# Improved complexity results for the robust mean absolute deviation problem on networks with linear vertex weights

Igor Averbakh[1,2], Oded Berman[2], and Marina Leal[2,3]

[1] Department of Management, University of Toronto Scarborough, 1265 Military Trail, Toronto, Ontario M1C 1A4, Canada;

[2] Rotman School of Management, University of Toronto, 105 St. George Street, Toronto, Ontario M5S3E6, Canada;

[3] Department of Statistics and Operations Research, Faculty of Mathematics, Campus Reina Mercedes, 41011, University of Seville, Seville, Spain.

emails: averbakh@utsc.utoronto.ca, berman@rotman.utoronto.ca, marinalealpalazon@gmail.com

November 28, 2016

**Abstract**. In a recent paper [16], an algorithmic approach was presented for the robust (minmax regret) absolute deviation single-facility location problem on networks with node weights which are linear functions of an uncertain or dynamically changing parameter. The problem combines the mean absolute deviation criterion, which is the weighted average of the absolute deviations of individual customer-facility distances from the mean customer-facility distance and is one of the standard measures of "inequality" between the customers, with the minmax regret approach to optimization under uncertainty. The uncertain data are node weights (demands) which are assumed to change in a correlated manner being linear functions of a single uncertain parameter. The analysis in [16] presented complexity bounds that are polynomial but way too high to be of practical value. In this paper, we present algorithmic and analytical improvements that significantly reduce the computational complexity bounds for the algorithm.

**Key words:** minmax regret, robust location, facility location under uncertainty, network algorithms.

# 1 Introduction

Significant research efforts have been devoted to facility location problems with uncertainty in data. In the last two decades, much attention has been focused on the minmax regret approach (e.g., [1, 2, 3, 4, 6, 7, 8, 9, 12, 13, 14, 15, 19, 20, 24]), where it is required to find a solution with an objective function value reasonably close to the optimal one for all possible realizations of data. For location problems on networks, there may be two types of uncertain parameters: node weights and/or edge lengths (transportation times). The literature on minmax regret location problems on networks is primarily focused on problems with uncertain node weights that may represent the sizes of the corresponding population centers, or their demands. Many facility location decisions are strategic in nature and correspond to locating facilities that are supposed to function for a long time, so uncertainty in node weights may reflect the population dynamics, or potential changes in demand. Models with uncertain edge lengths have fewer practical applications. Also, minmax regret versions of classical single-facility location problems such as 1-center or 1-median are strongly NP-hard on general networks if uncertainty in edge lengths is present [2], in contrast to single-facility problems with uncertain node weights that are typically polynomially solvable (e.g., [3, 4]). Thus, models with uncertain edge lengths have less potential for advanced algorithmic analysis.

The most commonly used approaches to representing uncertain data in minmax regret optimization are the interval data model, where it is assumed that each uncertain parameter can take on any value within its pre-specified uncertainty interval regardless of the values taken by other parameters, and the discrete scenario model, where a finite set of possible realizations of the vector of uncertain parameters is explicitly given as a part of the input [12, 14, 15]. One of the less studied approaches is the model where uncertain/changing parameters (e.g., node weights) are assumed to be linearly related, i.e. to be linear functions of a single parameter (e.g., time).

The literature on minmax regret facility location largely focuses on minmax regret versions of classical location objectives such as the median or center objectives. In [16], the minmax regret version of a location problem with an equity-based objective was considered; such an objective is an important service measure in the public sector. The mean absolute deviation is the weighted average of the absolute deviations of customer-facility distances from the mean distance between the facility and the customers, and is a measure of "inequality" between the customers that was extensively studied for problems without uncertainty or dynamic evolution of the parameters [5, 11, 17, 18]. Lopez-de-los-Mozos et al. [16] consider this objective in the context of dynamic evolution (or linearly-correlated uncertainty) of node weights which are assumed to be linear functions of a changing parameter (time, for instance), and study the minmax and minmax regret versions of the problem. For the minmax regret version, they develop a polynomial time algorithm, which, in spite of being polynomial, cannot be considered practically useful due to very high order of complexity obtained by the analysis in [16] (slightly higher than $O(n^8 \log n)$ for trees and slightly higher than $O(m^3 n^8 \log n)$ for general networks, where $n$ and $m$ are the numbers of vertices and edges, respectively). In this paper, we present improvements that reduce the complexity bounds to $O(m^2 n^3 (\log^* n)^2 \log n)$ for general networks, $O(n^4 (\log^* n)^2 \log n)$ for trees, and $O(n^3 (\log^* n)^2 \log n)$ for paths, where $\log^* n$ is the iterated logarithm which is a function that grows extremely slowly and can be considered as a constant for practical purposes.

## 2 Notation and the problem

Consider a network $N(V, E)$ with the set of nodes $V = \{v_1, ..., v_n\}$ and the set $E$ of $m$ rectifiable edges. We use the following notation (consistent with [16]):

- $l_e$ is the length of the edge $e \in E$;
- $N$ denotes the set of all points of the network, or the network itself;
- $d(x, y)$ is the shortest-path distance in the network between points $x$ and $y$;
- $w_i(t) = \alpha_i t + \beta_i \geq 0$ is the weight of node $v_i$ which is a linear function of a real-valued parameter $t$ (time, for instance) that can take any value from an interval $[t^-, t^+]$, for some constants $t^-, t^+,\ t^- \leq t^+$;
- $W(t) = \sum_{i=1}^n w_i(t) = At + B$, where $A = \sum_{i=1}^n \alpha_i$, $B = \sum_{i=1}^n \beta_i$, is the total weight, which is assumed to be strictly positive for any $t \in [t^-, t^+]$;
- $M(x, t) = \frac{1}{W(t)} \sum_{i=1}^n w_i(t) d(v_i, x)$ is the dynamic median function, for $t \in [t^-, t^+]$ and $x \in N$;
- $F(x, t) = \frac{1}{W(t)} \sum_{i=1}^n w_i(t) |d(v_i, x) - M(x, t)|$ is the dynamic mean absolute deviation (MAD) function;
- $F^*(t) = \min_{x \in N} F(x, t)$.

We interpret the nodes of $N$ as population centers, the weights $w_i(t)$ as the numbers of customers at the population centers, and $x$ as the location for a service facility that serves the customers. Different interpretations of the uncertain parameter $t$ are possible; e.g., it may be a summary indicator of the area affluence which defines the predicted sizes of the population centers, or the uncertain popularity index of the services provided by the facility that defines the predicted future demand for the service from different population centers depending on the centers' population characteristics (affluence, ethnic/cultural/social composition, etc.) However, for convenience of presentation and for consistency with literature (e.g., [16]), we interpret $t$ as time. Then, $M(x,t)$ is the average customer-facility distance at time $t$, and $F(x,t)$ is the mean absolute deviation of individual customer-facility distances from the average customer-facility distance at time $t$. $F(x,t)$ is one of the standard measures of "inequality" between customers [5].

Following [16], the minmax regret mean absolute deviation location problem (MMR-MAD) is

$$\min_{x\in N}\max_{t\in[t^-,t^+]}\left(F(x,t)-F^*(t)\right).$$

MMR-MAD is the minmax regret version of the problem of finding a location $x$ that minimizes the mean absolute deviation $F(x,t)$ considering $t$ as uncertain parameter that can take values in $[t^-,t^+]$. Different values of $t$ define different scenarios, and we want to find a location $x$ that minimizes the worst-case loss of optimality, or "opportunity loss" $F(x,t)-F^*(t)$, across all possible scenarios that correspond to the interval (period) of interest $[t^-,t^+]$.

