Theoretical Computer Science 627 (2016) 13–25

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Optimizing budget allocation for center and median points ☆

Boaz Ben-Moshe [a,*], Michael Elkin [b,1], Lee-Ad Gottlieb [a], Eran Omri [a]

[a] Department of Computer Science, Ariel University, Ariel, Israel
[b] Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva, Israel

ARTICLE INFO

Article history:
Received 9 August 2015
Received in revised form 3 February 2016
Accepted 12 February 2016
Available online 17 February 2016
Communicated by A. Marchetti-Spaccamela

Keywords:
Facility location
Graph radius
Budget graphs
Center point
Graph optimization

ABSTRACT

In typical graph minimization problems, we consider a graph G with fixed weights on the edges of G. The goal is then to find an optimal vertex or set of vertices with respect to some objective function, for example. We introduce a new framework for graph minimization problems, where the weights on the graph edges are not fixed, but rather must be assigned, and the weight is inversely proportional to the cost paid. The goal is to find a valid assignment for which the resulting weighted graph optimizes the objective function.

We present algorithms for finding the optimal *budget allocation* for the center point problem and for the median point problem on trees. Our algorithms run in linear time, both for the case where a candidate vertex is given as part of the input, and for the case where finding a vertex that optimizes the solution is part of the problem. We also present a hardness result for the center point problem on complete metric graphs, followed by an $O(\log^2(n))$ approximation algorithm in this setting.

1. Introduction

A typical graph minimization problem has the following structure: The input includes the edge weights and an objective function, and the goal is to find a vertex or set of vertices minimizing the objective function. One important problem in this class is the *center point* problem, in which the goal is to find one vertex that minimizes the maximum distance between that vertex and all other vertices in the graph. Henceforth, we refer to this distance as *graph radius*. Another important problem in this class is the *graph median* problem, where the goal is to find the vertex that minimizes the average distance (i.e., the sum of the distances) between it and all other vertices in the graph. In this paper we consider a new model for graph minimization problems, for which both problems are addressed.

1.1. The new model

This paper suggests a new model for budget allocation problems on weighted graphs. The new model addresses the problem of allocating a fixed budget onto the graph edges where the goal is to find a subgraph that optimizes some objective function (e.g., minimizing the graph radius). Problems such as center point and median point on trees and graphs

☆ A preliminary version of this paper appeared under the title "Optimizing budget allocation in graphs" [1].

* Corresponding author.
E-mail addresses: benmo@g.ariel.ac.il (B. Ben-Moshe), elkinm@cs.bgu.ac.il (M. Elkin), leead@ariel.ac.il (L. Gottlieb), omrier@ariel.ac.il (E. Omri).

[1] This author's research has been supported by the Binational Science Foundation, grant No. 2008390.

http://dx.doi.org/10.1016/j.tcs.2016.02.013
0304-3975/

have been studied extensively [8,13,17,19]. Yet in most cases the input for such problems consists of a given (fixed) graph. Motivated by well-known budget optimization problems [2,5,10,20,9] raised in the context of communication networks, we consider the graph to be a communication graph, where the weight of each edge (link) corresponds to the delay time of transferring a (fixed length) message over the link. We suggest a *Quality of Service* model for which the weight of each edge in the graph depends inversely on the budget assigned to it. In other words, paying more for a communication link decreases its delay time.

More formally, we consider an undirected graph $G = \langle V, E\rangle$ equipped with a length function $\ell(\cdot)$ on the edges. The weighted graph (G, ℓ) induces a metric space, i.e. the length function $\ell(\cdot)$ satisfies the triangle inequality. Let B be a positive value, which we call the *budget parameter*. In our model, allocating a budget $\mathcal{B}(e) \geq 0$ to an edge $e \in E$ of length $\ell(e)$ means that the *weight* of e will be $w(e) = w_{\mathcal{B}}(e) = \frac{\ell(e)}{\mathcal{B}(e)}$ (in case $\mathcal{B}(e) = 0$, we let $w(e) = +\infty$).

The distance between two vertices v and u in $(G, \mathcal{B})$ is the sum of weights on the edges of a minimal weighted path between them, denoted as $\delta_{\mathcal{B}}(v, u)$. The *radius* of (G, w) with respect to a designated vertex r ($\mathrm{Rad}_r(G, w)$) is the maximum distance (with respect to the underlying allocation $\mathcal{B}(\cdot)$) between r and any other vertex v in the graph. In the *rooted budget radius* problem, we are given a graph $G = \langle V, E\rangle$, equipped with a length function $\ell(\cdot)$, a budget parameter B and a designated vertex r. The goal is to assign non-negative values $\mathcal{B}(e)$ to the edges of G that sum up to B (i.e., $\Sigma_{e\in E}(\mathcal{B}(e)) = B$), so that the rooted radius of the graph (G, w) with $w = \frac{\ell}{\mathcal{B}}$ with respect to the vertex r is minimized.

In the *unrooted budget radius* problem, we are given a graph $G = \langle V, E\rangle$, equipped with a length function $\ell(\cdot)$, and a budget parameter B. The goal is to minimize the rooted budget radius problem over all choices of $r \in V$. That is, our goal is to find a vertex r and a corresponding assignment $\mathcal{B}(e)$ that minimizes the budget radius problem with respect to G, B, and r. The *budget diameter* problem is defined analogously, where the goal is to minimize the maximum distance between any pair of vertices in (G, ω).

In the *budget median* problem, we are given a graph $G = \langle V, E\rangle$, equipped with a length function $\ell(\cdot)$, and a budget parameter B. The goal is to find a vertex r and a corresponding assignment $\mathcal{B}(e)$ that minimize the *average distance* between r and other vertices of the graph, defined as $\frac{1}{n} \cdot \sum_{v\in V} \delta_{\mathcal{B}}(r, v)$. The vertex r that minimizes the budget median problem is called the *budget median*.

1.2. Motivation

We were motivated by communication optimization problems in which for a fixed 'budget' one needs to design the 'best' network layout. The quality of service (*QoS*) of a link between two nodes depends on two main factors: i) The distance between the nodes. ii) The infra-structure of the link. While the location of the nodes is often fixed and cannot be changed, the infra-structure type and service can be upgraded – it is a price-dependent service. Quality of service is related to different parameters such as: bandwidth, delay time, jitter, packet error rate and many others. Given a network graph, the desired objective is to have the best *QoS* for a given (fixed) budget.

In this paper we focus on minimizing the maximum and average delay–times using a fixed budget. Informally, consider the case of an *ISP* (Internet Service Provider) which would like to maximize the overall *QoS* of its network using a given budget. The problem is what links (edges) in the network graph should be upgraded and how to allocate the given budget over those links. We consider a model in which assigning a budget b to a network link e will imply a delay proportional to $\frac{1}{b}$. One may view our model as a *continuous* version of a multi-edge graph, in which the *QoS* of a network link is price related. Our model is motivated by various network design problems – for example, a better (and more expensive) network link implies lower delay and better *QoS* – yet these are commonly modeled as discrete problems, where there are only a fixed number of possible 'upgrades' to the network link. A continuous network design model is commonly used in traffic network problems (see e.g., [14]), and indeed our price-dependent model is continuous.

1.3. Related work

The problems of center and median points on graphs (networks) have been studied extensively, see [8,17] for a detailed survey on *facility location*. There are various optimization problems dealing with finding the best graph; A typical *graph or network improvement* problem considers a graph which needs to be improved by adding the smallest number of edges in order to satisfy some constraint (e.g., maximal radius), see [4,10,20]. Spanner graph problems [15] consider what can be seen as the *inverse* case of network improvement problems. In a typical spanner problem we would like to keep the smallest subset of edges from the original graph while maintaining some constraint, such as low degree or diameter. Observe that both network improvement and spanner graph problems can be modeled as discrete versions of our suggested new model. For a general framework for graph optimization and graph layout design see [7].

Lin and Mouratides [11,12] considered the model where each edge weight may be reduced by a some value specific to that edge, and they minimized the sum of the inter-point distances between source-destination pairs. The *upgrading arc* model is one in which any edge weight can be improved by the same fixed factor; minimizing radius, diameter and median distances have all been studied in this setting [21,2,16]. (There is also a variant of this model in which vertices are upgraded instead of edges [18].) A continuous weight reduction model with cost linear in the edge weight reduction was considered in [9,20]. Finally, Carmi and Chaitman-Yerushalmi [3] investigated the influence of Steiner points on the weight of the minimum spanning tree under the cost model of the current paper.

Table 1
Notations that are used throughout the paper to present the new budget graph model.

