{Opt}(t)$ defined as

$$R(x, t) = \Phi(x, t) - \text{Opt}(t). \tag{6}$$

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

$$MR(x) = \max_{t \in T} R(x, t). \tag{7}$$

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

$$\text{minimize } MR(x) \text{ subject to } x \in P. \tag{8}$$

Let x^* denote an optimal solution of (8).

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 the real interval X can be partitioned into subintervals X_1, X_2, \dots, X_m so that f is formed 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 a partition into subintervals is maximal in the sense that for any i and $i + 1$, we have $f_i \neq f_{i+1}$. 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 denotes 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:

$$\mathcal{U}_{\mathcal{F}}(y) = \max_{i=1, \dots, m} f_i(y), \quad \mathcal{L}_{\mathcal{F}}(y) = \min_{i=1, \dots, m} f_i(y), \quad (9)$$

where the maximum and the minimum are taken over those functions that are defined at y , respectively. For the upper envelope $\mathcal{U}_{\mathcal{F}}(y)$ of \mathcal{F} , there exists 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, the upper envelope $\mathcal{U}_{\mathcal{F}}(y)$ can be represented as a piecewise polynomial function. We call the above sequence $U_{\mathcal{F}} = \langle u_1, \dots, u_k \rangle$ the *function 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 (Hart and Sharir 1986; Hershberger 1989; Agarwal et al. 1989) *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 $O(n\alpha(n) \log n)$ time, where $\alpha(n)$ is the inverse Ackermann function. Moreover, if \mathcal{F} is a family of n partially defined linear functions, then $\mathcal{U}_{\mathcal{F}}$ and $\mathcal{L}_{\mathcal{F}}$ consist of $O(n\alpha(n))$ pieces, and one can obtain them in $O(n \log n)$ time. Furthermore, if \mathcal{F} is a family of n totally defined linear functions, then $\mathcal{U}_{\mathcal{F}}$ and $\mathcal{L}_{\mathcal{F}}$ consist of $O(n)$ pieces, and one can obtain them in $O(n \log n)$ time.*

Note that the number of pieces and the computation time for the upper/lower envelopes are associated with the maximum length of Davenport–Schinzel sequences; see Hart and Sharir (1986) 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 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

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

and we show the following inequality.

Property 1

$$\alpha(n^3) \leq \alpha(n) + 1 \quad (10)$$

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

Let us suppose that $n \geq 8$ and $k = \alpha(n) \geq 3$ because $A(2, 2) = 7$ holds. Inequality (10) holds if the inequality

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

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

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

The last inequality arises from the fact that

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

holds for any integer $k \geq 3$ and any positive integer $m \geq 1$, where $A(3, m) = 2^{m+3} - 3$ is shown by Porto and Matos (1980). Because $n \leq A(k, k)$ holds, we have $n^3 \leq (A(k, k))^3 \leq A(k + 1, k + 1)$. Thus, the proof is complete.

3 Algorithm

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 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 because we must treat an infinite set T . Furthermore, we are also required to find an optimal location among an infinite set e . To tackle 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 Sect. 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

Recalling (5) and (6), function $R(x, t)$ is based on function $\Phi(x, t)$. First, let us consider extracting a univariate function in t from $\Phi(x, t)$ as much as possible. According (3), $\Phi(x, t)$ is defined as the some of definite integrals $\theta_R^{e_i}(x, t, z)$ and $\theta_L^{e_i}(x, t, z)$ over z . By (1) and (2), $\theta_R^{e_i}(x, t, z)$ (resp. $\theta_L^{e_i}(x, t, z)$) can be decomposed into term $-\tau x$ and the other term independent of x , denoted by $f_R^{e_i, j}(t, z)$ (resp. $f_L^{e_i, j}(t, z)$). Formally, for $1 \leq i < j \leq n$, let function $f_R^{e_i, j}(t, z)$ be defined on $t \in T$ and $z \in (W_{i+1, j-1}(t), W_{i+1, n}(t))$ as

$$f_R^{e_i, j}(t, z) = \tau v_j + \frac{z - W_{i+1, j-1}(t)}{C_{i, j}}, \tag{11}$$

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

$$f_L^{e_i, j}(t, z) = -\tau v_j + \frac{z - W_{j+1, i}(t)}{C_{j, i}}. \tag{12}$$