In the complexity estimates, we will use some standard notation and facts from computational geometry:

- $\log^* n$ is the iterated logarithm function (the minimum number of times that the logarithm (base 2) operator should be applied to $n$ to obtain a number not greater than 1). $\log^* n$ is a very slowly growing function and can be considered almost constant for practical purposes [22];
- $\lambda_s(n)$ is the maximum length of a Davenport-Schinzel sequence of order $s$ on $n$ symbols. It is well known that for a set of $x$-monotone Jordan arcs with at most $s$ intersections between any pair of arcs, its lower envelope has an $O(\lambda_{s+2}(n))$ complexity, and can be computed in $O(\lambda_{s+1}(n)\log n)$ time ([22], Theorem 6.5). For any $s$, $\lambda_s(n)=O(n\log^* n)$ [23];
- $\alpha(n)$ is the inverse of the Ackermann function which grows very slowly, $\alpha(n)\leq 4$ for any "practical" values of $n$ [21].

For any integer $i,j$, $i\leq j$, let $[i:j]$ denote the set $\{i,i+1,...,j\}$.

Lopez-de-los-Mozos et al. [16] present an algorithm for MMR-MAD and analyze its complexity. Their analysis obtains the complexity bound $O(mn^3\lambda_6(m^2n^5)\log^* n\log n)$ (even though stated as $O(mn^3\lambda_6(n^5)\log^* n\log n)$ in [16], as we will discuss in Subsection 4.4) for general networks, which is slightly higher than $O(m^3n^8\log n)$, and the bound $O(n^3\lambda_6(n^5)\log^* n\log n)$ for trees which is slightly higher than $O(n^8\log n)$. These orders of complexity, although polynomial, are clearly of little practical value, and the purpose of this paper is to improve them. To describe the improvements, we need first to outline the algorithm from [16].

To facilitate reading the paper, in the next section we give an informal sketch of the approach of [16] and our improvements, with minimum notation and details. The purpose of this is to outline a "big picture" and some ideas without delving into technicalities. In Section 4 we give a more detailed (but still compressed) description of the algorithm and analysis of [16] which is necessary for rigorous presentation of our improved analysis, and in Sections 5 and 6 we present our improvements. In Section 7, we make some concluding remarks.

# 3 A sketch of the ideas of the approach of [16] and our improvements

The logic of the algorithm from [16] for MMR-MAD is essentially the same for a tree and for a general network. On a tree, the problem restricted to a single edge is considered, and then the best of the optimal solutions for different edges is chosen. On a general network, each edge is partitioned into $O(n)$ primary regions where all functions $d(v_i, x)$, $i \in [1 : n]$ are linear. The problem restricted to a primary region of an edge is considered, and then the best of the optimal solutions for different primary regions of different edges is chosen. The reason to consider separately each edge of a tree or each primary region of each edge of a general network is that this makes all facility-customer distances as functions of the facility location linear which simplifies the structure of the objective.

## 3.1 The case of a tree.

Suppose that $N$ is a tree. The big-picture scheme of the approach of Lopez-de-los-Mozos et al. [16] for MMR-MAD on a tree is as follows. First, they represent function $F^*(t)$ as the lower envelope of a polynomial collection of simple functions of $t$ partially defined on $[t^-, t^+]$, and obtain this lower envelope $L(t)$ explicitly using standard computational geometry techniques. By "*simple functions*" we mean differentiable functions, rational or algebraic, each having a single explicit analytic expression over its domain, such that the number of intersection points between any two of them is bounded by a constant. The lower envelope $L(t)$ is a piece-wise differentiable function with a polynomial number of differentiable arcs (represented by simple functions) and breakpoints expressed explicitly. Second, using the obtained representation of $F^*(t)$, they represent the single-variable function $\max_{t\in[t^-,t^+]}(F(x,t) - F^*(t))$ over each edge $e$ of the tree as the upper envelope of a polynomial collection of simple functions of $x$, obtain this upper envelope explicitly, choose its minimum point over the edge $e$, compare the minimum points over all edges and choose the best one. On a general network, the approach is similar, with primary regions of edges instead of edges of a tree. Let us now discuss the main technical ideas of the algorithm of [16], still at the intuitive non-detailed level and with minimum notation.

Consider the function $F(x,t)$ in the domain $D_e$ which is the Cartesian product of an edge $e$ of the tree $N$ and the interval $[t^-,t^+]$. In the interior of $D_e$, function $F(x,t)$ is differentiable everywhere except the points of at most $n$ breakpoint curves $\mathcal{B}_{e,i}$, $i \in [1:n]$ that satisfy $d(v_i,x) - M(x,t) = 0$. The breakpoint curves are hyperbolas or straight lines and partition the domain $D_e$ into full-dimensional cells inside which $F(x,t)$ is differentiable. This partition has $O(n^2)$ cells and $O(n^2)$ vertices (intersection points of the breakpoint curves with other breakpoint curves or the boundary of $D_e$) and is denoted $P(D_e)$. In the paper, we use the term "vertices" for vertices of geometric arrangements, while the term "nodes" is used for the nodes of $N$.

In Phase 1 of the algorithm from [16], function $F^*(t)$ is obtained. For any fixed $t \in [t^-,t^+]$ the minimum of $F(x,t)$ over all $x \in N$ is attained at some breakpoint curve or at a node of $N$, since $F(x,t)$ as a function of $x$ is piece-wise linear convex on $e$ with breakpoints at the breakpoint curves. Hence, for obtaining the function $F^*(t)$, function $F(x,t)$ needs to be considered only for the points $(x,t)$ of the breakpoint curves or points that correspond to $x \in V$.

The $t$-coordinates of all vertices of all partitions $P(D_e)$, $e \in E$ partition the interval $[t^-,t^+]$ into $O(n^3)$ basic subintervals. When $(x,t)$ moves along a breakpoint curve $\mathcal{B}_{e,i}$, $F(x,t)$ can be viewed as a function of only $t$ which has a single analytic expression (quotient of two polynomials) which does not change inside any basic subinterval but may be different for different basic subintervals. Considering this function over each basic subinterval as a separate function, Lopez-de-los-Mozos et al. [16] obtain overall $O(n^5)$ simple functions of $t$ (partially defined over $[t^-,t^+]$), as there are $n-1$ domains $D_e$, $e \in E$, each domain has $O(n)$ breakpoint curves, and $F(x,t)$ over each breakpoint curve is split into $O(n^3)$ pieces that correspond to the basic subintervals. Then, they obtain $F^*(t)$ as the lower envelope $L(t)$ of these $O(n^5)$ functions plus a smaller number of additional simple functions. The breakpoints of $L(t)$ are particularly important since at these values of $t$ function $F^*(t)$ (and, therefore, $F(x,t) - F^*(t)$) is not differentiable.

