Optimizing Budget Allocation in Graphs

Boaz Ben-Moshe* Michael Elkin† Eran Omri‡

Abstract

In a classical facility location problem we consider a graph G with fixed weights on the edges of G. The goal is then to find an optimal positioning for a set of facilities on the graph with respect to some objective function. We consider a new model for facility location problems, where the weights on the graph edges are not fixed, but rather should be assigned. The goal is to find the valid assignment for which the resulting weighted graph optimizes the facility location 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 work in linear time, both for the case that 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 an $O(\log^2(n))$ approximation algorithm for the center point problem over general metric spaces.

1 Introduction

A typical facility location problem has the following structure: the input includes a weighted set D of demand locations, a set F of feasible facility locations, and a distance function d that measures the cost of travel between a pair of locations. For each $F' \subseteq F$, the quality of F' is determined by some underlying objective function (obj). The goal is to find a subset of facilities $F' \subseteq F$, such that $obj(F')$ is optimized (maximized or minimized). One important class of facility location problems is the *center point*, in which the goal is to find one facility in F, that minimizes the maximum distance between a demand point and the facility. Henceforth, we refer to this distance as *graph radius*. In another important class of problems, *graph median*, the goal is to find the facility in F that minimizes the average distance (i.e., the sum of the distances) between a demand point and the facility. In this paper we consider a new model for facility location on graphs, for which both problems are addressed.

*Department of Computer Science, Ariel University Center of Samaria, Ariel 40700, Israel, benmo@g.ariel.ac.il

†Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel, elkinm@cs.bgu.ac.il

This research has been supported by the Binational Science Foundation, grant No. 2008390.

‡Department of Computer Science, Bar-Ilan University, Ramat-Gan, 52900, Israel, omrier@gmail.com

1.1 The New Model

This paper suggests a new model for budget allocation problems on weighted graphs. The new model addresses optimization problems 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 have been studied extensively [4, 6, 8, 9]. Yet, in most cases the input for such problems consists of a given (fixed) graph. Motivated by well-known budget optimization problems [1, 3, 5, 10] 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 on the budget assigned to it. In other words, paying more for a communication link decreases its delay time.

More formally, we consider the following model: Let $G =< V, E >$ be an undirected graph induced by some length function $\ell(e)$ for each $e \in E$. Let B be a positive budget value. Allocating a budget $\mathcal{B}(e)$ to edge $e \in E$ with length $\ell(e)$ implies that the resulting weight of e is $\frac{\ell(e)}{\mathcal{B}(e)}$. Given this weight function and a special node (root) $r \in V$, the *rooted budget radius* problem can be defined as follows: Divide B among the edges of E in a way that the radius of (G, ω) with respect to r is minimized, where the weight function $\omega(e)$ is given by $\frac{\ell(e)}{\mathcal{B}(e)}$, for $\mathcal{B}(e) > 0$, such that $\sum_{e\in E} \mathcal{B}(e) = B$. In the *unrooted budget radius* problem the goal is to divide the budget B among the edges of E in a way that the radius of (G, ω) with respect to some vertex r is minimized, where $\omega(e) = \frac{\ell(e)}{\mathcal{B}(e)}$, $B(e) > 0$, and $\sum_{e\in E} \mathcal{B}(e) = B$.

Analogously, one can ask to minimize the *diameter* of (G, ω), defined as the maximum distance between a pair of vertices in (G, ω).

We also define the *median radius* of the graph (G, ω) with respect to a designated vertex r, denoted $\mathrm{MR}((G, \omega), r)$, as the average distance $\frac{1}{n} \cdot \sum_{v\in V} \mathrm{DIST}_{(G,\omega)}(r, v)$ between r and other vertices of the graph ($\mathrm{DIST}_{(G,\omega)}(r, v)$ represents the distance between vertices r and v in the graph (G, ω)). The vertex

r with the smallest median radius, i.e., $\mathrm{MR}((G,\omega),r) = \min_{v\in V}\mathrm{MR}((G,\omega),v)$ is called the *median* of (G,ω). The *budget median radius* of (G,ℓ) with respect to a designated vertex r and budget B, denoted $\mathrm{BMR}(G,r)$, is the minimum median radius of $(G,\omega=\frac{\ell}{\mathcal{B}})$ with respect to r, taken over all possible budget allocations $\mathcal{B}(\cdot)$. The *budget median radius* of (G,ℓ) with respect to budget B is the minimal budget median radius of (G,ℓ) with respect to some vertex v and budget B. Finally, the vertex r that realizes the budget median radius, i.e., such that $\mathrm{BMR}((G,\ell),r) = \min_{v\in V}\mathrm{BMR}((G,\ell),v)$ is called the *budget median* of the graph (G,ℓ).