Notation	Explanation
$G = \langle V, E \rangle$	a general undirected graph (induced by some metric space)
$\ell(e)$	the (a priori) length of an edge $e \in E$
$\mathcal{B}(e)$	the budget fraction allocated to $e \in E$
$\mathcal{B} = \{b_1, \ldots b_{\lvert E \rvert}\}$	an alternative notation for the function $\mathcal{B}$
$\omega(e) = \frac{\ell(e)}{\mathcal{B}(e)}$	the (budget implied) weight of $e \in E$
$(G, w) = G(\mathcal{B})$	the *budget-graph* implied by an allocation $\mathcal{B}$
$\delta_{\mathcal{B}}(v, u)$	the distance between two vertices in $G(\mathcal{B})$

1.4. Our contribution

In this paper we present linear time algorithms for rooted and unrooted budget radius and budget median problems on trees. We also prove that the budget radius problem is NP-hard on complete metric graphs, and devise an $O(\log^2(n))$ approximation algorithm in this setting.

2. Preliminaries

In this section we introduce basic notations and definitions that are used for describing the suggested budget graph framework. Let $G = \langle V, E \rangle$ be a graph with some length function $\ell : E \mapsto \mathbb{R}^+$. A *valid budget allocation* $\mathcal{B}(\cdot)$ to E is a non-negative real function, such that $\sum_{e \in E} \mathcal{B}(e) = 1$.[2] Let $E = \{e_1, \ldots, e_{|E|}\}$. We denote $b_i \stackrel{\text{def}}{=} \mathcal{B}(e_i)$, and for every $E' \subseteq E$ we denote $\mathcal{B}(E') = \sum_{e_i \in E'} b_i$. Given a valid budget allocation $\mathcal{B}$ to E, the *weight* of an edge $e \in E$, denoted $\omega_{\mathcal{B}}(e)$, is a function of $\ell(e)$ and $\mathcal{B}(e)$. Throughout this paper we consider the case where $\omega_{\mathcal{B}}(e) \stackrel{\text{def}}{=} \frac{\ell(e)}{\mathcal{B}(e)}$.

Definition 2.1 *(Weighted distance).* Let $u, v \in V$ be two vertices. The *weighted distance* between v and u, denoted $\delta_{\mathcal{B}}(v, u)$, is the minimum weight over all simple paths between u and v. Namely, $\delta_{\mathcal{B}}(v, u) \stackrel{\text{def}}{=} \min(\{\sum_{e \in P} \omega_{\mathcal{B}}(e) : P \text{ is a simple path from } v \text{ to } u\})$.

Table 1 summarizes the above notations. In the rest of this section we present two specific facility location problems in the budget graph framework. In Section 2.1, we describe the budget radius problem and in Section 2.2, we describe the budget median problem.

2.1. The budget radius problem

We next introduce some definitions and notations to define the setting of the *budget radius* problem on graphs. In the first setting we consider, a candidate center node to the graph is given and the goal is to find an optimal budget allocation that minimizes the radius of graph with respect to the resulting distances between nodes in the graph (the weighted radius). In a second setting, the candidate center node is not given and the goal is to find both a center node and a budget allocation that are together optimal with respect to the weighted radius. The formal definitions follow.

Definition 2.2 *(Rooted weighted radius).* Given a valid budget allocation $\mathcal{B}$ to E and a vertex $r \in V$, the *weighted radius* of G with respect to r is defined as $\text{wr}_{\mathcal{B}}(r) = \text{wr}_{\mathcal{B}}(G, r) \stackrel{\text{def}}{=} \max_{v \in V}(\delta_{\mathcal{B}}(r, v))$.

Given a graph $G = \langle V, E \rangle$ (induced by some metric) and a node $r \in V$, we next define what it means for an allocation to be optimal with respect to the budget radius problem. We usually omit G from the notation whenever it can be derived from the context.

Definition 2.3 *(Rooted budget radius).* An optimal allocation for (G, r) with respect to the budget radius problem is a valid allocation for which the weighted radius with respect to r is minimized. There may be several optimal allocations. We take an arbitrary optimal allocation and denote it by $\mathcal{B}^*_r = \mathcal{B}^*_r(G)$. We further refer to this allocation as *the optimal allocation* for (G, r).

The *budget radius* of G with root r, denoted $\text{BR}(r) = \text{BR}(G, r)$, is the weighted radius of G with respect to r and $\mathcal{B}^*_r$, i.e., $\text{BR}(r) = \text{wr}_{\mathcal{B}^*_r}(r)$.

[2] Here and in the rest of the paper we assume that the total budget B equals 1; this is without loss of generality since an optimal solution with budget of 1 is easily scaled to any budget B.

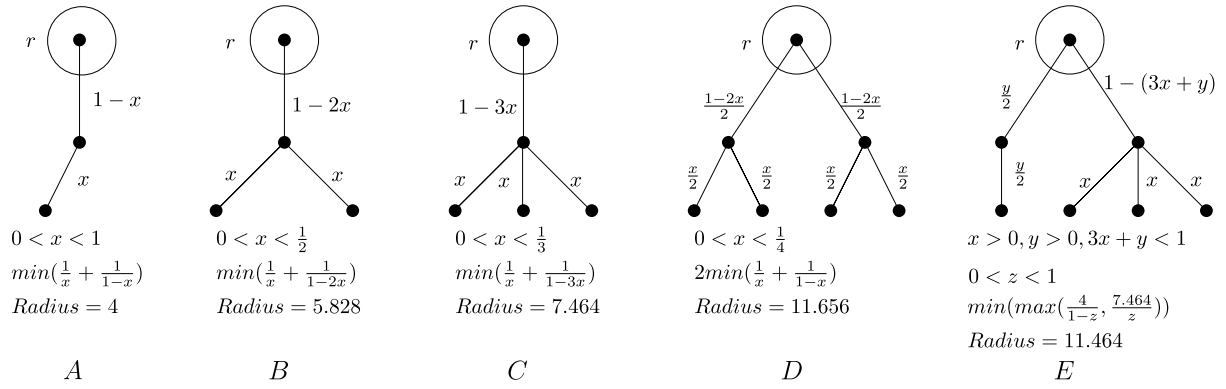

Fig. 1. A few simple examples of a budget graph radius problem on rooted trees (with a given center r). Assume that each edge e has length $\ell(e) = 1$. The total budget is 1 and the optimal budget allocation of each edge is shown as a function of x. In case A, the lower edge is assigned budget x, and so the upper edge is assigned budget $1 - x$. The optimal solution minimizes $f(x) = \frac{1}{x} + \frac{1}{1-x}$, so $x = \frac{1}{2}$ and the radius is 1. In case B, the optimal solution assigns the same weight x to the two lower edges, and then minimizes $f(x) = \frac{1}{1-2x} + \frac{1}{x}$. So $x = \frac{1}{2+\sqrt{2}}$ and the radius is approximately 5.828. In case C the function is: $f(x) = \frac{1}{1-3x} + \frac{1}{x}$, so $x = \frac{1}{3+\sqrt{3}}$ and the radius is approximately 7.464. Case D is composed of two instances of B and therefore the radius is twice the budget radius of case B, 11.656. Case E is composed from cases A, C therefore the radius is the sum of the two radii = 11.464.

We next give the appropriate definitions for the setting where the root is not given as part of the input to the problem. We call this problem the *unrooted* budget radius problem, where the goal is to find a node in the graph that minimizes the radius is sometimes referred to as the center point problem.

Definition 2.4 *(Unrooted budget radius).* The *budget radius* of a graph G is the value $\mathrm{BR} = \mathrm{BR}(G) \stackrel{\text{def}}{=} \min_{v \in V} \mathrm{BR}(G, v)$. We refer to a pair $(\mathcal{B}^*, r^*)$ as the optimal allocation for G (in the unrooted setting) if $\mathcal{B}^*$ is a valid allocation to E and $r^* \in V$ is the vertex with the smallest corresponding radius, i.e., $\mathrm{wr}_{\mathcal{B}^*}(r^*) = \mathrm{BR}$.

We demonstrate the above definitions using a few examples presented in Fig. 1.

2.2. The budget median problem

Given a graph, the general median problem is defined as follows: find a node (m^*) in the graph from which the sum of all weighted distances to all other nodes in the graph is minimized. On *budget graphs*, the *budget median* problem is to find a median node and a corresponding optimal allocation which minimizes the sum of distances between the median node and all the other nodes in the graph (with respect to the budget allocation).