In Phase 2, they obtain a finer partition $P'(D_e)$ of each domain $D_e$, $e \in E$ by crossing the cells of the original partition $P(D_e)$ by the lines $t = t'$ for all breakpoints $t'$ of the lower envelope $L(t)$. The significance of this partition is that in the interior of each cell of $P'(D_e)$, $F(x,t) - F^*(t)$ is a differentiable function, hence for a fixed $x \in e$, if the maximum of $F(x,t) - F^*(t)$ over $t \in (t^-,t^+)$ is in the interior of a cell of $P'(D_e)$, then it is on the stationary curve defined by the equation $\frac{\partial(F(x,t)-F^*(t))}{\partial t} = 0$. They evaluate the cardinality of the partition $P'(D_e)$ as $O(n^2)\cdot$(the number of breakpoints of $L(t)$) since the initial partition $P(D_e)$ has cardinality $O(n^2)$. When $(x,t)$ moves along a stationary curve of a cell of $P'(D_e)$, $F(x,t) - F^*(t)$ can be expressed as a function of $x$ only. Then, for any edge $e$, $\max_{t\in[t^-,t^+]}(F(x,t) - F^*(t))$ as a function of $x \in e$ is the upper envelope of all these functions plus a smaller number of additional simple functions. Obtaining this upper envelope is Phase 3 of the algorithm. Then, finding a minimum point of the upper envelope is straightforward.

Now let us outline our improvements. Our first improvement is based on the observation that when $(x,t)$ moves along a breakpoint curve $\mathcal{B}_{e,i}$, the analytic expression of the value $F(x,t)$ as a function of $t$ can change only when $(x,t)$ crosses another breakpoint curve $\mathcal{B}_{e,i}$ that corresponds to the same $e$. Thus, instead of considering this function over each basic subinterval as a separate function and thus breaking it into $O(n^3)$ pieces as done in [16], we can break it into only $O(n)$ pieces that correspond to the parts between the intersections with other breakpoint curves of the same edge $e$, and then each of these $O(n)$ pieces will still be a simple function and have a single analytic expression. This allows us to represent $F^*(t)$ as the lower envelope of only $O(n^3)$ simple functions instead of $O(n^5)$ functions, thus improving the overall complexity by approximately a factor of $n^2$, since this also reduces the number of breakpoints of the lower envelope and consequently the size of the partition $P'(D_e)$.

Our second improvement is based on analysis of the total cardinality of the partitions $P(D_e)$, $e \in E$. Although for a specific edge $e$, the partition $P(D_e)$ can have $O(n^2)$ cells, edges and vertices, using a more careful analysis based on specifics of the tree metric we show that all $n-1$ partitions $P(D_e)$, $e \in E$ altogether have $O(n^2)$ cells, edges and vertices. This results in representing $F^*(t)$ as the lower envelope of only $O(n^2)$ simple functions, further improving the overall complexity by approximately a factor of $n$.

Our third improvement is based on careful analysis of the cardinalities of partitions $P'(D_e)$, $e \in E$. Lopez-de-los-Mozos et al. [16] evaluate the cardinality of $P'(D_e)$ for a specific $e \in E$ as $O(n^2)\cdot$(the number of breakpoints of $L(t)$). We prove that the cardinality of $P'(D_e)$ is $O(n)\cdot$(the number of breakpoints of $L(t)$), which translates into a further improvement of the overall complexity by a factor of $n$.

Our fourth improvement is applicable if $N$ is a path. In this case, we prove that although $O(n)\cdot$(the number of breakpoints of $L(t)$) is an upper bound on the number of cells of the partition $P'(D_e)$ for a specific $e \in E$, it is also an upper bound on the total number of cells in all $n-1$ partitions $P'(D_e)$, $e \in E$. This leads to a reduction of the overall complexity by a factor of $n$. The result is based on analysis of properties of relevant functions under the path metric.

Therefore, for a tree, we have an improvement approximately by a factor of $n^4$, and for a path approximately by a factor of $n^5$ over the complexity estimates of [16].

## 3.2 The general network case.

The algorithm from [16] for a general network is essentially the same as the algorithm for the tree, but with primary regions of edges considered separately instead of the edges. Interestingly, although our second and fourth improvements above are heavily based on properties of the tree and path metrics, respectively, the ideas of all four improvements (in modified forms) are applicable for general networks. One reason for this is that primary regions of an edge of a general network form a structure somewhat similar to a path. Another reason is that although the involved distance-based functions considered over an edge of a general network have very different properties than on a path or a tree (for example, $d(v_i, x)$ and $M(x,t)$ for a fixed $t$ are concave piece-wise linear on an edge of a general network but are convex piece-wise linear on a path), these properties can also be used to ensure applicability of the main ideas of the improvements. In fact, on general networks our complexity improvement is the strongest: we improve the complexity bound approximately by a factor of $mn^5$ with respect to the bound obtained using the analysis in [16].

# 4 A compressed description of the algorithm from [16]

In this section, we provide a more detailed description of the algorithm from [16] which is necessary for presenting our improved analysis. Some of the arguments from the previous section will have to be repeated (in extended form), but this is needed for completeness of presentation. Following [16], we describe the algorithm for a tree and then comment on extending it to a general network. So, suppose that $N(V,E)$ is a tree. Thus, $|E| = n-1$.

## 4.1 The breakpoint curves and initial domain partition.

For an edge $e = [u, v]$ with endpoint nodes $u$ and $v$, let any real number $x_e \in [0, l_e]$ denote the point of $e$ such that the subedge $[u, x_e]$ has length $x_e$. Functions $d(v_i, x_e)$, $i \in [1:n]$ are linear on $e$. Function $F(x_e, t)$ in the interior of the domain $D_e = [0, l_e] \times [t^-, t^+]$ is differentiable everywhere except for the points of the breakpoint curves, where a breakpoint curve $\mathcal{B}_{e,i}$ is the set of points of $D_e$ that satisfy $d(v_i, x_e) - M(x_e, t) = 0$ for some $i \in [1:n]$. Since $d(v_i, x_e) = d(v_i, u) + \delta_i x_e$ where $\delta_i \in \{-1, 1\}$, each breakpoint curve $\mathcal{B}_{e,i}$ can be defined by an equation of the form $f_{e,i}(x_e, t) = 0$ where $f_{e,i}(x_e, t) = W(t)(d(v_i, x_e) - M(x_e, t)) = a_{e,i} x_e t + b_{e,i} t + c_{e,i} x + d_{e,i}$ for some $a_{e,i}, b_{e,i}, c_{e,i}, d_{e,i} \in \mathbb{R}$ [16], so the breakpoint curves are hyperbolas or straight lines. It is mentioned in [16] that two different breakpoint curves can intersect in at most two points. In fact, this statement can be strengthened: two different breakpoint curves $\mathcal{B}_{e,i}$ and $\mathcal{B}_{e,j}$ can intersect in at most one point since at the intersection point $(x_e, t)$ the equality $d(v_i, x_e) =$

$d(v_j, x_e)$ must hold, but this difference is not essential. The $n$ breakpoint curves in the domain $D_e$ define a partition $P(D_e)$ of the domain into $O(n^2)$ full-dimensional cells $R_j$, $j \in [1 : r_e]$; the function $F(x_e, t)$ is differentiable in the interior of each cell of this partition. The set $V(P(D_e))$ of vertices of $P(D_e)$ has cardinality $O(n^2)$. Computing the arrangement of the $n$ breakpoint curves can be done in $O(n^2\alpha(n))$ time and $O(n^2)$ space [10].

### 4.2 Phase 1: Obtaining $F^*(t)$.

For fixed $t \in [t^-, t^+]$ and $e \in E$, function $F(x_e, t)$ is piece-wise linear and convex on $[0, l_e]$ as a function of $x_e$, with breakpoints that belong to the breakpoint curves. The minimum of $F(x, t)$ over all $x \in N$ for any fixed $t \in [t^-, t^+]$ is attained at some breakpoint curve in $D_e$ or at one of the boundary segments $x_e = 0$, $x_e = l_e$ of $D_e$ for some $e \in E$ [16].