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 (between the two nodes). 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 like, 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 the average delay time using a fixed budget.

1.3 Related Work

The problems of Center Point, Median Point on graphs (networks) have been studied extensively, see [4, 8] for a detailed surveys 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 [1, 2, 5, 10]. Spanner graph problems [7] 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. See [7] for a detailed survey on spanners. Observe that both *network improvement* and *spanner graph* problems can be modeled as a discrete version of our suggested new model.

1.4 Our Results

In this paper we present linear time algorithms for rooted and unrooted budget radius and budget median problems on trees. We also devise an $O(\log^2(n))$ approximation algorithm for the budget radius problem on general metric spaces.

1.5 Definitions

Let $G =< V,E >$ be a graph with some length $\ell(e)$ for each edge $e \in E$. We next introduce some definitions and notations to define the setting of the *budget radius* problem on graphs. We consider both the case where a candidate center node to the graph is given and an optimal budget allocation is sought, and the more general case where finding the center node yielding an optimal solution is part of the problem (as well as seeking an optimal budget allocation given such a center node). To simplify our notation we omit G from the notation whenever it is clear from the context.

Let $E = \{e_1,\ldots,e_{|E|}\}$. A *valid budget allocation* $\mathcal{B}(\cdot)$ to E is a non-negative real function, such that $\sum_{e\in E}\mathcal{B}(e) = 1$ (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). We denote $b_i \stackrel{\mathrm{def}}{=} \mathcal{B}(e_i)$, 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 $\omega_{\mathcal{B}}(e) \stackrel{\mathrm{def}}{=} \frac{\ell(e)}{\mathcal{B}(e)}$.

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

Table 1 includes the notations used in this paper.

Notation	Explanation
$G =< V,E >$	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 $e \in E$.
$\mathcal{B}=\{b_1,\ldots b_{\vert E\vert}\}$	an alternative notation for the function $\mathcal{B}$.
$\omega(e) = \frac{\ell(e)}{\mathcal{B}(e)}$	the (budget implied) weight of $e \in E$.
$G(\mathcal{B})$	the *budget-graph* implied by an allocation $\mathcal{B}$
$\delta_{\mathcal{B}}(u,v)$	the distance between two vertices in $G(\mathcal{B})$.

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

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 $\mathrm{wr}_{\mathcal{B}}(r) = \mathrm{wr}_{\mathcal{B}}(G,r) \stackrel{\mathrm{def}}{=} \max_{v\in V}(\delta_{\mathcal{B}}(r,v))$.

Given a graph $G =< V,E >$ (induced by some metric) and a node $r \in V$, we define the following:

An optimal allocation for (G,r): a valid allocation for which the weighted radius from r is minimized. There may be several optimal allocations. We denote an arbitrary optimal allocation by $\mathcal{B}^*_r = \mathcal{B}^*_r(G)$ and refer to it as *the optimal allocation* for (G,r).

The budget radius of G with center r: denoted

$\mathrm{BR}(r) = \mathrm{BR}(G, r)$, is the weighted radius of G with center r with the optimal allocation for G and r, i.e., $\mathrm{BR}(r) = \mathrm{wr}_{\mathcal{B}^*_r}(G, r)$.

The budget radius of G: $\mathrm{BR} = \mathrm{BR}(G) \stackrel{\text{def}}{=} \min_{v \in V} \mathrm{BR}(G, v)$.

An optimal allocation for G: a pair $(\mathcal{B}^*, r^*)$, where $\mathcal{B}^*$ 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 the following toy-example in Figure 1.5.

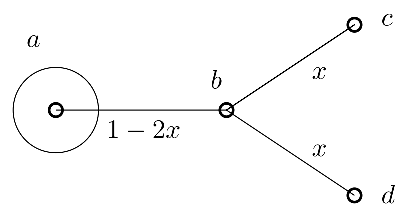


Figure 1: A simple example of a budget graph problem on a tree with a given center (a). Assume that each edge e has length $\ell(e) = 1$, and let x denote the fraction of the budget assigned to each of the edges (b, c) and (b, d). Observe that in order to have a valid budget allocation, it must hold that $0 < x < \frac{1}{2}$. The optimal solution minimizes the following function: $f(x) = \frac{1}{1-2 \cdot x} + \frac{1}{x}$. Note that in cases where x equals $\frac{1}{3}$ or $\frac{1}{4}$ the radius is 6, while the optimal allocation of x is approximately 0.293, and the radius is approximately 5.828.