Definition 2.5 *(Rooted weighted budget median).* Given a valid budget allocation $\mathcal{B}$ to E and a vertex $m \in V$, the *weighted median* of G with respect to m is defined as $\mathrm{wm}_{\mathcal{B}}(m) = \mathrm{wm}_{\mathcal{B}}(G, m) \stackrel{\text{def}}{=} \sum_{v \in V} (\delta_{\mathcal{B}}(m, v))$.

Given a graph $G = \langle V, E \rangle$ (induced by some metric) and a node $m \in V$, we next define what it means for an allocation to be optimal with respect to the budget median problem.

Definition 2.6 *(Rooted budget median).* An optimal allocation for (G, m) with respect to the budget median problem is a valid allocation for which the sum of all weighted distances to m is minimized. There may be several optimal allocations. We take an arbitrary optimal allocation and denote it by $\mathcal{B}^*_m = \mathcal{B}^*_m(G)$. We further refer to this allocation as *the optimal allocation* for (G, m).

The *budget median* of G with root m, denoted $\mathrm{BM}(m) = \mathrm{BM}(G, m)$, is the weighted sum of distances from m to all other nodes of G with respect to $\mathcal{B}^*_m$, i.e., $\mathrm{BM}(m) = \mathrm{wm}_{\mathcal{B}^*_m}(m)$.

We next give the definitions for the setting where the root is not given as part of the input to the problem. We call this problem the *unrooted* budget median problem.

Definition 2.7 *(Unrooted budget median).* The *budget median* of a graph G is the value $\mathrm{BM} = \mathrm{BM}(G) \stackrel{\text{def}}{=} \min_{m \in V} \mathrm{BM}(G, m)$. We refer to a pair $(\mathcal{B}^*, m^*)$ as the optimal allocation for G (in the unrooted setting) if $\mathcal{B}^*$ is a valid allocation to E and $m^* \in V$ is the vertex with the smallest corresponding sum of distances to all other nodes in the graph, i.e., $\mathrm{wm}_{\mathcal{B}^*}(m^*) = \mathrm{BM}$.

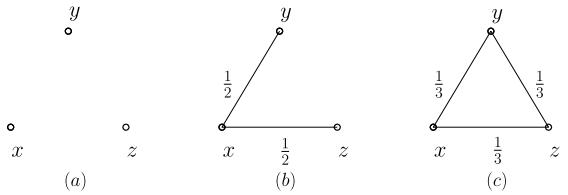

Fig. 2. (*a*) Three points in the plane: x, y, z are the nodes of a unit equilateral triangle. (*b*) Tree: optimal radius (2), non-optimal diameter (4). (*c*) Cycle graph: non-optimal radius (3), optimal diameter (3).

3. The budget radius problem for trees

In this section, we present an algorithm for solving the budget radius problem for trees. We start by showing that the solution to the budget radius problem for a general graph G is always in the form of a tree spanning G. In Section 3.1, we solve the rooted budget radius problem for trees. In Section 3.2, we turn to the unrooted variant of the problem.

Lemma 3.1. *Let $G = \langle V, E\rangle$ be a connected graph and let ℓ be a length function on E. An optimal budget allocation (with respect to the budget radius problem) (B^*, r^*) for G, has the property that the support graph $H = \langle V, E_{\mathcal{B}^*}\rangle$, where $E_{\mathcal{B}^*} = \{e_i \in E : b_i^* > 0\}$ is a tree spanning G.*

Proof. First, assume by contradiction that H is not connected. Then for any center vertex r^*, there is a vertex $v \in V$ such that there is no path in H connecting r^* to v. In this case, however, any path between r^* and v in $(G, w) = G(\mathcal{B}^*)$ contains an edge of infinite weight. Hence, $\delta_{\mathcal{B}^*}(r, v) = \infty$, and so the radius implied by the budget allocation $\mathcal{B}^*$ is ∞. This means that $\mathcal{B}^*$ is not optimal, since there are allocations yielding a finite radius, e.g., the uniform allocation, assigning an equal portion of the budget to each edge (i.e., $\mathcal{B}(e) = \frac{1}{|E|}$ for all $e \in E$).

Second, assume by contradiction that H contains a cycle. Let H' be a shortest path tree rooted at r^*. Hence, there exists at least one edge $e \in E_H \setminus E_{H'}$. By definition, $\mathcal{B}^*(e) > 0$. We define a new budget allocation $\mathcal{B}'$ distributing the budget portion allocated to e evenly among all edges in H'. It is easy to verify that the budget radius of $\mathcal{B}'$ is strictly smaller than that of $\mathcal{B}^*$. This is a contradiction to the optimality of $\mathcal{B}^*$. Hence, H is a spanning tree. □

An analogous argument applies to the budget median problem. We note, however, that the above is not true for the budget diameter problem (see Fig. 2).

The above lemma motivates us to explore the budget radius problem for the subclass of trees, and indeed we present an algorithm solving this problem. We first consider the case where a designated center node r is given as a part of the input, and an optimal budget allocation $\mathcal{B}^*$ is sought. Using standard terminology, we refer to r as the root of the tree (as opposed to the center). We then consider the general case in trees, where the problem is to find a pair $(\mathcal{B}^*, r^*)$ minimizing the budget radius of the tree.

3.1. The budget radius problem for rooted trees

We next consider two sub-families of rooted trees that will later be the basis for our recursive construction of an optimal valid budget allocation to the edges of general trees. First, we consider a tree in which the root has only one child.

Lemma 3.2. *Let T be a tree rooted at r, with some length function ℓ on the edges of T. Assume r has a single child r' (the root of the subtree T'), and let $R' = \mathrm{BR}(T', r')$ and $q = \ell(r, r')$. Then an optimal budget allocation $\mathcal{B}^*$ assigns to the edge $e = (r, r')$ a fraction $\beta = \frac{\sqrt{q}}{\sqrt{R'}+\sqrt{q}}$ of the budget. It follows that $\mathrm{BR}(T, r) = \frac{q}{\beta} + \frac{R'}{1-\beta} = (\sqrt{q} + \sqrt{R'})^2$.*

Proof. Let E be the set of edges in T and $E' = E \setminus \{e\}$ be the set of edges of T' (see Fig. 3a). Given any valid budget allocation $\mathcal{B}$ for E, let $\mathcal{B}'$ be the scaling of the restriction of $\mathcal{B}$ to E', defined by $\mathcal{B}'(e') = \frac{\mathcal{B}(e')}{1-\mathcal{B}(e)}$ for every $e' \in E'$. Note that with this scaling, $\mathcal{B}'$ is a valid budget allocation for E', i.e., $\sum_{e' \in E'} \mathcal{B}'(e') = 1$. Since any path from r to any leaf of T must start with the edge $e = (r, r')$, it follows that

$$\mathrm{wr}_{\mathcal{B}}(r) = \frac{q}{\mathcal{B}(e)} + \frac{\mathrm{wr}_{\mathcal{B}'}(r')}{1 - \mathcal{B}(e)}.$$

Hence, for $\mathcal{B}$ to be optimal for T with root r, we must have $\mathcal{B}'$ be optimal for T' and r'. In addition, given $R' = \mathrm{BR}(T', r')$, the budget radius of T with root r is obtained by assigning a β fraction of the budget to e, for β that minimizes the function $\mathrm{wr}_{\mathcal{B}}(r) = \frac{q}{\beta} + \frac{R'}{1-\beta}$. It follows that $\mathrm{BR}(T, r) = \frac{q}{\beta} + \frac{R'}{1-\beta}$ for $\beta = \frac{\sqrt{q}}{\sqrt{R'}+\sqrt{q}}$. □

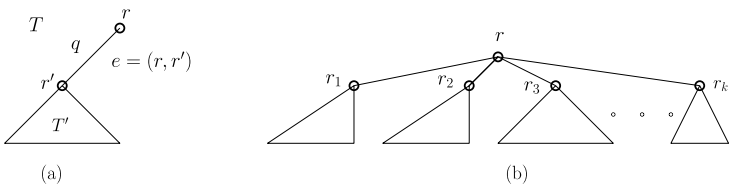

Fig. 3. (a) The case that r has only a single child. (b) The general case.