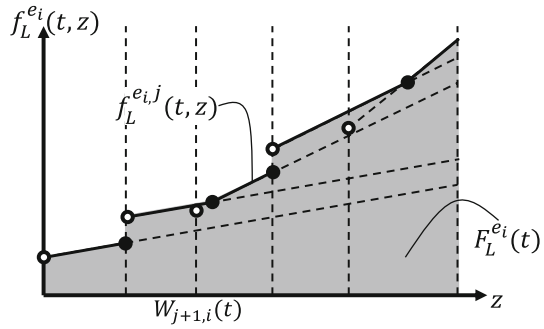
In addition, let $F_L^{e_i}(t)$ and $F_R^{e_i}(t)$ denote univariate functions defined as

$$F_L^{e_i}(t) = \int_0^{W_{1, i}(t)} f_L^{e_i}(t, z) dz, \quad F_R^{e_i}(t) = \int_0^{W_{i+1, n}(t)} f_R^{e_i}(t, z) dz, \tag{13}$$

where $f_L^{e_i}(t, z)$ and $f_R^{e_i}(t, z)$ denote functions defined as

$$f_L^{e_i}(t, z) = \max_{1 \leq j \leq i} \left\{ f_L^{e_i, j}(t, z) \right\}, \quad f_R^{e_i}(t, z) = \max_{i+1 \leq j \leq n} \left\{ f_R^{e_i, j}(t, z) \right\}. \tag{14}$$

Fig. 1 Graph of $f_L^{e_i}(t, z)$ w.r.t. z with a focus on $f_L^{e_i, j}(t, z)$. The gray area represents $F_L^{e_i}(t)$



See also Fig. 1. Recall the definition of the aggregate evacuation time $\Phi(x, t)$ given in (3). We observe that for $x \in e_i$, $\Phi(x, t)$ can be represented as

$$\begin{aligned} \Phi(x, t) &= (W_{1,i}(t) - W_{i+1,n}(t))\tau x + \int_0^{W_{1,i}(t)} f_L^{e_i}(t, z) dz + \int_0^{W_{i+1,n}(t)} f_R^{e_i}(t, z) dz \\ &= (W_{1,i}(t) - W_{i+1,n}(t))\tau x + F_L^{e_i}(t) + F_R^{e_i}(t). \end{aligned} \tag{15}$$

Similarly, by the definition in (4) and by (13), we have

$$\Phi(v_i, t) = (W_{1,i-1}(t) - W_{i+1,n}(t))\tau v_i + F_L^{e_{i-1}}(t) + F_R^{e_i}(t). \tag{16}$$

Let us focus on the function $F_L^e(t)$. As t increases, while the function sequence of $f_L^e(t, z)$ w.r.t. z remains the same, function $F_L^e(t)$ is represented as the same polynomial, whose degree is at most two by (11)–(13). In other words, a breakpoint of $F_L^e(t)$ corresponds to the value t such that the function sequence of $f_L^e(t, z)$ w.r.t. z changes. We notice that such a change happens only when then three functions $f_L^{e,h}(t, z)$, $f_L^{e,i}(t, z)$, and $f_L^{e,j}(t, z)$ intersect each other, which can happen at most once. We immediately see that $F_L^e(t)$ 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 lemmas show that the number of pieces is actually $O(n^2\alpha(n))$ and $O(n)$ for the cases with general edge capacities and uniform edge capacity, respectively.

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 $O(n^2\alpha(n))$ pieces, and they can be computed in $O(n^3\alpha(n))$ time.

Lemma 2 If all the edge capacities are uniform, for each $e \in E$, $F_L^e(t)$ and $F_R^e(t)$ are piecewise polynomial functions of degree at most two with $O(n)$ pieces, and they can be computed in $O(n^2 \log n)$ time.

Proof of Lemma 1 Let us suppose that x is on $e_i \in E$. We prove the lemma only for $F_L^{e_i}(t)$ because the case of $F_R^e(t)$ can be proved in a symmetric manner. We prove the following statements separately.

- (i) The number of pieces of $F_L^{e_i}(t)$ is $O(n^2\alpha(n))$.

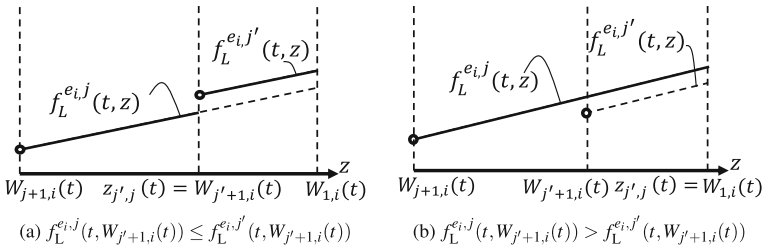


Fig. 2 Illustration of $z_{j',j}^*(t)$ for the case of $C_{j',j} = C_{j,j}$ with $1 \leq j' < j \leq i$

- (ii) The function of each piece of $F_L^{e_i}(t)$ is a piecewise polynomial function of degree at most two in t .
- (iii) All pieces of $F_L^{e_i}(t)$ can be obtained in $O(n^3\alpha(n) \log n)$ time.