To obtain differentiable analytic expressions for relevant functions, Lopez-de-los-Mozos et al. [16] define the list $\Sigma = \{t_1, ..., t_\sigma\}$ of $t$-coordinates of all vertices of the initial partitions $P(D_e)$, $e \in E$, sorted in non-decreasing order; thus, $t_1 = t^-$, $t_\sigma = t^+$, $t_{k+1} \geq t_k$ for any $k = 1, ..., \sigma - 1$, $\sigma = |\Sigma| = O(n^3)$. Then, they consider *basic subintervals* $[t_k, t_{k+1}]$, $k \in [1 : \sigma - 1]$. When $(x_e, t) \in e \times [t_k, t_{k+1}]$ moves along a breakpoint curve $f_{e,i}(x_e, t) = a_{e,i}x_e t + b_{e,i}x_e + c_{e,i}t + d_{e,i} = 0$, $x_e$ can be expressed as a function of $t$, $x_e = \hat{f}_{e,i}(t)$, unless $a_{e,i} = b_{e,i} = 0$. They define $\hat{I}_{e,i}$ as the projection of the whole breakpoint curve $f_{e,i}(x_e, t) = a_{e,i}x_e t + b_{e,i}x_e + c_{e,i}t + d_{e,i} = 0$, $(x_e, t) \in D_e$ on the $t$-edge of $D_e$(if $a_{e,i} = b_{e,i} = 0$, this projection is a single point $t^{e,i} = -d_{e,i}/c_{e,i}$; in this special case, they define $\hat{f}_{e,i}(t) = x^{e,i}$ where $x^{e,i}$ is the minimizer of $F(x_e, t)$ over $x_e \in [0, l_e]$ for a fixed $t = t^{e,i}$). Each function $F(\hat{f}_{e,i}(t), t)$, $t \in \hat{I}_{e,i}$, has a single analytic expression as the factor $(W(t))^{-2}$ multiplied by the quotient of two polynomials with maximum degrees bounded by 3 and 1, respectively, within any basic subinterval $[t_k, t_{k+1}]$ in $\hat{I}_{e,i}$ (but this expression can be different for different basic subintervals); therefore, they consider these functions defined on different basic subintervals $[t_k, t_{k+1}]$ separately (thereby each function $F(\hat{f}_{e,i}(t), t)$, $t \in \hat{I}_{e,i}$ is split into at most $|\Sigma| - 1 = \sigma - 1$ consecutive pieces), and point out that the function $F^*(t)$ is obtained from the lower envelope $L(t)$ of the collection of $(\sigma - 1)(n + 2)(n - 1) = O(n^5)$ functions

$$\{\{F(\hat{f}_{e,i}(t), t),\ t \in \hat{I}_{e,i},\ i \in [1 : n],\ F(0, t),\ F(l_e, t)\},\ e \in E,\ t \in [t_k, t_{k+1}],\ k \in [1 : \sigma - 1]\}. \quad (1)$$

These functions are partially defined on $[t^-, t^+]$, have differentiable analytic expressions, and the number of intersection points between any two of them is at most four [16]. Then, invoking standard computational geometry results [22], the lower envelope $L(t)$ can be computed in $O(\lambda_5(n^5) \log n)$ time and consists of $O(\lambda_6(n^5))$ consecutive differentiable arcs.

### 4.3 Phases 2 and 3: Solving MMR-MAD.

Let $\tau = \{t'_1, ..., t'_g\}$ be the list of consecutive breakpoints of the lower envelope $L(t)$. Lopez-de-los-Mozos et al. [16] consider a new (finer) partition $P'(D_e)$ of the domain $D_e$ which is obtained when the cells $\{R_j,\ j \in [1 : r_e]\}$ of the initial partition $P(D_e)$ are crossed by the set of lines $t = t'_k,\ t'_k \in \tau$. Obtaining the partition $P'(D_e)$ they consider as **Phase 2** of the algorithm. Let $R'_s,\ s \in [1 : r'_e]$ be the cells of $P'(D_e)$. Then, in the interior of each cell $R'_s$ of $P'(D_e)$, $F(x,t) - F^*(t)$ is a differentiable function. It is shown in [16] that for a fixed $x_e \in [0, l_e]$, the maximum of $F(x_e, t) - F^*(t)$ over $t \in (t^-, t^+)$ cannot be on a breakpoint curve, and, if it is in the interior of a cell $R'_s$ of $P'(D_e)$, it must satisfy

$$H_s(x_e, t) = \frac{\partial(F(x_e, t) - F^*(t))}{\partial t} = 0 \tag{2}$$

They showed that this equation can be represented as a bivariate polynomial equation in $x, t$ of maximum degree 3 in $t$ and 1 in $x$. Thus, when $(x_e, t)$ moves along the stationary curve $H_s(x_e, t) = 0$, isolating (if possible) $t$ from (2) can be done in closed form, $t = h_s(x_e)$, and values of $F(x_e, t) - F^*(t)$ over the stationary curve can be expressed as $F(x_e, h_s(x_e)) - F^*(h_s(x_e))$ for $x \in J_s$ where $J_s$ is the projection of $H_s(x_e, t) = 0$ on the $x$-edge. If it is not possible to isolate $t$ from (2), then the stationary curve is described as $x_e = x^s_e$ for a constant $x^s_e$, and $J_s = \{x^s_e\}$. In this special case, for consistency, they define $h_s(x_e) = t^s$ where $t^s$ is the value where $\max_t\{F(x^s_e, t) - F^*(t) :\ (x^s_e, t) \in R'_s\}$ is attained.

Lopez-de-los-Mozos et al. [16] state that the partition $P'(D_e)$ has $r'_e = O(n^2\lambda_6(n^5))$ cells since the initial partition $P(D_e)$ has $O(n^2)$ cells and $|\tau| = g = O(\lambda_6(n^5))$, and point out that $\max_{t\in[t^-,t^+]}\{F(x_e, t) - F^*(t)\}$ as a function of $x_e \in [0, l_e]$ is the upper envelope $U'_e(x_e)$ of the collection of functions

$$\{F(x_e, h_s(x_e)) - F^*(h_s(x_e)),\ x_e \in J_s,\ s \in [1 : r'_e], \quad F(x_e, t'_k) - F^*(t'_k),\ k \in [1 : |\tau|]\}\,. \tag{3}$$

Obtaining the upper envelope $U'_e(x)$ they consider as **Phase 3** of the algorithm. Since $r'_e = O(n^2\lambda_6(n^5))$, and the number of intersection points of any two functions from the collection (3) is bounded from above by a constant that does not depend on $n$, they obtain complexity $O(n^2\lambda_6(n^5)\log^* n \log n)$ for constructing $U'_e(x)$ and solving MMR-MAD restricted to a single edge $e$, which results in $O(n^3\lambda_6(n^5)\log^* n \log n)$ complexity for solving MMR-MAD over the whole tree. We note that since $\lambda_6(n^5)$ grows (slightly) faster than $n^5$, the complexity is (slightly) higher than $O(n^8 \log n)$.