We next consider the general family of (rooted) trees (Fig. 3b). Let $T = (V, E)$ be a tree, rooted at r, such that r has k children $r_1, r_2, \ldots, r_k$, where r_i is the root of its subtree $T_i = (V_i, E_i)$. Denote by $T'_i = (V'_i, E'_i)$ the subtree of T, rooted at r and containing T_i. That is, $V'_i = V_i \cup \{r\}$, and $E'_i = E_i \cup \{(r, r_i)\}$. Clearly, the edge-sets E'_i are all disjoint. Given a valid budget allocation $\mathcal{B}$ to E, for each index $i \in \{1, 2, .., k\}$ denote by $l_i \in V_i$ the leaf l for which $\delta_{\mathcal{B}}(r, l)$ is the largest within T'_i. Recall that the weighted radius $\mathrm{wr}_{\mathcal{B}}(r)$ is determined by the maximum weighted distance to some l_i. In other words, $\mathrm{wr}_{\mathcal{B}}(r) = \max_{1 \le i \le k}(\delta_{\mathcal{B}}(r, l_i))$. Next, we show that in any optimal budget allocation $\mathcal{B}^*$ for such T, the fraction of the budget assigned to the edges of each subtree T'_i is directly correlated to its relative weighted radius.

Lemma 3.3. *Let $T = (V, E)$ be a tree rooted at r, with some length function ℓ on the edges of T. Assume that r has k children $r_1, r_2, \ldots, r_k$ where r_i is the root of the subtree $T_i = (V_i, E_i)$, and let $\mathcal{B}^*$ be an optimal budget allocation to E. For each $1 \le i \le k$, let T'_i and l_i be as in the foregoing discussion (i.e., l_i is maximal in T'_i with respect to $\delta_{\mathcal{B}^*}(r, \cdot)$). Then, for all $1 \le i, j \le k$ we have $\delta_{\mathcal{B}^*}(r, l_i) = \delta_{\mathcal{B}^*}(r, l_j)$.*

Proof. Assume that for some $1 \le i, j \le k$, it holds that $\delta_{\mathcal{B}^*}(r, l_i) > \delta_{\mathcal{B}^*}(r, l_j)$. We show that it is then possible to present a better budget allocation for T, which is a contradiction. Let $\rho = \frac{\delta_{\mathcal{B}^*}(r, l_j)}{\delta_{\mathcal{B}^*}(r, l_i)}$ and consider an alternative budget allocation in which each edge e in E'_j is assigned a ρ fraction of its current budget, i.e., $\rho \cdot \mathcal{B}^*(e)$ (while assignments to all other edges stay the same as before). The length of each path from r to a leaf in T'_j would be multiplied by $1/\rho$. Hence, the maximum distance from r to any leaf in the T'_j would be at most $\delta_{\mathcal{B}^*}(r, l_i)$. This allocation is therefore as good as $\mathcal{B}^*$ (with respect to the weighted radius) although the sum of assigned values is not 1, but rather, $1 - (1 - \rho) \cdot \mathcal{B}^*(E'_j) < 1$. Turn this into a valid budget allocation by dividing the remaining $(1 - \rho) \cdot \mathcal{B}^*(E'_j)$ budget equally among all edges in E, we obtain a better valid budget allocation to E. That is, we fix a new allocation $\mathcal{B}'$ by setting $\mathcal{B}'(e) = \rho \cdot \mathcal{B}^*(e) + \frac{(1-\rho) \cdot \mathcal{B}^*(E'_j)}{|E|}$ if $e \in E'_j$, and $\mathcal{B}'(e) = \mathcal{B}^*(e) + \frac{(1-\rho) \cdot \mathcal{B}^*(E'_j)}{|E|}$ otherwise. The budget radius of $\mathcal{B}'$ is smaller than that of $\mathcal{B}^*$, in contradiction to the optimality of $\mathcal{B}^*$. □

The following corollary describes how any optimal valid budget allocation must divide the budget among the disjoint sets of edges of the subtrees T'_i.

Corollary 3.4. *Let T be a tree as above. Then in any optimal budget allocation $\mathcal{B}^*$ for E it holds that $\mathcal{B}^*(E'_i) = \frac{\mathrm{BR}(T'_i, r)}{\sum_{j=1}^{k} \mathrm{BR}(T'_j, r)}$. Thus, an optimal solution in this case is given by $\mathrm{BR}(T, r) = \sum_{j=1}^{k} \mathrm{BR}(T'_j, r)$.*

Proof. Let l_i's be as above, i.e., $l_i \in V_i$ is the leaf l for which $\delta_{\mathcal{B}^*}(r, l)$ is maximal within T'_i. Denote $\beta_i = \mathcal{B}^*(E'_i)$ and let $\mathcal{B}'_i$ be the scaling of the restriction of $\mathcal{B}^*$ to E'_i, defined by $b'_i(e') = \frac{\mathcal{B}^*(e')}{\beta_i}$ for every $e' \in E'_i$. Clearly, each $\mathcal{B}'_i$ is a valid budget allocation to T'_i. We claim that it is also an optimal one. By Lemma 3.3, for all $1 \le i, j \le k$ it holds that $\delta_{\mathcal{B}^*}(r, l_i) = \delta_{\mathcal{B}^*}(r, l_j)$. If for some tree-leaf l_i it holds that $\mathcal{B}'_i$ is not optimal for T'_i, then choose an optimal valid budget allocation $\mathcal{B}''_i$ for T'_i and scale it back to obtain a valid budget allocation $\hat{\mathcal{B}}$ by setting $\hat{\mathcal{B}}(e') = \mathcal{B}''(e') \cdot \beta_i$ for each $e' \in E'_i$, and $\hat{\mathcal{B}}(e) = \mathcal{B}^*_i(e)$ for each $e' \notin E'_i$. This reduces the distance of the farthest leaf from r within T'_i, while the distance to any leaf outside T'_i stays as with $\mathcal{B}^*$. Specifically, we have $\mathrm{wr}_{\hat{\mathcal{B}}}(r) \le \mathrm{wr}_{\mathcal{B}^*}(r)$. However, by Lemma 3.3, $\hat{\mathcal{B}}$ is not optimal and hence, $\mathcal{B}^*$ is not optimal either – a contradiction.

By the above it holds for all $1 \le i \le k$ that $\mathrm{BR}(T'_i, r) = \beta_i \cdot \delta_{\mathcal{B}^*}(r, l_i)$. Thus, by Lemma 3.3, for all $1 \le i, j \le k$ it holds that $\frac{\mathrm{BR}(T'_i, r)}{\beta_i} = \frac{\mathrm{BR}(T'_j, r)}{\beta_j}$. Since it also holds that $\sum_{i=1}^{k} \beta_i = 1$, we have that for all $1 \le i \le k$ it holds that $\beta_i = 1 - \sum_{j \ne i} \beta_j = 1 - \beta_i \cdot \sum_{j \ne i} \frac{\mathrm{BR}(T'_j, r)}{\mathrm{BR}(T'_i, r)}$. Hence, $\beta_i = \frac{\mathrm{BR}(T'_i, r)}{\sum_{j=1}^{k} \mathrm{BR}(T'_j, r)}$. Furthermore, since for any $1 \le i \le k$ we have that $\mathrm{BR}(T, r) = \delta_{\mathcal{B}^*}(r, l_i)$ (specifically, since $\mathrm{BR}(T, r) = \delta_{\mathcal{B}^*}(r, l_1) = \frac{\mathrm{BR}(T'_1, r)}{\beta_1}$), it follows that $\mathrm{BR}(T, r) = \sum_{j=1}^{k} \mathrm{BR}(T'_j, r)$. □

Theorem 3.5. *Given a tree T rooted at r, an optimal valid budget allocation for T and r can be found in time linear in the size of T.*

Proof. T is a rooted tree, and we will utilize an inductive construction. First, assume T is a single node r. In this case, no budget is needed and $\mathrm{BR}(T,r)=0$. Assume T is rooted at r, such that r has k children $r_1, r_2, \ldots, r_k$. Denote by T_i the subtree of T rooted at r_i, and containing all vertices (and edges) of the subtree rooted at r_i (and only these vertices). Denote by $T'_i=(V'_i,E'_i)$ the subtree of T rooted at r, induced by adding the edge (r,r_i) to T_i. Formally, $V'_i=V_i\cup\{r\}$, and $E'_i=E_i\cup\{(r,r_i)\}$. Thus, each T'_i is a rooted tree where the root (r) has a single child, no T'_i, T'_j for $i\neq j$ share any vertex other than r, and E'_i, E'_j are disjoint for all $i\neq j$.

By Corollary 3.4, if we know $\mathrm{BR}(T'_i,r)$ for all $1\leq i\leq k$, we can derive an optimal valid budget allocation for T,r. In order to obtain a $\mathrm{BR}(T'_i,r)$, it suffices to have an optimal solution for the subtree of r_i, which, by the induction hypothesis can be done (using Lemma 3.2).