First, we prove statement (i). 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 $C_{j',i} = C_{j,i}$, then we have

$$z_{j',j}^*(t) = \begin{cases} W_{j'+1,i}(t) & \text{if } f_L^{e_i,j}(t, W_{j'+1,i}(t)) \leq f_L^{e_i,j'}(t, W_{j'+1,i}(t)), \quad (17) \\ W_{1,i}(t) & \text{if } f_L^{e_i,j}(t, W_{j'+1,i}(t)) > f_L^{e_i,j'}(t, W_{j'+1,i}(t)), \quad (18) \end{cases}$$

as shown in Fig. 2, otherwise; in other words, if $C_{j',i} < C_{j,i}$, then we have

$$z_{j',j}^*(t) = \begin{cases} W_{j'+1,i}(t) & \text{if } f_L^{e_i,j}(t, W_{j'+1,i}(t)) \leq f_L^{e_i,j'}(t, W_{j'+1,i}(t)), \quad (19) \\ \min\{z_{j',j}^*(t), W_{1,i}(t)\} & \text{if } f_L^{e_i,j}(t, W_{j'+1,i}(t)) > f_L^{e_i,j'}(t, W_{j'+1,i}(t)), \quad (20) \end{cases}$$

where $z_{j',j}^*(t)$ is the solution for z of the equation $f_L^{e_i,j'}(t, z) = f_L^{e_i,j}(t, z)$ as shown in Fig. 3. Note that when t changes from $t \leq t_{j',j}$ to $t > t_{j',j}$, the function $z_{j',j}^*(t)$ changes from (17) to (18) or from (18) to (17), where $t_{j',j}$ is the solution for t of the equation $f_L^{e_i,j}(t, W_{j'+1,i}(t)) = f_L^{e_i,j'}(t, W_{j'+1,i}(t))$. The same thing holds for (19) and (20). This change happens at most once while $t \in [t^-, t^+]$.

For any $t \in T$, let $U(t)$ be the function sequence of $f_L^{e_i}(t, z)$ in z . Now, for some t' , suppose that $U(t') = \langle u_1, \dots, u_k \rangle$. Note that $u_1 > \dots > u_k$ holds. Then, the h -th smallest breakpoint of $f_L^{e_i}(t', z)$ for $z \in (0, W_{1,i}(t'))$ is $z_{u_{h+1}, u_h}(t')$. We notice that if t changes from t' under the conditions that $U(t)$ and the formula for each $z_{u_{h+1}, u_h}(t)$ remain the same, then the formula for $F_L^{e_i}(t)$ also remains the same. In other words, a breakpoint of $F_L^{e_i}(t)$ corresponds to when $U(t)$ or the formula for some $z_{u_{h+1}, u_h}(t)$ changes. Let us consider in what cases this happens.

- Case 1: When t reaches some value t_1 , the formula for some $z_{u_{h+1}, u_h}(t)$ changes from (17) to (18) or from (19) to (20) (see Fig. 4).
- Case 2: When t reaches some value t_2 , the formula for some $z_{u_{h+1}, u_h}(t)$ changes from (18) to (17) or from (20) to (19) (see Fig. 5).

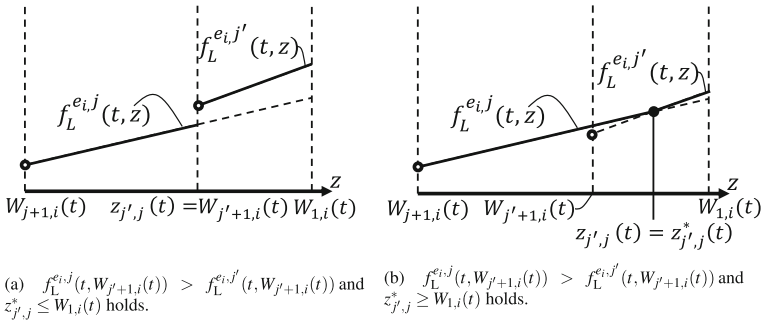


Fig. 3 Illustration of $z_{j',j}^*(t)$ for the case of $C_{j',j} < C_{j,j}$ with $1 \leq j' < j \leq i$

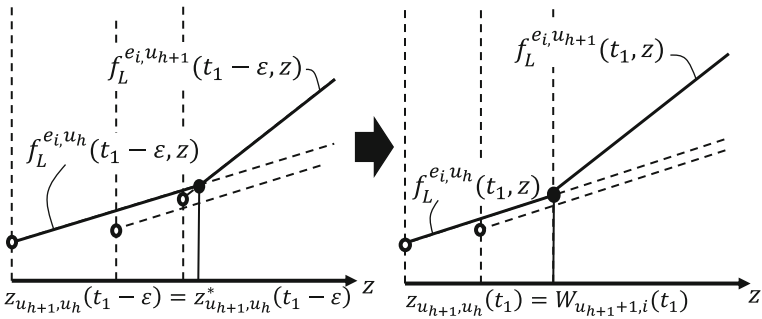


Fig. 4 When t reaches t_1 , the formula for some $z_{u_{h+1},u_h}(t)$ changes to $W_{u_{h+1}+1,i}(t)$ from z_{u_{h+1},u_h}^*