### 4.4 MMR-MAD on a general network.

For a general network N, a point $x_e$ of an edge $e = [u, v]$ is called a *bottleneck point* with respect to $v_i \in V$, if the distance from $x_e$ to $v_i$ is the same via vertex $u$ as via vertex $v$.

The bottleneck points partition the edge into $O(n)$ *primary regions* over which the distance function $d(x_e, v_j)$ is linear for any $v_j \in V$, and thus a primary region of an edge of the general network is similar to an edge of a tree. Lopez-de-los-Mozos et al. [16] point out that MMR-MAD restricted to a primary region in a general network can be solved and analyzed using the same procedure as MMR-MAD restricted to an edge of a tree. They claim that this results in $O(mn^3\lambda_6(n^5)\log^* n \log n)$ complexity for solving MMR-MAD on the whole network since there are $O(mn)$ primary regions. However, there is an error in this argumentation: apparently they did not take into consideration that applying the same logic in the case of a general network would result in the estimate $O(mn^3)$ for the cardinality of the list $\Sigma$ in Phase 1 (instead of $O(n^3)$ for the tree case), because there are $O(mn)$ primary regions that contribute points to this list instead of $O(n)$ edges of a tree. Also, instead of $n-1$ edges $e \in E$ in the case of a tree, for a general network we would have $O(mn)$ primary regions in (1). Following the logic of [16], function $F(x,t)$ over each breakpoint curve is split into $O(|\Sigma|) = O(mn^3)$ consecutive pieces, there are $O(n)$ breakpoint curves in each primary region and $O(mn)$ primary regions. Therefore, a direct extension of the analysis of [16] from the tree case to the general network case would result in the bound $O(m^2n^5)$ for the cardinality of the collection (1) in the case of a general network and in the bound $O(\lambda_6(m^2n^5))$ for the number of arcs of the lower envelope $L(t)$. In turn, the estimate of the cardinality of the list $\tau$ in Phase 2 would be $O(\lambda_6(m^2n^5))$ instead of $O(\lambda_6(n^5))$, which, following the logic of [16], would result in the bound $O(mn^3\lambda_6(m^2n^5)\log^* n \log n)$ for the complexity of their algorithm for a general network (instead of $O(mn^3\lambda_6(n^5)\log^* n \log n)$ stated in [16]), which is (slightly) higher than $O(m^3n^8 \log n)$.

# 5 Improvements for the tree case

In this section, we assume that $N$ is a tree, and present our improvements for this case.

## 5.1 Improvement 1.

In this subsection, we present an algorithmic improvement. In the algorithm of [16], in Phase 1, $F^*(t)$ is obtained as the lower envelope $L(t)$ of the collection (1) of $O(n^5)$ simple functions partially defined on $[t^-, t^+]$. Thus, as argued in [16], $L(t)$ has $O(\lambda_6(n^5))$ differentiable arcs and breakpoints. Here we show that $F^*(t)$ can be obtained as the lower envelope of a much smaller collection of simple functions.

To avoid ambiguity, if breakpoint curves $\mathcal{B}_{e,i}$ and $\mathcal{B}_{e,j}$ are identical, we consider that they do not have intersection points.

Consider a breakpoint curve $\mathcal{B}_{e,i}$ defined by $f_{e,i}(x_e,t) = a_{e,i}x_et + b_{e,i}x_e + c_{e,i}t + d_{e,i} = 0$, $(x_e,t) \in [0,l_e] \times [t^-,t^+] = D_e$, and its projection $\hat{I}_{e,i}$ on the $t$-edge of $D_e$, and suppose that $\hat{I}_{e,i}$ is not a single point. Then, as discussed in Subsection 4.2, when $(x_e,t)$ moves along the breakpoint curve, $x_e$ can be expressed as a function of $t$, $x_e = \hat{f}_{e,i}(t)$. Let $t_1^{e,i}, t_2^{e,i}, ..., t_{q_{e,i}}^{e,i}$ be the sorted distinct $t$-coordinates of the intersection points of $\mathcal{B}_{e,i}$ with the other breakpoint curves $\mathcal{B}_{e,j}$ of edge $e$ or with the boundary of $D_e$, $t_1^{e,i} < t_2^{e,i} < ... < t_{q_{e,i}}^{e,i}$; $q_{e,i} = O(n)$. Observe that function $F(\hat{f}_{e,i}(t),t)$ on $\hat{I}_{e,i}$ is differentiable everywhere except the points $t_k^{e,i}$, $k \in [1 : q_{e,i}]$; only at these points the analytic expression for $F(\hat{f}_{e,i}(t),t)$ may change. Consequently, to break $F(\hat{f}_{e,i}(t),t)$ into pieces with single differentiable analytic expressions (simple functions), it is sufficient to break it into $O(n)$ pieces defined by the points $t_k^{e,i}$, $k \in [1 : q_{e,i}]$, instead of $O(n^3)$ pieces defined by the list $\Sigma$ as done in [16].

If $\hat{I}_{e,i}$ is a single point $t^{e,i}$, we define $\hat{f}_{e,i}(t)$ as done in [16] for this case, i.e., $\hat{f}_{e,i}(t) = x^{e,i}$ where $x^{e,i}$ is a minimizer of $F(x_e,t)$ over $x_e \in [0,l_e]$ for a fixed $t = t^{e,i}$, and for consistency of notation define $q_{e,i} = 2$, $t_1^{e,i} = t_2^{e,i} = t^{e,i}$.

We can treat the boundary segments $x_e = 0$ and $x_e = l_e$ of $D_e$ similarly to the breakpoint curves, formally considering them as breakpoint curves with indices 0 and $n+1$. Each of the functions $F(0,t)$ and $F(l_e,t)$, $t \in (t^-,t^+)$ is differentiable everywhere except the $t$-coordinates of the $O(n)$ points where the breakpoint curves $\mathcal{B}_{e,i}$, $i \in [1 : n]$ intersect the corresponding segment of the boundary of $D_e$. Let us define $\hat{f}_{e,0}(t) = 0$, $\hat{f}_{e,n+1}(t) = l_e$, and let $t_1^{e,0}, t_2^{e,0}, ..., t_{q_{e,0}}^{e,0}$ be the sorted distinct $t$-coordinates of the intersection points of the breakpoint curves $\mathcal{B}_{e,i}$, $i \in [1 : n]$, with the segment $x_e = 0$ of the boundary of $D_e$ plus the values $t^-, t^+$, i.e., $t^- = t_1^{e,0} < t_2^{e,0} < ... < t_{q_{e,0}}^{e,0} = t^+$. Let $t_1^{e,n+1}, t_2^{e,n+1}, ..., t_{q_{e,n+1}}^{e,n+1}$ be the values defined similarly for the segment $x_e = l_e$ of $D_e$. Define $\hat{I}_{e,0} = \hat{I}_{e,n+1} = [t^-,t^+]$.

**Observation 1**. The function $F^*(t)$ is obtained from the lower envelope $L'(t)$ of the collection of functions

$$\left\{F(\hat{f}_{e,i}(t),t),\ t \in \hat{I}_{e,i} \cap [t_k^{e,i}, t_{k+1}^{e,i}],\ k \in [1 : q_{e,i}-1],\ i \in [0 : n+1],\ e \in E\right\}. \tag{4}$$

Since $q_{e,i} = O(n)$, and $|E| = n-1$, the collection has $O(n^3)$ functions. All of the functions have differentiable analytic expressions, are partially defined on $[t^-,t^+]$, and any pair of them intersect in at most four points for the reasons discussed in [16] (see also Subsection 4.2). Consequently, $L'(t)$ has $O(\lambda_6(n^3))$ consecutive differentiable arcs and can be computed in $O(\lambda_5(n^3)\log n)$ time.