Note that we evaluate the optimal solution for every subtree of every vertex in T exactly once and thus the procedure runs in linear time. □

Remark 3.6. In the finite precision setting, the algorithm implied by Theorem 3.5 requires words of $O(\log(nL/\epsilon))$ bits in order to compute a solution within a factor $1+\epsilon$ of the true solution, where $L=\sum_{e\in E}\ell(e)$ and we have assumed the minimum length in the graph is at least 1. To see this, first note that increasing the length of all edges by at most a factor of $1+\epsilon$ can increase the budget radius by at most this factor. If the total budget is 1, the optimal budget radius cannot be greater than nL, as this radius can be achieved by assigning each edge e a budget of $\frac{\ell(e)}{L}$. Hence, it must be that $\frac{1}{nL}\leq\mathcal{B}^*(e)\leq 1$ or else the optimal budget radius is not achieved. It follows that a word of $O(\log(nL/\epsilon))$ bits is sufficient to record an edge budget assignment to within a factor of $1+\epsilon$. The rounding down due to limited precision is equivalent to increasing the length of that edge by a factor of at most $1+\epsilon$, and the resulting budget radius by at most this factor.

The following lemma will prove helpful below, but is interesting in its own right. It captures some of the tricky nature of the budget radius problem, as it shows the connection between two seemingly unrelated quantities. The first is the weight of a minimum spanning tree (MST) of a given graph and the second is the optimal solution for the budget radius problem for that graph.

Lemma 3.7. *Given a tree $T=(V,E)$ rooted at r, with some length function ℓ on E, the budget radius of T is at least the sum of lengths of the edges of T, i.e.,* $\mathrm{BR}(T,r)\geq\sum_{e\in E}\ell(e)$.

Proof. We prove the above lemma by induction. If T has no edges, then both values are 0. If r has only one child r' (the root of the subtree T'), then by Lemma 3.2, since any optimal allocation $\mathcal{B}^*$ must assign $\mathcal{B}^*((r,r'))>0$, we have $\mathrm{BR}(T,r)>\ell((r,r'))+\mathrm{BR}(T',r')$, which, by the induction hypothesis is at least $\sum_{e\in E}\ell(e)$. Otherwise, assume r has k children $(r_1\ldots r_k)$ and denote T_i the subtree induced by r and the vertices of the subtree of r_i. By 3.4 $\mathrm{BR}(T,r)=\sum_{i=1}^{k}\mathrm{BR}(T_i,r)$. Hence, by the induction hypothesis, the lemma follows. □

3.2. The budget radius for unrooted trees

In this section we consider the budget radius problem for unrooted trees, i.e., where the root of the tree is not given as part of the input. Clearly, one can invoke the algorithm from Theorem 3.5 with every vertex v as a candidate center vertex r, and select the vertex v for which $\mathrm{BR}(T,v)$ is minimal. However, this naive algorithm requires $O(n^2)$ time. We next show how to construct a linear time algorithm for this problem (in fact, for a tree T, our algorithm computes $\mathrm{BR}(T,v)$ for every v in T).

Lemma 3.8. *Let $T=(V,E)$ be a tree rooted at r, with some length function ℓ on E. Let $v\in V$ be a neighbor (a child) of r. Denote by $T_v=(V_v,E_v)$ the subtree of v, and denote by T'_v the subtree of v augmented by the edge $e=(r,v)$ (i.e., $T'_v=(V_v\cup r, E_v\cup e)$). It is possible to compute, in constant time,* $\mathrm{BR}(T,v)$ *given* $\mathrm{BR}(T,r)$, $\mathrm{BR}(T_v,v)$, *and* $\mathrm{BR}(T'_v,r)$, *see Fig. 4.*

Proof. Denote by $\hat{T}$ the tree obtained by omitting T_v from T, formally, $\hat{T}=(\hat{V},\hat{E})$, where $\hat{V}=V\backslash(V_v\backslash\{v\})$ and $\hat{E}=E\backslash E_v$. In addition, denote by $\hat{T}'$ the tree obtained by omitting the edge $e=(r,v)$ from $\hat{T}$, i.e., $\hat{T}'=(\hat{V}\backslash\{v\},\hat{E}\backslash\{e\})$.

It can be easily derived from Corollary 3.4 that $\mathrm{BR}(T,v)=\mathrm{BR}(T_v,v)+\mathrm{BR}(\hat{T},v)$. By Lemma 3.2, we can compute $\mathrm{BR}(\hat{T},v)$ from $\mathrm{BR}(\hat{T}',r)$ and $\ell(e)$, in constant time. Finally, we compute $\mathrm{BR}(\hat{T}',v)$, using Corollary 3.4 again, to obtain $\mathrm{BR}(\hat{T}',r)=\mathrm{BR}(T,r)-\mathrm{BR}(T'_v,r)$. □

Roughly, our algorithm will traverse the tree twice. First, we traverse the tree, computing the algorithm of Theorem 3.5 for an arbitrary root r (say, $r=v_1$). Recall that this algorithm traverses the tree in a bottom-up fashion, i.e., from the leaves to the root, and that an optimal solution for each vertex is calculated, with respect to the subtree below it. We then traverse the tree in a top–down fashion, while for each vertex v that is a child of v', we compute an optimal budget radius for the tree with root v, given an optimal budget radius for the tree with root v' and the information stored in v from the first traversal.

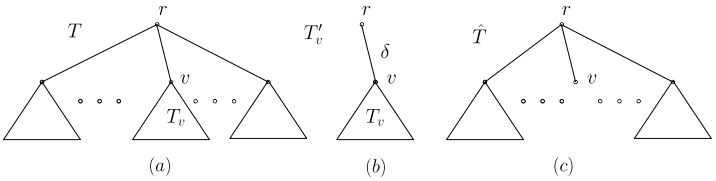

Fig. 4. (a) The original tree rooted at r. (b) Considering v as the root of T'_v. (c) The tree $\hat{T}$.

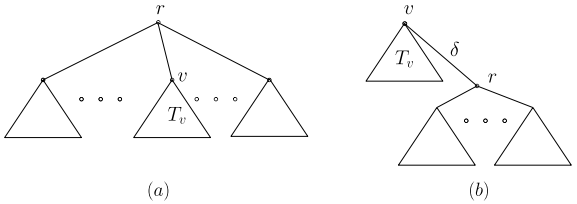

Fig. 5. (a) The original tree rooted at r. (b) Considering v as the root of the tree. Computation of an optimal budget allocation for the tree rooted at v can be done in constant time, given an optimal budget allocation for the tree rooted at r.

Theorem 3.9. *Given a tree $T = (V, E)$ with some length function ℓ on E, it is possible to compute an optimal allocation for T, i.e., a pair $(\mathcal{B}^*, r^*)$, such that $\mathrm{wr}_{\mathcal{B}^*}(r^*) = \mathrm{BR}(T)$. Furthermore, this can be done in linear time in the size of T (see Fig. 5).*

Proof. Our algorithm traverses the tree twice. In the first pass, we set an arbitrary vertex r to be the root (say, $r = v_1$) and traverse the tree in a bottom up manner, following the algorithm described in Theorem 3.5.

For any vertex $v \in V$, denote by $T_v = (V_v, E_v)$ the subtree of v, and denote by $p(v)$ the parent of v. Denote by T'_v the subtree of v augmented by the edge $(p(v), v)$, i.e., $T'_v = (V_v \cup \{p(v)\}, E_v \cup \{(p(v), v)\})$. We compute for each vertex v the value of an optimal budget radius with respect to the subtree of v, i.e., $\mathrm{BR}(T_v, v)$. Recall that in order to do so, we compute for each child u of v, not only the budget radius for T_u (i.e., $\mathrm{BR}(T_u, u)$), but also the budget radius for the augmented subtree of u (with root v), i.e., $\mathrm{BR}(T'_u, v)$. Here, we also store the two local values, $\mathrm{BR}(T_v, v)$ and $\mathrm{BR}(T_{p(v)}, p(v))$, for each node v we traverse.

In the second pass we traverse the tree in a top–down manner, starting at the root r and moving from each vertex to all its children. For each vertex v, we compute the budget radius for the whole tree T with root v (i.e., $\mathrm{BR}(T, v)$). By Lemma 3.8, we can do so in constant time since we have already computed $\mathrm{BR}(T, p(v))$, as well as $\mathrm{BR}(T_v, v)$ and $\mathrm{BR}(T_{p(v)}, p(v))$. The algorithm returns the pair $(\mathcal{B}^*, r^*)$, for which the budget radius is minimal. □

4. Generalization to median point

In this section we generalize the center point algorithms to *median point*. We begin with the rooted weighted budget median. First, as was done for budget radius (Lemma 3.2), we consider a tree in which the root has only one child.