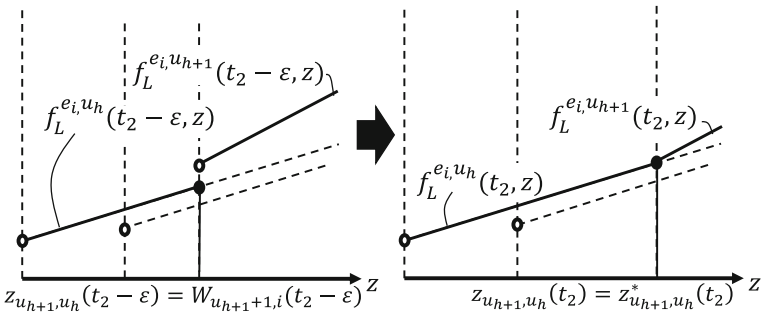


Fig. 5 When t reaches t_2 , the formula for some $z_{u_{h+1},u_h}(t)$ changes to z_{u_{h+1},u_h}^* from $W_{u_{h+1}+1,i}(t)$

Case 3: Just after t reaches some value t_3 , that is, when $t = t_3 + \epsilon$, $f_L^{e_i,j}(t_3 + \epsilon, z)$ for some j appears as a part of $f_L^{e_i}(t_3 + \epsilon, z)$ so that $U(t_3 + \epsilon) = \langle u_1, \dots, u_h, j, u_{h+1}, \dots, u_k \rangle$ (see Fig. 6).

Case 4: When t reaches some value t_4 , $f_L^{e_i,u_h}(t_4, z)$ disappears from $f_L^{e_i}(t_4, z)$, that is, $U(t_4) = \langle u_1, \dots, u_h, u_{h+2}, \dots, u_k \rangle$ (see Fig. 7).

Note that t_1 and t_2 are the solutions of

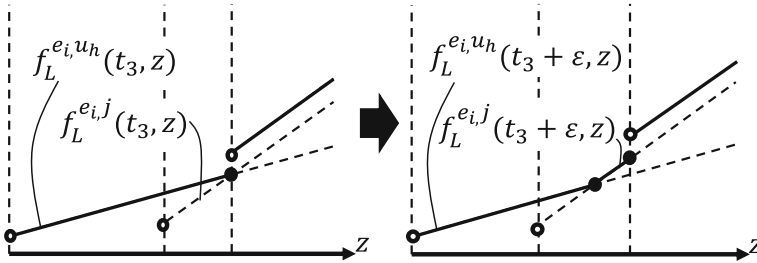


Fig. 6 When $t = t_3 + \epsilon$, $f_L^{e_i, j}(t_3 + \epsilon, z)$ appears as a part of $f_L^{e_i}(t_3 + \epsilon, z)$ so that $U(t_3 + \epsilon) = \langle \dots, u_h, j, u_{h+1}, \dots \rangle$

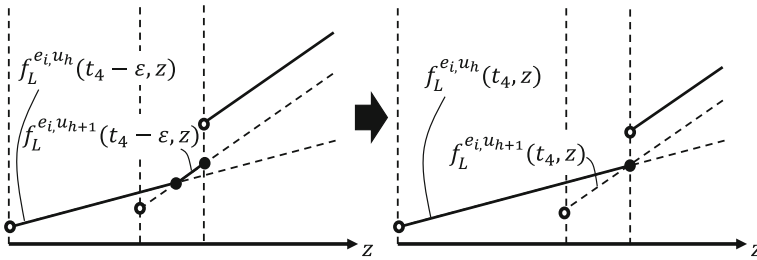


Fig. 7 When t reaches t_4 , $f_L^{e_i, u_h}(t_4, z)$ disappears from $f_L^{e_i}(t_4, z)$, that is, $U(t_4) = \langle \dots, u_h, u_{h+2}, \dots \rangle$

$f_L^{e_i, u_h}(t, W_{u_{h+1}+1, i}(t)) = f_L^{e_i, u_{h+1}}(t, W_{u_{h+1}+1, i}(t))$. For any $t \in T$ and j with $2 \leq j \leq i$, $z_j(t)$ is defined as

$$z_j(t) = \min_h \{z_{h, j}(t) \mid 1 \leq h \leq j - 1\}. \tag{21}$$

Then, t_1, t_2, t_3 , and t_4 in Cases 1–4 are breakpoints of $z_{u_h}(t)$. Therefore, the number of breakpoints of $F_L^{e_i}(t)$ is at most the number of all the breakpoints of the functions $z_j(t)$ for all j with $2 \leq j \leq i$. Recall that $z_j(t)$ is the lower envelope of $O(j)$ line segments by (17)–(21). Thus, the number of breakpoints of $z_j(t)$ for any j is $O(n\alpha(n))$ by Theorem 2. Therefore, the total number of breakpoints of the functions $z_j(t)$ for all j with $2 \leq j \leq i$ is $O(n^2\alpha(n))$. This completes the proof for property (i).