Keeping the remainder of the computations and the analysis the same as in [16], we obtain that the list $\tau$ in Phase 2 will have cardinality $O(\lambda_6(n^3))$, and the overall complexity bound of the algorithm with Improvement 1 will be $O(n^3\lambda_6(n^3)\log^* n \log n)$ instead of $O(n^3\lambda_6(n^5)\log^* n \log n)$ in [16].

## 5.2 Improvement 2.

In this subsection, we show that the collection (4) defined in the previous subsection in fact has a smaller cardinality $O(n^2)$ instead of $O(n^3)$, and therefore the lower envelope $L'(t)$ in fact has a smaller number of arcs and breakpoints, $\lambda_6(n^2)$ instead of $\lambda_6(n^3)$. To do so, instead of the "local" analysis (estimating separately the worst-case contributions of single edges and taking the sum of the worst-case contributions over all edges) we have to use a "global" analysis - estimating the overall simultaneous contribution of all edges of the tree.

For any two points $b, c$ of the tree $N$, let $P(b, c)$ denote the unique path between $b$ and $c$.

**Observation 2**. For any $i, j \in [1 : n]$, the set of points $a \in N$ that are equidistant from $v_i$ and $v_j$ (i.e., $d(a, v_i) = d(a, v_j)$) is either a single point of $N$, or is a subtree of $N$.

**Proof**. If the middlepoint of the path $P(v_i, v_j)$ is an interior point of some edge, this is the only point of $N$ equidistant from $v_i$ and $v_j$. If the middlepoint of $P(v_i, v_j)$ is a node $v_k$, then the set of points equidistant from $v_i$ and $v_j$ is the set of points $a \in N$ such that $v_k \in P(a, v_i)$, $v_k \in P(a, v_j)$ which is a subtree of $N$. $\square$

An intersection point $(x_e, t)$ between breakpoint curves $\mathcal{B}_{e,i}$ and $\mathcal{B}_{e,j}$ is called a *proper intersection point* if $x_e \neq 0$, $x_e \neq l_e$, i.e. $x_e$ is an interior point of $e$. The following result shows that for any $i, j \in [1 : n]$, at most one edge $e$ of the tree $N$ contains a proper intersection point of the breakpoint curves $\mathcal{B}_{e,i}$ and $\mathcal{B}_{e,j}$.

**Lemma 1**. For any $i, j \in [1 : n]$, there is at most one proper intersection point of the breakpoint curves $\mathcal{B}_{e,i}$ and $\mathcal{B}_{e,j}$ over all edges $e$ of the tree $N$.

**Proof**. If $(x_e, t)$ is an intersection point between $\mathcal{B}_{e,i}$ and $\mathcal{B}_{e,j}$, $x_e$ must be equidistant from $v_i$ and $v_j$. If there is only one point of $N$ that is equidistant from $v_i$ and $v_j$, the statement of the lemma is straightforward. If there is a subtree of points equidistant from $v_i$ and $v_j$, then for any edge $e$ of this subtree $\mathcal{B}_{e,i}$ and $\mathcal{B}_{e,j}$ are identical and thus do not contribute intersection points. Other edges of $N$ do not have interior points equidistant from $v_i$ and $v_j$ and thus do not contribute proper intersection points between $\mathcal{B}_{e,i}$ and $\mathcal{B}_{e,j}$. The lemma is proven. $\square$

**Theorem 1.** The collection (4) defined in the previous subsection has $O(n^2)$ functions.

**Proof**. For any $i \in [1 : n]$ and any $e \in E$, $\mathcal{B}_{e,i}$ has a constant number (no more than 4) of intersection points with the boundary of $D_e$, thus for a fixed $i \in [1 : n]$ all $\mathcal{B}_{e,i}$, $e \in E$ together have $O(n)$ intersection points with the boundaries of $D_e$, $e \in E$. Any non-proper intersection point between $\mathcal{B}_{e,i}$ and $\mathcal{B}_{e,j}$ is also an intersection point of these curves with the boundary of $D_e$. Using Lemma 1, we have that there are $O(n^2)$ proper intersection points between breakpoint curves over all $e \in E$, and $O(n^2)$ intersection points between breakpoint curves and the boundaries of the corresponding domains $D_e$. Also, we get that $\sum_{e \in E} q_{e,i} = O(n)$

for any $i \in [1:n]$, and $\sum_{e \in E} q_{e,i} = O(n^2)$ for $i = 0$ and $i = n+1$. The statement of the theorem follows. $\square$

Keeping the remainder of the computations and the analysis the same as in [16], we obtain that the list $\tau$ in Phase 2 will have cardinality $O(\lambda_6(n^2))$, and the overall complexity bound of the algorithm with Improvements 1 and 2 will be $O(n^3\lambda_6(n^2)\log^* n \log n)$ instead of $O(n^3\lambda_6(n^5)\log^* n \log n)$ in [16].

**Remark.** The discussion above implies that the initial domain partitions $P(D_e)$, $e \in E$ altogether have $O(n^2)$ vertices and edges. Since each edge of each partition is adjacent to at most two cells, this implies that the partitions $P(D_e)$, $e \in E$ altogether have $O(n^2)$ cells. This observation will be used later in Subsection 5.4.

## 5.3 Improvement 3.

Improvements 1 and 2 focused on Phase 1. Now we focus on Phase 2. Lopez-de-los-Mozos et al. [16] estimate the number of cells in the partition $P'(D_e)$ obtained in Phase 2 as the product of the estimate of the cardinality of the list $\tau$ which is $O(\lambda_6(n^5))$ in [16] and the number of cells of $P(D_e)$ which is $O(n^2)$, obtaining an estimate $O(n^2\lambda_6(n^5))$ for the number of cells in $P'(D_e)$. Using the same logic, in our case we would obtain a bound $O(n^2\lambda_6(n^2))$ for the number of cells in $P'(D_e)$, because with Improvements 1 and 2, $|\tau| = O(\lambda_6(n^2))$. However, using a different argument, we will obtain a better bound $O(n\lambda_6(n^2))$ for the number of cells in $P'(D_e)$.

Again, let $\tau = \{t'_1, ..., t'_g\}$ be the list of consecutive breakpoints of the lower envelope $L'(t)$, $|\tau| = O(\lambda_6(n^2))$ according to the discussion above. Partition $P'(D_e)$ of $D_e$ is obtained as discussed in Subsection 4.3. Observe that each line $t = t'_k$, $t'_k \in \tau$ has $O(n)$ intersection points with the breakpoint curves in $D_e$. (If the line $t = t'_k$ is itself a breakpoint curve in $D_e$, we consider that it does not have intersection points with itself, and we can ignore the line as it does not contribute new cells to $P'(D_e)$). These intersection points partition the segment $\{t = t'_k,\ x_e \in [0, l_e]\}$ of $D_e$ into $O(n)$ subsegments which we will call $\tau$-segments. Thus, there are $O(n\lambda_6(n^2))$ $\tau$-segments in $D_e$. Observe that each $\tau$-segment is an edge of at most two cells of $P'(D_e)$. Consequently, at most $O(n\lambda_6(n^2))$ cells of $P'(D_e)$ are adjacent to some $\tau$-segments. Observe that if a cell $R'_s$ of $P'(D_e)$ is not adjacent to any $\tau$-segment, then $R'_s$ is one of the cells of the original partition $P(D_e)$. Since there are $O(n^2)$ cells in $P(D_e)$, the number of cells of $P'(D_e)$ is $O(n\lambda_6(n^2)) + O(n^2) = O(n\lambda_6(n^2))$.