Lemma 4.1. *Let T be a tree rooted at r, with some length function ℓ on the edges of T. Assume r has a single child r' (the root of the subtree T' of size $n = |T'| = |T| - 1$), and let $M' = \mathrm{BM}(T', r')$ and $q = \ell(r, r')$. Then an optimal budget allocation $\mathcal{B}^*$ assigns to the edge $e = (r, r')$ a fraction $\beta = \frac{\sqrt{nq}}{\sqrt{M'}+\sqrt{nq}}$ of the budget. It follows that $\mathrm{BM}(T, r) = \frac{nq}{\beta} + \frac{M'}{1-\beta} = (\sqrt{nq} + \sqrt{R'})^2$.*

Proof. Let E be the set of edges in T and $E' = E \setminus \{e\}$ be the set of edges of T' (see Fig. 3a). Given any valid budget allocation $\mathcal{B}$ for E, let $\mathcal{B}'$ be the scaling of the restriction of $\mathcal{B}$ to E', defined by $\mathcal{B}'(e') = \frac{\mathcal{B}(e')}{1-\mathcal{B}(e)}$ for every $e' \in E'$. Note that with this scaling, $\mathcal{B}'$ is a valid budget allocation for E', i.e., $\sum_{e' \in E'} \mathcal{B}'(e') = 1$. Since any path from r to any leaf of T must start with the edge $e = (r, r')$, it follows that

$$\mathrm{wm}_{\mathcal{B}}(r) = \frac{nq}{\mathcal{B}(e)} + \frac{\mathrm{wm}_{\mathcal{B}'}(r')}{1 - \mathcal{B}(e)}.$$

Hence, for $\mathcal{B}$ to be optimal for T with root r, we must have $\mathcal{B}'$ be optimal for T' and r'. In addition, given $M' = \mathrm{BM}(T', r')$, the budget median of T with root r is obtained by assigning a β fraction of the budget to e, for β that minimizes the function $\mathrm{wm}_{\mathcal{B}}(r) = \frac{nq}{\beta} + \frac{M'}{1-\beta}$. It follows that $\mathrm{BM}(T, r) = \frac{nq}{\beta} + \frac{M'}{1-\beta}$ for $\beta = \frac{\sqrt{nq}}{\sqrt{M'}+\sqrt{nq}}$. □

We next consider the general family of (rooted) trees (Fig. 3b). Recall the definitions that preceded Lemma 3.3.

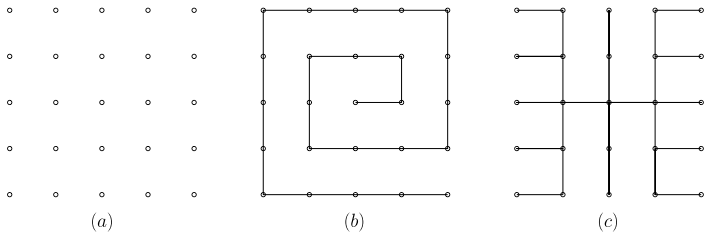

Fig. 6. Using an MST-like heuristic may lead to an $O(n^{0.5})$ approximation ratio with respect to the *center point* problem (minimum radius). (a) A grid based set of points. (b) A path-like MST. (c) A solution with radius $O(n^{0.5})$.

Lemma 4.2. *Let $T = (V, E)$ be a tree rooted at r, with some length function ℓ on the edges of T. Assume that r has k children $r_1, r_2, \ldots, r_k$ where r_i is the root of the subtree $T_i = (V_i, E_i)$. Let T_i' denote the subtree of T rooted at r and containing also T_i. Let $X_i = \mathrm{BM}(T_i', r)$. Then an optimal budget $\mathcal{B}^*$ assigns to edge $e_i = (r_i, r)$ a fraction $\frac{\sqrt{X_i}}{\sum_{j=1}^{k}\sqrt{X_j}}$ of the budget, and so $\mathrm{BM}(R, r) = (\sum_{j=1}^{k}\sqrt{X_j})^2$.*

Proof. Recall that the value of X_i is given in Lemma 4.1. Then the budget between all subtrees can be normalized using the following minimization problem: $\min(\frac{X_1}{B_1} + \frac{X_2}{B_2} + \ldots + \frac{X_k}{B_k})$ where $B_1 + B_2 + \ldots + B_k = 1$. The result follows. □

Theorem 4.3. *Given a tree T rooted at r, an optimal valid budget allocation for the rooted and unrooted budget median problem on T and r can be found in time linear in the size of T.*

Proof. The rooted case follows immediately from Lemma 4.2. For the unrooted case, consider an edge connecting two vertices v_1, v_2 and their respective disjoint subtrees T_1, T_2. Note that if $|T_1| < |T_2|$, then v_1 cannot be the optimal median: the cost of taking v_2 as a median is less than the cost of taking v_1, since then edge $e = (v_1, v_2)$ will be traversed only $|T_1|$ times instead of $|T_2|$ times. It follows that the optimal median is a 'middle' vertex of the graph, one possessing the property that for all edges incident to it, it is always a member of the heavier subtree. A middle vertex can be found in linear time. □

5. Budget radius – on complete metric graphs

In this section we consider the problem of optimizing the budget radius for a complete graph over n vertices, induced by some metric space $M = (V, d)$. We present a hardness proof showing that in this setting the budget radius problem is NP-hard. We then present an $O(\log^2(n))$ approximation algorithm for this problem.

In order to demonstrate the non-triviality of a logarithmic approximation factor, we start by showing that a naive Minimum Spanning Tree (MST) heuristic may lead to an $O(n^{0.5})$ approximation factor (see Fig. 6). Assume we have n points on a square uniform grid. Its MST may have a path like shape, with $\Omega(n)$ radius. Hence its budget radius is $\Omega(n^2)$. On the other hand, each of the n points may be connected to the center with a path of length $O(n^{0.5})$. Hence, the budget radius of this metric is $O(n^{1.5})$.

5.1. Hardness results

In this section we present a reduction showing that finding the optimal budget allocation for radius budget problem on general metric graphs is an NP-hard problem. We prove the following theorem:

Theorem 5.1. *For complete metric graphs, the decision version of the rooted budget radius problem is NP-hard, and its minimization version is APX-hard.*

Before proving Theorem 5.1, we require a preliminary lemma and technical observation:

Lemma 5.2. *Let $G = (V, E)$ be a star graph with $V = \{r, s, v_1, \ldots, v_k\}$. Let s be the center of the star, and let edge $e = (r, s)$ have length $\ell(e) = 1$ and the k edges $e_i = (s, v_i)$ (for $i = 1, \ldots, k$) have length $\ell(e_i) = x$. Then in the optimal solution to the rooted budget radius problem on G the radius equals $(1 + \sqrt{xk})^2$.*

Proof. First note that for the budget radius problem on the subgraph rooted at s and excluding r, the optimal solution for a budget of 1 assigns each edge equal weight, and has a budget radius of kx. Then the result follows from Lemma 3.2 scaled down by the larger budget B. □

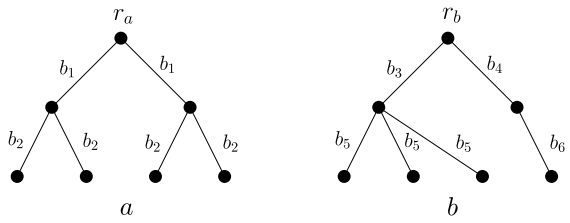

Fig. 7. The radius budget of the right tree (b) is smaller than the left tree (a).

Observation 5.3. *Consider Fig. 7. The optimal allocations for the left tree (a) and the right tree (b) are not the same; the right tree has a smaller budget radius. In fact, the optimal allocation of tree (a) can be used on the right tree (b) leading to the same radius (allocating to $b_3 = b_4 = b_1$ and $b_5 = b_6 = b_2$). The optimal budget allocation for both trees is presented in Fig. 1 cases: D, E).*

We now return to the proof of Theorem 5.1:

Proof. The theorem is proved via reduction from Set Cover, in particular the version in which each set consists of three elements, while an element appears in exactly two sets. This version of Set Cover is NP-hard, and its minimization version is APX-hard [6]. In the reduction, we will assume that in the minimum radius problem the target radius is fixed to 1 and the goal is to find the minimum cost B – by scaling, this is equivalent to the scenario where B is fixed as the goal is to minimize the radius.