Next, we prove statement (ii). 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 the formula for each $z_{u_{h+1}, u_h}(t)$ remains the same. We then have

$$H(t) = \sum_{h=1}^k \int_{z_{u_h, u_{h-1}}(t)}^{z_{u_{h+1}, u_h}(t)} f_L^{e_i, u_h}(t, z) dz, \tag{22}$$

where $z_{u_1, u_0}(t) = 0$ and $z_{u_{k+1}, u_k}(t) = W_{1, i}(t)$. By (12), it holds for h with $1 \leq h \leq k$ that

$$f_L^{e_i, u_h}(t, z) = \rho_1^h z + \rho_2^h t + \rho_3^h, \tag{23}$$

where $\rho_1^h, \rho_2^h, \rho_3^h$ are some constants. Substituting (23) into (22), we obtain

$$H(t) = \sum_{h=1}^k \left\{ \frac{\rho_1^h}{2} \left(z_{u_{h+1}, u_h}(t)^2 - z_{u_h, u_{h-1}}(t)^2 \right) + (\rho_2^h t + \rho_3^h) \left(z_{u_{h+1}, u_h}(t) - z_{u_h, u_{h-1}}(t) \right) \right\}, \tag{24}$$

which is quadratic in t because $z_{u_{h+1}, u_h}(t)$ and $z_{u_h, u_{h-1}}(t)$ are linear in t by (17)–(20). Note that if the coefficient of t^2 in the function $H(t)$ of (24) is zero, then $H(t)$ is a linear function. This completes the proof for property (ii).

Finally, we prove statement (iii). 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 $z_j(t)$, which is a lower envelope of $z_{j', j}(t)$ for integers j' with $1 \leq j' < j$ by (21). Because $z_{j', j}(t)$ is a piecewise linear function with at most three pieces, we can obtain all breakpoints of $z_j(t)$ in $O(n \log n)$ time for each j by Theorem 2. Thus, Step 1 requires $O(n^2 \log n)$ time.

Step 2: Obtain functions of all pieces of $F_L^{e_i}(t)$. Let T' be an interval of a piece of $F_L^{e_i}(t)$. Choosing the parameter $t \in T'$, we obtain a function sequence $U(t) = \langle u_1, \dots, u_k \rangle$ of $f_L^{e_i}(t, z)$. Because $f_L^{e_i}(t, z)$ is the upper envelope of partially defined linear functions by (12) and (14), $U(t)$ can be computed in $O(n \log n)$ time by Theorem 2. We obtain a polynomial of degree at most two in $O(n)$ time by evaluating (24). Because the number of pieces of $F_L^{e_i}(t)$ is $O(n^2 \alpha(n))$, Step 2 requires $O(n^3 \alpha(n) \log n)$ time.

In total, our algorithm requires $O(n^2 \log n + n^3 \alpha(n) \log n) = O(n^3 \alpha(n) \log n)$ time. □

Proof of Lemma 2 Let us suppose that x is on $e_i \in E$. We prove the lemma only for $F_L^{e_i}(t)$ because the case of $F_R^e(t)$ can be proved in a symmetric manner. We prove the following statements separately, as in the proof of Lemma 1.

- (i) The number of pieces of $F_L^{e_i}(t)$ is $O(n)$.
- (ii) The function of each piece of $F_L^{e_i}(t)$ is a piecewise polynomial function of degree at most two in t .
- (iii) All pieces of $F_L^{e_i}(t)$ can be obtained in $O(n^2 \log n)$ time.

Because statement (ii) has been shown in the proof of Lemma 1, we will prove only statements (i) and (iii).

First, we prove statement (i). For any $t \in T$, let $U(t)$ be the function sequence of $f_L^{e_i}(t, z)$ in z as in Lemma 1. Note that only Cases 3 and 4 in the proof of Lemma 1 happen when all edge capacities are uniform. Therefore, some t' is a breakpoint of $F_L^{e_i}(t)$ if and only if $U(t)$ changes at $t = t'$. In the following, we will prove that any index comes into $U(t)$ at most once, and therefore $U(t)$ changes $O(n)$ times. Suppose that when t reaches t' , j comes out from $U(t)$. We then show that j never comes into $U(t)$ for any t with $t' < t$. In this case, some j' with $j < j' \leq i$ comes to dominate j at time t' ; that is, when $t = t' - \varepsilon$ for sufficiently small $\varepsilon > 0$, $f_L^{e_i, j'}(t, z)$ is above $f_L^{e_i, j}(t, z)$ for $z \in (W_{j+1, i}(t), W_{1, i}(t)]$, and when $t = t'$, $f_L^{e_i, j}(t, z) = f_L^{e_i, j'}(t, z)$

holds for $z \in (W_{j+1,i}(t'), W_{1,i}(t'))$ because the slopes in z of $f_L^{e_i,h}(t, z)$ for all h are the same. Recall that in the proof of Lemma 1, the function $z_{j',h}(t)$ for $h = 1, \dots, j'$ changes from (17) to (18) or from (18) to (17) at most once. Therefore, we can see that while $t > t'$, $f_L^{e_i,j}(t, z)$ remains below $f_L^{e_i,j'}(t, z)$ for $z \in (W_{j+1,i}(t), W_{1,i}(t))$. Thus, the number of changes of $U(t)$ w.r.t. j is at most two. Because a breakpoint of $F_L^{e_i}(t)$ corresponds to t where some j comes either into or out from $U(t)$, the number of breakpoints of $F_L^{e_i}(t)$ is $O(n)$.