Keeping the remainder of the computations and the analysis the same as in [16], we obtain that the collection of functions (3) will have cardinality $O(n\lambda_6(n^2))$, and the overall complexity bound of the algorithm with Improvements 1, 2 and 3 will be $O(n^2\lambda_6(n^2)\log^* n \log n)$ instead of $O(n^3\lambda_6(n^5)\log^* n \log n)$ in [16].

## 5.4 Improvement 4: The case of a path.

In this subsection, we assume that $N$ is a path network. Since a path is a special case of a tree, Improvements 1-3 are still valid. A path can be viewed as embedded in the number line. In addition to the notation $x_e$ that represents a point of an edge $e$, we will also use notation $x$ to represent the point of the path that is $x$ units away from its leftmost point.

Observe that for any fixed $t' \in [t^-, t^+]$, function $M(x, t')$ is continuous piece-wise linear convex as a function of $x$, with breakpoints at the nodes of $N$. For any $i \in [1 : n]$, function $d(v_i, x)$ is a convex tooth-function, i.e. continuous piecewise-linear convex function with at most two linear pieces and one breakpoint at $x = v_i$. This implies the following observation.

**Observation 3**. For fixed $t' \in [t^-, t^+]$ and $i' \in [1 : n]$, the equation $d(v_{i'}, x) - M(x, t') = 0$ over $x \in N$ has at most four isolated solutions, and also may hold over one or more full edges.

Consider some $t'$ from the list $\tau$ of breakpoints of the lower envelope $L'(t)$, and an $i' \in [1 : n]$. Suppose that the equation $d(v_{i'}, x) - M(x, t') = 0$ has only isolated solutions (at most four according to Observation 3). Then, the $n-1$ segments $\{t = t',\ x_e \in [0, l_e]\}$, $e \in E$ have at most four intersections with the breakpoint curves $\mathcal{B}_{e,i'}$, $e \in E$ over all $n-1$ domains $D_e$, $e \in E$. If the equation $d(v_{i'}, x) - M(x, t') = 0$ also holds over some full edges $e' \in E$, then for each such edge $e'$ the segment $\{t = t',\ x \in [0 : l_{e'}]\}$ coincides with the breakpoint curve $\mathcal{B}_{e',i'}$ and thus can be ignored as it does not contribute additional cells to $P'(D_{e'})$. Thus, the overall number of $\tau$-segments in all domains $D_e$, $e \in E$ that correspond to a fixed $t' \in \tau$ is $O(n)$. Since $|\tau| = O(\lambda_6(n^2))$, the overall number of $\tau$-segments in all domains $D_e$, $e \in E$ is $O(n\lambda_6(n^2))$. (Note that for the tree case, $O(n\lambda_6(n^2))$ was the estimate of the number of $\tau$-segments in just one domain $D_e$.) So, the number of cells that are adjacent to $\tau$-segments in all partitions $P'(D_e)$, $e \in E$ is $O(n\lambda_6(n^2))$. The number of cells not adjacent to any $\tau$-segments in all $P'(D_e)$, $e \in E$ is $O(n^2)$ since each such cell is a cell of some initial partition $P(D_e)$, and the total number of cells in all $P(D_e)$, $e \in E$ is $O(n^2)$ as derived in the discussion of Improvement 2 (see the remark at the end of Subsection 5.2). Consequently, the total number of cells in all $P'(D_e)$, $e \in E$ is $O(n\lambda_6(n^2))$. Let $C_e$ be the cardinality of the collection of functions (3) taking into account all the improvements; we obtain $\sum_{e \in E} C_e = O(n\lambda_6(n^2))$. Therefore, the overall complexity bound of the algorithm reduces to $O(n\lambda_6(n^2) \log^* n \log n)$ for the path case.

The following theorem summarizes the results of this section.

**Theorem 2.** a) MMR-MAD on a tree can be solved in $O(n^2\lambda_6(n^2) \log^* n \log n) = O(n^4(\log^* n)^2 \log n)$ time.
b) MMR-MAD on a path can be solved in $O(n\lambda_6(n^2) \log^* n \log n) = O(n^3(\log^* n)^2 \log n)$ time.

This is a significant improvement over the complexity $O(n^3\lambda_6(n^5) \log^* n \log n) = O(n^8(\log^* n)^2 \log n)$ obtained in [16].

# 6 Improvements for the case of a general network

In this section, we assume that $N$ is a general network. Thus, instead of domains $D_e$ that correspond to edges in the tree case, we deal with domains $D_{e,k}$ that correspond to primary regions, with $k$ being the index of the primary region on the edge $e$.

Improvement 1 is applicable without any changes, using primary regions of the general network instead of edges of a tree. Instead of breaking function $F(x,t)$ over a breakpoint curve of a primary region into $O(|\Sigma|) = O(mn^3)$ pieces, we break it into $O(n)$ pieces defined by the intersection points of this breakpoint curve with other breakpoint curves of the same primary region and the boundary of the corresponding domain. Each piece has a single differentiable expression as a function of $t$. This results in $F^*(t)$ being represented as the lower envelope of a collection of $O(mn^3)$ functions (there are $O(mn)$ primary regions, $O(n)$ breakpoint curves for each primary region, and $F(x,t)$ over each breakpoint curve is split into $O(n)$ consecutive pieces), which leads to the estimate $O(mn^3\lambda_6(mn^3)\log^* n \log n)$ for the complexity of the overall algorithm for MMR-MAD. To emphasize that we are considering the general network case, we will call this *Improvement 1G*.

The idea of Improvement 2 is applicable in a modified form.

**Observation 4.** For any two nodes $v_i$, $v_j$ and an edge $e \in E$, the equation $d(v_i,x) = d(v_j,x)$ has at most one isolated solution on the edge $e$, and also may hold on one or more full primary regions of $e$; for such primary regions, the breakpoint curves that correspond to $v_i$ and $v_j$ coincide if exist.

Observation 4 implies that there is at most one intersection point of the breakpoint curves that correspond to $v_i$ and $v_j$ over all primary regions of $e$ and the corresponding domains $D_{e,k}$. So, the overall number of intersection points of breakpoint curves over all primary regions and the corresponding domains of an edge $e$ is $O(n^2)$. The overall number of intersection points between the breakpoint curves and the boundaries of all domains $D_{e,k}$ of an edge $e$ is also $O(n^2)$. This will result in $F^*(t)$ being represented as the lower envelope of $O(mn^2)$ functions, which leads to the estimate $O(mn^3\lambda_6(mn^2)\log^* n \log n)$ for the complexity of the overall algorithm for MMR-MAD. We will call this *Improvement 2G*.

Improvement 3 is applicable straightforwardly with the same logic, with domains $D_{e,k}$ instead of domains $D_e$, and will be called *Improvement 3G*. It reduces the overall complexity bound by a factor of $n$ to $O(mn^2\lambda_6(mn^2)\log^* n \log n)$.