Given a set cover instance (S, E) of sets and elements, we create a graph G as follows. First create root node n_r. For each element e_i in the set cover instance, create a single node n_{e_i}. We call these nodes *element-nodes*. For each 3-Set $S_j = \{e_a, e_b, e_c\}$, we create seven nodes called *set-nodes*. Choose x so that $(1+\sqrt{x})^2 = 2.1$ ($x \sim 0.202$), y so that $(1+\sqrt{2y})^2 = 3.1$ ($y \sim 0.289$), and z so that $(1+\sqrt{3z})^2 = 4.1$ ($z \sim 0.350$).

1. Set-node n_{S_j,e_a} is connected to root n_r by an edge of length 1, and to element-node n_{e_a} by an edge of length x. Set-nodes n_{S_j,e_b} and n_{S_j,e_c} are similarly connected to n_r, and respectively to element-nodes n_{e_b} and n_{e_b}. These set-nodes will represent the solution case where set S_j is needed to cover only a single element.
2. Set-node n_{S_j,e_a,e_b} is connected to root n_r by an edge of length 1, and to element-nodes n_{e_a} and n_{e_b} by edges of length x. Set-nodes n_{S_j,e_a,e_c} and n_{S_j,e_b,e_c} are similarly connected to n_r, and respectively to element-node pairs n_{e_a} and n_{e_c} or n_{e_b} and n_{e_c}. These set-nodes will represent the solution case where set S_j is needed to cover exactly two elements.
3. Set-node n_{S_j,e_a,e_b,e_c} is connected to root n_r by an edge of length 1, and to each of element-nodes $n_{e_a}, n_{e_b}, n_{e_c}$ by an edge of length z. This set-node represents the solution case where set S_j is needed to cover all three elements.

Complete the metric graph by adding all missing edges. The length of each added edge is simply the distance between its endpoints on the original graph.

Recall that we wish to achieve weighted radius 1, and we will show that finding the minimum cost B achieving this radius on G is equivalent to solving 3-Set Cover on the input instance. Now in any optimal solution, the following hold:

- For each set-node, weight is assigned to the edge connecting it to the root, and no weight is assigned to edges connecting it to other set-nodes.
- For each element-node, weight is assigned to an edge connecting it a single set-node, and to no other edges incident on the element-node.

For the first item, some weighted path must exist from the root to the set-node, while the edge from the root to the set-node is shorter than all other edges incident to the set-node. So an optimal solution places weight on that edge. Since an optimal solution has no cycles (Lemma 3.1), it places no weight on edges connecting set-nodes.

For the second item, first note that an element-node cannot be connected to more than one set-node, as then the graph would contain a cycle. An element node cannot be connected to another element-node, since the edge connecting two element-nodes is longer than the edge connecting the element-node to the root. And in fact, the edge connecting the element-node to the root cannot be used in the optimal solution: Let the element-node be n_{e_a} and consider some set-node n_{S_j,e_a}. Then separately connecting the element-node and set-node to the root incurs a cost of at least $(1+x)+1 \sim 2.202$, while by Lemma 5.2 connecting the element-node to the set-node incurs a cost of only $(1+\sqrt{x})^2 = 2.1$.

Given a sub-optimal solution that does not obey the above items, we can easily create a solution that does: First, weight placed on edges connecting two set-nodes is transfered evenly to the two edges connecting the set-nodes to the root, and this cannot increase the weighted radius. Second, the weight on all edges incident on an element-node is shifted to the path that achieves the smallest weighted radius, and this cannot increase the weighted radius. Henceforth, we will assume that any solution we consider obeys the above two items. Hence, we assume that the weighted edges of any solution form a tree, itself a set of star-graphs connected at the root.

Now let the set of set-nodes for S_j be N_j. Consider a set N_j of set-nodes for set S_j, and all edges incident on this set. Suppose there are exactly c element-nodes whose minimum paths are routed through set-nodes of N_j. Clearly $0 \leq c \leq 3$.

1. If $c = 0$, then weight is assigned only to the length 1 edges connecting the seven set-nodes of N_j directly to the root, at a total cost of 7.
2. If $c = 1$, then since $x < y < z$, weight is assigned to the length x edge connecting the relevant element-node n_{e_a} to set-node n_{S_j,e_a}. Consider the star graph formed by nodes $n_r, n_{S_j,e_a}, n_{e_a}$: By Lemma 5.2, to achieve a path length of 1 the total weight assigned to the two graph edges is $(1+\sqrt{x})^2 = 2.1$. The other six edges connecting n_r to N_j are each assigned weight 1, for a total cost of 8.1.
3. If $c = 2$, the optimal solution will assign weight to the length y edges connecting the relevant element-nodes n_{e_a}, n_{e_b} to set-node n_{S_j,e_a,e_b}. Consider the star graph formed by nodes $n_r, n_{S_j,e_a,e_b}, n_{e_a}, n_{e_b}$: By Lemma 5.2, to achieve path lengths of 1 the total weight assigned to the three graph edges is $(1+\sqrt{2y})^2 = 3.1$. The other six edges connecting n_r to N_j are each assigned weight 1, for a total cost of 9.1.
   Another option would have been to assign weight to the two edges connecting n_{e_a} to n_{S_j,e_a} and n_{e_b} to n_{S_j,e_b}. In this case, the total cost is $5 + 2(1+\sqrt{x})^2 = 9.2 > 9.1$, so this assignment is suboptimal.
4. If $c = 3$, the optimal solution will assign weight to the length z edges connecting element-nodes $n_{e_a}, n_{e_b}, n_{e_c}$ to set-node n_{S_j,e_a,e_c}. Consider the star graph formed by nodes $n_r, n_{S_j,e_a,e_b,e_c}, n_{e_a}, n_{e_b}, n_{e_c}$: By Lemma 5.2, to achieve path lengths of 1 the total weight assigned to the three graph edges is $(1+\sqrt{3z})^2 = 4.1$. The other six edges connecting n_r to N_j are each assigned weight 1, for a cost of 10.1.
   Another option would have been to assign weight to three edges connecting n_{e_a} to n_{S_j,e_a}, n_{e_b} to n_{S_j,e_b}, and n_{e_c} to n_{S_j,e_c}. In this case, the total cost is $4 + 3(1+\sqrt{x})^2 = 10.3 > 10.1$, so this assignment is suboptimal. If we would assign weight to the edges connecting n_{e_a} to n_{S_j,e_a} and both n_{e_b}, n_{e_c} to set-node n_{S_j,e_b,e_c}, then the total weight would be $5 + (1+\sqrt{x})^2 + (1+\sqrt{2y})^2 = 10.2 > 10.1$, so this assignment is suboptimal.

Recall that in the optimal solution, each element-node is incident on a single edge with non-zero weight, and this edge connects the element-node to a set-node. We will say that the set-node *covers* this element node. As above, for a given set N_j, at most one set-node in N_j covers some element-nodes. In this case we say that N_j is *used*, and that N_j covers these element-nodes.

We may then view the above cost assignment as follows: For each set N_j, if the set is not used we pay a single overhead cost of 7, and if the set is used we pay a overhead cost of $\frac{1}{10}$ plus a cost of 1 for each element-node covered by N_j. Then the total cost for the graph is an overhead charge of $7|S|+|E|$ plus a cost of $\frac{1}{10}$ for each used set. The minimum cost is achieved by using the minimum number of set-nodes to cover all element-nodes. This is equivalent to solving the Set Cover problem on the input. Since each element appears in at most two sets, $|S| \leq 2|E|$ and so $7|S| + |E| \leq 15|E|$. Since each set covers at most 3 elements, the number of used sets is between $\frac{|E|}{3}$ and $|E|$, and so the constant-factor inapproximability of the minimum Set Cover instance implies a constant-factor inapproximability for the minimization of the weighted radius. □

5.2. Approximation algorithm – budget radius on complete metric graphs

5.2.1. The special case of a line

We first consider a setup in which M is defined by some n points all residing on the interval $[0, 1]$, where for any two points p_1, p_2 within this interval, $d(p_1, p_2)$ is the Euclidean distance between p_1 and p_2. Let $G = (V, E)$ be the complete graph induced by M. We present a valid budget allocation $\mathcal{B}$ to E with budget radius at most $\log^2 n + 1$ and such that the graph induced by $\{e : \mathcal{B}(e) > 0\}$ is a tree spanning V.

Lemma 5.4. *Let $G = (V, E)$ be the complete graph described above, then* $\mathrm{BR}(G) \leq \log^2 n + 1$.

Proof. Let $P = \{p_1, p_2, \ldots, p_n\}$ be a set of n points on the interval $[0, 1]$ (in increasing order). Assume for the moment that n is a power of 2, and we construct a full binary tree T over P. Its root is $p_{\frac{n}{2}}$, and the root's children are $p_{\frac{n}{4}}$ and $p_{\frac{3}{4}n}$, etc.