Next, we prove statement (iii). We need only prove that all of $O(n)$ breakpoints of $F_L^{e_i}(t)$ can be obtained in $O(n^2 \log n)$ time because after that, we can apply the same operation as in Step 2 in the proof of Lemma 1 for computing $O(n)$ pieces of $F_L^{e_i}(t)$. Let us consider a value $t = t'$ where $U(t)$ changes. Then, for some j with $1 \leq j \leq i$, the index of the function consisting of $f_L^{e_i}(t, z)$ over $z \in (W_{j+1,i}(t), W_{j,i}(t))$ changes at $t = t'$. To compute such t' , it is enough to consider $f_L^{e_i}(t, z)$ with $z = W_{j,i}(t)$; that is, t' is a breakpoint of the univariate function $g^j(t) := f_L^{e_i}(t, W_{j,i}(t))$. Thus, t' is a value where $U(t)$ changes if and only if t' is a breakpoint of $g^j(t)$ for some j with $1 \leq j \leq i$.

From the above discussion, we can compute $F_L^{e_i}(t)$ by the following two steps.

Step 1: Obtain breakpoints of $F_L^{e_i}(t)$. For all $j = 1, \dots, i$, compute the breakpoints of $g^j(t)$. The function $f_L^{e_i,h}(t, W_{j,i}(t))$ is a linear function for any t in $[t^-, t^+]$ by substituting $W_{j,i}(t)$ into (12). For any j with $1 \leq j \leq i$, $g^j(t)$ is the upper envelope of $f_L^{e_i,h}(t, W_{j,i}(t))$ for all h with $j \leq h \leq i$, so we can compute the breakpoints of $g^j(t)$ in $O(n \log n)$ time and the number of breakpoints is $O(n)$ by Theorem 2. Therefore, we can compute the breakpoints of $g^j(t)$ for all j with $1 \leq j \leq i$ in $O(n^2 \log n)$ time. Note that some breakpoints may be obtained multiple times. Then, by removing duplicates, we obtain all $O(n)$ breakpoints of $F_L^{e_i}(t)$. Because the number of obtained breakpoints before removing duplicates is $O(n^2)$, this operation takes $O(n^2 \log n)$ time. In total, Step 1 takes $O(n^2 \log n)$ time.

Step 2: Obtain functions of all pieces of $F_L^{e_i}(t)$. The number of pieces of $F_L^{e_i}(t)$ is $O(n)$ by statement (i). Therefore, by the same operation as that of Step 2 of the proof of Lemma 1, we can obtain polynomial functions of all the pieces of $F_L^{e_i}(t)$ in $O(n^2 \log n)$ time.

In total, our algorithm requires $O(n^2 \log n)$ time to compute $F_L^{e_i}(t)$. □

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 $N_F = O(n^2 \alpha(n))$, 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 Lemmas 1 and 2 imply the following lemma.

Lemma 3 *Opt(t) is a piecewise polynomial function of degree at most two with $O(nN_F 2^{\alpha(n)})$ pieces, and it 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 Equation (5) 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 Lemmas 1 and 2, because $F_L^e(t)$ and $F_R^e(t)$ for any $e \in E$ are piecewise polynomial functions 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 fact that $N_F = O(n^3)$ and Property 1. This completes the proof. \square

Let N_{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 (15) for (6), 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

$$G_j^{e_i}(x) = \max\{R(x, t) \mid t \in T_j^{e_i}\}. \tag{25}$$

We then have the following lemma.

Lemma 4 *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 it 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

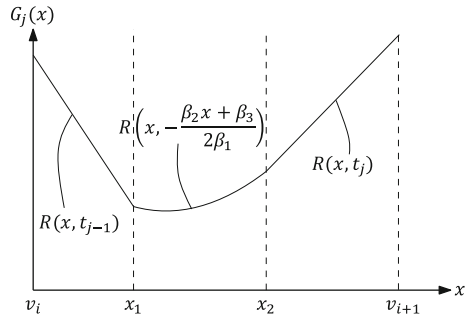
$$R(x, t) = \Phi(x, t) - \text{Opt}(t) = (W_{1,i}(t) - W_{i+1,n}(t))\tau x + F_L^{e_i}(t) + F_R^{e_i}(t) - \text{Opt}(t).$$

Because $W_{1,i}(t)$, $W_{i+1,n}(t)$ are linear in t and $F_L^{e_i}(t) + F_R^{e_i}(t) - \text{Opt}(t)$ is a 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 β_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 consider 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 $\beta_1 \geq 0$, then $R(x, t)$ 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 most two pieces, because both $R(x, t_{j-1})$ and $R(x, t_j)$ are linear in x . Let us consider the other case in which $\beta_1 < 0$ holds.