An analog of Improvement 4, which will be called *Improvement 4G*, is applicable for every edge, because the primary regions of an edge are similar to edges of a path. However, there are some differences. Function $M(x,t')$ over an edge $e$ for a fixed $t' \in [t^-, t^+]$ is a continuous *concave* piecewise-linear function of $x$ (instead of a convex function $M(x,t)$ on a path) with breakpoints at the endpoints of primary regions; $d(v_i, x)$ over an edge $e$ for any $i \in [1:n]$ is a *concave* continuous piecewise-linear function (instead of a *convex* function $d(v_i, x)$ on a path) with at most two linear pieces and a breakpoint at an endpoint of a primary region; so we have the following analog of Observation 3.

**Observation 5**. For fixed $t' \in [t^-, t^+]$ and $i' \in [1:n]$, for any edge $e \in E$, the equation $d(v_{i'}, x) - M(x, t') = 0$ over $x \in e$ has at most four isolated solutions, and also may hold over one or more full primary regions of $e$.

All subsequent arguments are similar, with primary regions of an edge $e \in E$ and the corresponding domains $D_{e,k}$ instead of edges $e \in E$ and the corresponding domains $D_e$ for a path. Improvement 4G reduces the overall complexity bound by a factor of $n$. So, we obtain the following theorem that summarizes the results of this section.

**Theorem 3**. MMR-MAD on a general network can be solved in $O(mn\lambda_6(mn^2)\log^* n \log n) = O(m^2n^3(\log^* n)^2 \log n)$ time.

This is a significant improvement over the complexity bound $O(mn^3\lambda_6(m^2n^5)\log^* n \log n) = O(m^3n^8(\log^* n)^2 \log n)$ obtained using the analysis in [16].

# 7 Conclusion

In this paper, for the minmax regret mean absolute deviation single facility location problem MMR-MAD, we provided a further analysis of the polynomial time solution framework developed in [16]. We presented one algorithmic improvement (Improvement 1 or 1G) and three analytical improvements (Improvements 2-4 or 2G-4G) which significantly improve the computational complexity bounds for the algorithm. Specifically, for trees our complexity bound is $O(n^2\lambda_6(n^2)\log^* n \log n) = O(n^4(\log^* n)^2 \log n)$ as opposed to the bound $O(n^3\lambda_6(n^5)\log^* n \log n) = O(n^8(\log^* n)^2 \log n)$ obtained in [16]. For paths, our bound is further reduced to $O(n\lambda_6(n^2)\log^* n \log n) = O(n^3(\log^* n)^2 \log n)$. For general networks, our complexity bound is $O(mn\lambda_6(mn^2)\log^* n \log n) = O(m^2n^3(\log^* n)^2 \log n)$, as opposed to the bound $O(mn^3\lambda_6(m^2n^5)\log^* n \log n) = O(m^3n^8(\log^* n)^2 \log n)$ that follows from the analysis in [16].

We note that our improvements do not change in a major way the algorithmic framework of [16]; rather, they represent a deeper analysis of that framework. The complexity bounds obtained in [16] are polynomial but way too high to be of any practical use. Our complexity bounds are much lower and can be viewed as a justification that the algorithmic framework suggested in [16] has a potential for practice.

**ACKNOWLEDGEMENT**. The research of Igor Averbakh was supported by the Discovery Grant 238234-2012-RGPIN from the Natural Sciences and Engineering Research Council of Canada (NSERC). The research of Oded Berman was supported by the Discovery Grant 10547-2011-RGPIN from NSERC. The research of Marina Leal was supported by the Spanish Ministry of Science and Technology project MTM2013-46962-C2-1-P.

# References

[1] H.Aissi, C.Bazgan, and D.Vanderpooten. Minmax and minmax regret versions of combinatorial optimization problems: A survey. *European Journal of Operational Research* 197 (2009), 427-438.

[2] I.Averbakh, Complexity of robust single facility location problems on networks with uncertain edge lengths. *Discrete Applied Mathematics* 127 (2003), 505-522.

[3] I.Averbakh. Minmax regret solutions for minimax optimization problems with uncertainty. *Operations Research letters* 27 (2000), 57-65.

[4] I.Averbakh and O.Berman. Minmax regret median location on a network under uncertainty. *INFORMS Journal on Computing* 12 (2000), 104-110.

[5] O.Berman and E.H.Kaplan. Equity maximizing facility location schemes. *Transportation Science* 24 (1990),137-144.

[6] R.E.Burkard and H.Dollani. Robust location problems with Pos/Neg weights on a tree. *Networks* 38(2001), 102-113.

[7] R.E.Burkard and H.Dollani. A note on the robust 1-center problem on trees. *Annals of Operations Research* 110 (2002), 69-82.

[8] E.Conde. Minmax regret location-allocation problem on a network under uncertainty. *European Journal of Operational Research* 179 (2007), 1025-1039.

[9] E.Conde. A note on the minmax regret centdian location on trees. *Operations Research Letters* 36 (2008), 271-275.

[10] H.Edelsbrunner, L.Guibas, J.Pach, R.Pollack, R.Seidel, and M.Sharir. Arrangements of curves in the plane - topology, combinatorics, and algorithms. *Theoretical Computer Science* 92 (1992), 319-336.

[11] E.Erkut. Inequality measures for location problems. *Location Science* 1 (1993), 199-217.

[12] A.Kasperski. *Discrete optimization with interval data: minmax regret and fuzzy approach.* Berlin, Springer-Verlag, 2008.

[13] A.Kasperski and P.Zielinski. An approximation algorithm for interval data minmax regret combinatorial optimization problems. *Information Processing Letters* 97 (5) (2006), 177-180.

[14] A.Kasperski and P.Zielinski. Robust discrete optimization under discrete and interval uncertainty: a survey. In: Robustness Analysis in Decision Aiding, Optimization, and Analytics. *International Series in Operations Research and Management*, Volume 241, pp. 113-143. Springer, 2016.

[15] P.Kouvelis and G.Yu. *Robust discrete optimization and its applications.* Kluwer, Boston, 1997.

[16] M.C.Lopez-de-los-Mozos, J.Puerto, and A.M.Rodriguez-Chia. Robust mean absolute deviation problems on networks with linear vertex weights. *Networks*, 61 (1) (2013), 76-85.

[17] M.T.Marsh and D.A.Schilling. Equity measurement in facility location analysis: A review and framework. *European Journal of Operational Research* 74 (1994), 1-17.

[18] J.A.Mesa, J.Puerto, and A.Tamir. Improved algorithms for several location problems with equity measures. *Discrete Applied Mathematics*, 130 (2003), 437-448.

[19] R.Montemanni. A Benders decomposition approach for the robust spanning tree problem with interval data. *European Journal of Operational Research* 174 (3) (2006), 1479-1490.

[20] R. Montemanni, J. Barta, M. Mastrolilli, L. M. Gambardella, The robust traveling salesman problem with interval data. *Transportation Science* 41 (2007), 366-381.

[21] J.Pach and M.Sharir. Computational geometry and its algorithmic applications. *The Alcala Lectures.* AMS: Mathematics Survey Monographs 152, 2009.

[22] M.Sharir and P.K.Agarwal. *Davenport-Schinzel sequences and their geometric applications.* Cambridge University Press, New York, NY, 1995.

[23] E.Szemeredi. On a problem of Davenport and Schinzel. *Acta Arithmetica* 25 (1974), 213-224.

[24] H. Yaman, O.E. Karasan, M.C. Pinar, The robust spanning tree problem with interval data, *Operations Research Letters* 29 (2001) 31-40.