The set of possible solutions for the budget radius problem for T is a subset of the set of possible solutions for the budget radius problem for G. Thus, it suffices to present a solution for T with budget radius at most $\log^2(n)$, that is, a pair of the form $(\mathcal{B}, r')$, where $\mathcal{B}$ is a valid budget allocation to E_T and $r' \in V_t$, such that, $\mathrm{wr}_{\mathcal{B}}(r') \leq \log^2(n)$. Clearly, fixing $r' = r$ only further restricts the set of solutions we allow. We next describe one such solution.

We first divide the edges of T into sets, defined by the depth of a given edge from the root r. Formally, if e is an edge in T, we say that e is at level i in T if one of its vertices has depth i and the other has depth $i+1$. We denote the set of all edges in T of level i by S_i. For every edge $e \in S_i$ we set $\mathcal{B}(e) = \alpha_e$, where $\alpha_e \stackrel{\text{def}}{=} \frac{1}{\log(n)} \frac{\ell(e)}{\sum_{e' \in S_i} \ell(e')}$. We first need to show that this allocation is valid and sums up to at most 1. This is true since for every level i, we divide a $\frac{1}{\log(n)}$ fraction of the budget among the edges in S_i. Since there are no more than $\log(n)$ levels, we do not exceed our budget.

To bound the budget radius of T under this allocation, observe that as T is a search tree, it holds for every i that $\sum_{e' \in S_i} \ell(e') \leq 1$. Thus, for every edge e of T, we have $\alpha_e \geq \frac{\ell(e)}{\log(n)}$. Now, let P be a simple path from r to some leaf ℓ. The weighted length of P (the weighted distance between r and ℓ) is $\sum_{e \in P} \omega_{\mathcal{B}}(e) = \sum_{e \in P} \frac{\ell(e)}{\alpha_e} \leq \sum_{e \in P} \log(n)$, which is at most $\log^2(n)$ since P consists of at most $\log(n)$ edges.

If n is not a power of 2, we can pad the set with points on the end of the interval, resulting in fewer than $2n$ total points, and the result follows. □

5.2.2. Complete metric graphs

We next define an approximation algorithm $\mathcal{A}$, such that given a complete graph $G = (V, E)$, induced by some metric space $M = (V, d)$, approximates the Budget Radius problem for G by a factor of $O(\log^2 n)$. Assume that a minimum spanning tree for G has a total weight LB, we proceed as follows:

1. Find an Hamiltonian path (*HP*) visiting all nodes with weight no more than $2 \cdot \text{LB}$.
2. Let G' be the result of unfolding *HP* to a straight line, i.e., G' is defined by n points, situated on an interval, such that, the distance between every two points is the length of the path between them on the Hamiltonian path *HP* (specifically, the length of the whole interval is exactly the length of *HP*).
3. Scale the above (*HP*) interval length to 1.
4. Build a balanced binary search tree (BT) over G'.
5. Apply the algorithm of Theorem 3.5 to BT. Assign the appropriate budget to all edges in BT and 0 to all other edges in E.

Theorem 5.5. *Let $G = (V, E)$ be a complete graph induced by some metric space $M = (V, d)$. Then algorithm $\mathcal{A}$ results in a valid budget allocation to the edges of E that approximates* $\text{BR}(G)$ *by a* $2(\log^2(n) + 1)$ *factor.*

Proof. First note that finding an Hamiltonian path (*HP*) with weight no more than $2 \cdot \text{LB}$ is feasible using an MST for G. More importantly, note that by Lemma 3.7, it holds that LB is a lower bound on the optimal solution (i.e., on $\text{BR}(G)$). This is true since an optimal budget allocation defines a tree (see Lemma 3.1), which has a total weight of at least LB (by the minimality of an MST). Thus, algorithm $\mathcal{A}$ yields an optimal budget allocation for BT, which by Lemma 5.4 yields a budget radius of at most $2 \cdot \text{LB} \cdot (\log^2(n) + 1) \leq 2 \cdot \text{BR}(G) \cdot (\log^2(n) + 1)$. □

6. Conclusion and future work

The paper introduces a new model for optimization problems on graphs. The suggested *budget* model was used to define facility location problems such as center and median point. For the tree case, optimal algorithms are presented for both aforementioned problems. For the metric center point problem, an $O(\log^2(n))$ approximation algorithm is presented. The new model raises a set of open problems e.g.: i) Complexity: there is still a considerable gap between the hardness result and the approximation factor in the complete metric graphs case of budget radius. ii) Facility location: Find approximation algorithms for the k-center and k-median problems on general graphs. iii) Graph optimization: minimizing the diameter of the budget graph. iv) Cost functions: It would be interesting to investigate the cost function $w(e) = \frac{\ell(e)}{\mathcal{B}(e)^a}$ for $0 < a < 1$. The general approach we presented for trees carries over to this setting as well, although our approximation algorithm for complete metric graphs (Lemma 5.4) does not.

References

[1] B. Ben-Moshe, M. Elkin, E. Omri, Optimizing budget allocation in graphs, in: CCCG, 2011.
[2] A.M. Campbell, T.J. Lowe, L. Zhang, Upgrading arcs to minimize the maximum travel time in a network, Network 47 (2) (2006) 72–80.
[3] Paz Carmi, Lilach Chaitman-Yerushalmi, Unexplored Steiner ratios in geometric networks, in: Computing and Combinatorics, Springer, 2012, pp. 275–286.
[4] V. Chepoi, H. Noltemeier, Y. Vaxès, Upgrading trees under diameter and budget constraints, Networks 41 (1) (2003) 24–35.
[5] V. Chepoi, Y. Vaxès, Augmenting trees to meet biconnectivity and diameter constraints, Algorithmica 33 (2) (2002) 243–262.
[6] M. Chlebík, J. Chlebíková, Inapproximability results for bounded variants of optimization problems, in: A. Lingas, B.J. Nilsson (Eds.), Fundamentals of Computation Theory, in: Lecture Notes in Comput. Sci., vol. 2751, Springer, Berlin Heidelberg, 2003.
[7] Ivan Contreras, Elena Fernández, General network design: a unified view of combined location and network design problems, European J. Oper. Res. 219 (3) (2012) 680–697.
[8] M.S. Daskin, Network and Discrete Location: Models, Algorithms, and Applications, Wiley-Interscience, 1995.
[9] Sven O. Krumke, Madhav V. Marathe, Hartmut Noltemeier, R. Ravi, S.S. Ravi, Approximation algorithms for certain network improvement problems, J. Comb. Optim. 2 (3) (1998) 257–288.
[10] S.T. McCormick, C. Li, D. Simchi-Levi, On the minimum-cost-bounded diameter and the fixed-budget-minimum-diameter edge addition problems, Oper. Res. Lett. 11 (1992) 303–308.
[11] Yimin Lin, Kyriakos Mouratidis, Best upgrade plans for large road networks, in: Advances in Spatial and Temporal Databases, Springer, 2013, pp. 223–240.
[12] Yimin Lin, Kyriakos Mouratidis, Best upgrade plans for single and multiple source-destination pairs, GeoInformatica 19 (2) (2015) 365–404.
[13] N. Megiddo, The weighted Euclidean 1-center problem, Math. Oper. Res. 8 (4) (1983) 498–504.

[14] Sanjay Melkote, Mark S. Daskin, An integrated model of facility location and transportation network design, Transp. Res., Part A, Policy Pract. 35 (6) (2001) 515–538.
[15] G. Narasimhan, M. Smid, Geometric Spanner Networks, Cambridge University Press, 2007.
[16] Kali P. Nepal, Dongjoo Park, Chang-Ho Choi, Upgrading arc median shortest path problem for an urban transportation network, J. Transp. Eng. 135 (10) (2009) 783–790.
[17] S. Nickel, J. Puerto, Location Theory: A Unified Approach, Springer, 2005.
[18] Doowon Paik, Sartaj Sahni, Network upgrading problems, Networks 26 (1) (1995) 45–58.
[19] A. Tamir, Improved complexity bounds for center location problems on networks by using dynamic data structures, SIAM J. Discrete Math. 1 (3) (1988) 377–396.
[20] J.Z. Zhang, X.G. Yang, M.C. Cai, A network improvement problem under different norms, Comput. Optim. Appl. 27 (3) (2004) 305–319.
[21] Li Zhang, Upgrading arc problem with budget constraint, in: Proceedings of the 43rd Annual Southeast Regional Conference – Volume 1, ACM-SE 43, ACM, 2005, pp. 150–152.