Fig. 8 Graph of $G_j(x)$ in the case of $v_i < x_1 < x_2 < v_{i+1}$



When the axis of symmetry of $R(x, t)$ w.r.t. t , that is, $t = -(\beta_2 x + \beta_3)/(2\beta_1)$, is contained in T_j , it holds that $G_j(x) = R(x, -(\beta_2 x + \beta_3)/(2\beta_1))$. Thus we have

$$G_j(x) = \begin{cases} R(x, t_{j-1}) & \text{if } -\frac{\beta_2 x + \beta_3}{2\beta_1} < t_{j-1}, \\ R\left(x, -\frac{\beta_2 x + \beta_3}{2\beta_1}\right) & \text{if } t_{j-1} \leq -\frac{\beta_2 x + \beta_3}{2\beta_1} \leq t_j, \\ R(x, t_j) & \text{if } -\frac{\beta_2 x + \beta_3}{2\beta_1} > t_j. \end{cases} \tag{26}$$

Note that the inequality conditions in (26) 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 the equation $-(\beta_2 x + \beta_3)/(2\beta_1) = t_{j-1}$ and x_2 is the solution for x of the equation $-(\beta_2 x + \beta_3)/(2\beta_1) = t_j$ (see Fig. 8). Therefore, $G_j(x)$ is a piecewise polynomial function with at most three polynomials 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 5 *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 $x \in e$. Because functions $F_L^e(t), F_R^e(t)$, and $\text{Opt}(t)$ are available, we can apply Lemma 4 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 $G_1^e(x), \dots, G_{N_e}^e(x)$. Because $G_j^e(x)$ 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 that minimizes $MR(x)$ in constant time, and from 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 a small modification to the algorithm of Lemma 5 allows us to also compute $MR(v)$ for all $v \in V$ in $O(N_{\text{Opt}})$ time.

Lemma 6 *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 $O(N_{\text{Opt}})$ time to obtain $MR(v_i)$ for each $v_i \in V$. Substituting (16) for (6), 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 Algorithm and time analysis

Finally, let us give an algorithm that finds a sink location that minimizes the maximal regret, as well as an 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 pre-process. Applying Lemmas 1–3, 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 $O(nN_{\text{Opt}}\alpha(n) \log n)$ time by applying Lemma 5. Then, we also compute $MR(v)$ for all $v \in V$ in $O(nN_{\text{Opt}})$ time by applying Lemma 6. Finally, we find an optimal sink location x^* in $O(n)$ time by evaluating $MR(x)$ for $x \in \{x^{*,e}\} \cup V$.

Because we have $N_{\text{Opt}} = O(nN_F 2^{\alpha(n)})$, the bottleneck of our algorithm is in computing $x^{*,e}$ 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 $N_F = O(n^2 \alpha(n))$, and for the case with uniform edge capacity, $N_F = O(n)$.

4 Concluding remarks

MMR-ATSL addresses a situation where sink locations are decided under the uncertainty that population changes by time. We have presented an $O(n^4 2^{\alpha(n)} \alpha(n)^2 \log n)$

time algorithm for MMR-ATSL on dynamic flow path networks with parametric weights. When all the edge capacities are uniform, MMR-ATSL can be solved in $O(n^3 2^{\alpha(n)} \alpha(n) \log n)$ time.

MMR-ATSL on other networks as trees, grids or more general networks is still open. Also, MMR-CTSL has not been solved for any network class. It is worth designing algorithms for these problems in order to make an effective evacuation plan.

Funding This work is supported by JSPS KAKENHI Grant Numbers 19H04068, 20H05794, 22K11910, 23H03349, and the preparation for the establishment of university fellowships towards the creation of science technology innovation.

Data availability Data sharing is not applicable to this article as no datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors have no conflict of interest.

References

- Agarwal PK, Sharir M, Shor P (1989) Sharp upper and lower bounds on the length of general davenport-schinzel sequences. *J Comb Theory Ser A* 52(2):228–274
- Alstrup S, Gavoille C, Kaplan H, Rauhe T (2002) Nearest common ancestors: a survey and a new distributed algorithm. In: Proceedings of the 14th annual ACM symposium on Parallel algorithms and architectures (SPAA 2002), pp 258–264
- Arumugam GP, Augustine J, Golin MJ, Srikanthan P (2019) Minmax regret k -sink location on a dynamic path network with uniform capacities. *Algorithmica* 81(9):3534–3585
- Belmonte R, Higashikawa Y, Katoh N, Okamoto Y (2015) Polynomial-time approximability of the k -sink location problem. *CoRR arXiv:1503.02835*
- Bender MA, Farach-Colton M (2000) The LCA problem revisited. In: Proceedings of the 4th Latin American Symposium on Theoretical Informatics (LATIN 2000), pp 88–94
- Benkoczi R, Bhattacharya B, Higashikawa Y, Kameda T, Katoh N (2018) Minsum k -sink problem on dynamic flow path networks. In: Proceedings of the 29th international workshop on combinatorial algorithms (IWCOA 2018), pp 78–89
- Benkoczi R, Bhattacharya B, Higashikawa Y, Kameda T, Katoh N (2020) Minsum k -sink problem on path networks. *Theor Comput Sci* 806:388–401
- Bhattacharya B, Kameda T (2015) Improved algorithms for computing minmax regret sinks on dynamic path and tree networks. *Theor Comput Sci* 607:411–425
- Bhattacharya B, Golin MJ, Higashikawa Y, Kameda T, Katoh N (2017) Improved algorithms for computing k -sink on dynamic flow path networks. In: Proceedings of the 15th workshop on algorithms and data structures (WADS 2017), pp 133–144
- Bhattacharya B, Higashikawa Y, Kameda T, Katoh N (2018) An $O(n^2 \log^2 n)$ time algorithm for min-max regret minsum sink on path networks. In: Proceedings of the 29th international symposium on algorithms and computation (ISAAC 2018)
- Chen D, Golin MJ (2016) Sink evacuation on trees with dynamic confluent flows. In: 27th international symposium on algorithms and computation (ISAAC 2016)
- Chen D, Golin MJ (2023) Minmax centered k -partitioning of trees and applications to sink evacuation with dynamic confluent flows. *Algorithmica* 85(7):1948–2000
- Ford LR, Fulkerson DR (1958) Constructing maximal dynamic flows from static flows. *Oper Res* 6(3):419–433
- Fujie T, Higashikawa Y, Katoh N, Teruyama J, Tokuni Y (2021) Minmax regret 1-sink location problems on dynamic flow path networks with parametric weights. *Algorithms and Computation, WALCOM*, pp 52–64

- Golin MJ, Sandeep S (2018) Minmax-regret k -sink location on a dynamic tree network with uniform capacities. CoRR [arXiv:1806.03814](https://arxiv.org/abs/1806.03814)
- Golin MJ, Sandeep S (2022) Minmax regret for sink location on dynamic flow paths with general capacities. *Discrete Appl Math* 315:1–26
- Hart S, Sharir M (1986) Nonlinearity of Davenport-Schinzel sequences and of generalized path compression schemes. *Combinatorica* 6(2):151–177
- Hershberger J (1989) Finding the upper envelope of n line segments in $O(n \log n)$ time. *Inf Process Lett* 33(4):169–174
- Higashikawa Y (2014) Studies on the space exploration and the sink location under incomplete information towards applications to evacuation planning. Ph.D. thesis, Kyoto University, Japan
- Higashikawa Y, Golin MJ, Katoh N (2014) Minimax regret sink location problem in dynamic tree networks with uniform capacity. *J Graph Algorithms Appl* 18(4):539–555
- Higashikawa Y, Augustine J, Cheng SW, Golin MJ, Katoh N, Ni G, Su B, Xu Y (2015) Minimax regret 1-sink location problem in dynamic path networks. *Theor Comput Sci* 588:24–36
- Higashikawa Y, Golin MJ, Katoh N (2015) Multiple sink location problems in dynamic path networks. *Theor Comput Sci* 607:2–15
- Higashikawa Y, Cheng SW, Kameda T, Katoh N, Saburi S (2018) Minimax regret 1-median problem in dynamic path networks. *Theory Comput Syst* 62(6):1392–1408
- Hoppe B, Tardos E (2000) The quickest transshipment problem. *Math Oper Res* 25(1):36–62
- Kouvelis P, Yu G (1997) Robust discrete optimization and its applications. Kluwer Academic Publishers, London
- Li H, Xu Y (2016) Minimax regret 1-sink location problem with accessibility in dynamic general networks. *Eur J Oper Res* 250(2):360–366
- Li H, Xu Y, Ni G (2016) Minimax regret vertex 2-sink location problem in dynamic path networks. *J Comb Optim* 31(1):79–94
- Mamada S, Uno T, Makino K, Fujishige S (2006) An $O(n \log^2 n)$ algorithm for a sink location problem in dynamic tree networks. *Discrete Appl Math* 154:2387–2401
- Porto A, Matos A (1980) Ackermann and the superpowers. *ACM SIGACT News* 12:90–95. <https://doi.org/10.1145/1008861.1008872>
- Skutella M (2009) An introduction to network flows over time. *Research trends in combinatorial optimization*. Springer, Berlin, pp 451–482
- Vairaktarakis GL, Kouvelis P (1999) Incorporation dynamic aspects and uncertainty in 1-median location problems. *Naval Res Log (NRL)* 46(2):147–168

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.