Due to space limitations some of the proofs are omitted from this extended abstract and will appear in the full version of the paper.

2 The Budget Radius Problem for Trees

Given a connected graph $G =< V, E >$ with a length function on E, one may consider any subgraph G' of G, induced by some subset $E' \subseteq E$ and look for an optimal budget allocation for G' (i.e., $\mathrm{BR}(G')$). In particular, the class of trees is an important set of such subgraphs.

Lemma 1 *An optimal budget allocation (with respect to the budget radius problem)* (B^*, r^*) *for* G*, has the property that* $G(\mathcal{B}^*) =< V, E_{\mathcal{B}^*} >$*, where* $E_{\mathcal{B}^*} = \{e_i \in E : b_i^* > 0\}$ *is a tree spanning* G*.*

Proof. Clearly, all vertices are connected to r^* in $G(\mathcal{B}^*)$. Assume towards a contradiction that $G(\mathcal{B}^*)$ contains a cycle. Let T_D be the tree of shortest paths obtained by invoking the Dijkstra algorithm on $G(\mathcal{B}^*)$ and r^*. Hence, there is a an edge $e_i \in E_{\mathcal{B}^*}$ (i.e., $b_i^* > 0$) such that e does not appear in any shortest path from r^* to any vertex in V. Thus, we can obtain a better budget allocation by (say, equally) dividing the budget portion allocated to e_i among all edges in T_D. This is a contradiction to the optimality of $\mathcal{B}^*$. □

We note that the above is not true for the problem of the minimum budget diameter of a graph (see figure 2).

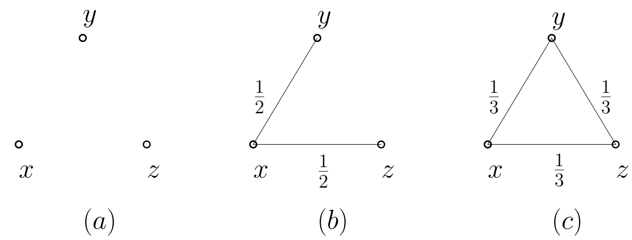


Figure 2: (a) Three points in the plane: x, y, z are the nodes of a unite equilateral triangle. (b) Tree: optimal radius (2), non optimal diameter (4). (c) Cycle graph: non-optimal radius (3), optimal diameter (3).

The above lemma suggests that it is interesting to consider the Budget Radius problem for the subclass of trees. In the sequel, 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. We use the standard terminology and refer to r as the root of the tree (rather than, the center). Thereafter, we consider the general case in trees, where the problem is to find a pair $(\mathcal{B}^*, r^*)$ minimizing the budget radius of the tree.

2.1 The Budget Radius for a Rooted Tree

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

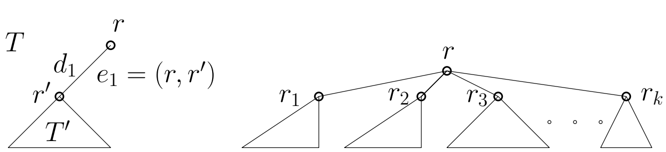
Lemma 2 *Let* T *be a tree rooted at* r*, with some length function* ℓ *on the edges of* T*. Assume* r *has a single child* r' *(the root of the subtree* T'*), and let* $R' = \mathrm{BR}(T', r')$ *and* $d_1 = \ell(r, r')$*. Then, an optimal budget allocation* $\mathcal{B}^*$ *assigns to the edge* $e_1 = (r, r')$ *a fraction* $b_1^* = \frac{\sqrt{d_1}}{\sqrt{R'} + \sqrt{d_1}}$*. It follows that* $\mathrm{BR}(T, r) = \frac{d_1}{b_1^*} + \frac{R'}{1 - b_1^*}$*.*

Proof. Let E be the set of edges in T and $E' = E \setminus \{e_1\}$ be the set of edges of T' (see Figure 2.1-a). Given any valid budget allocation $\mathcal{B}$ to E, let $\mathcal{B}'$ be the scaling of the restriction of $\mathcal{B}$ to E', defined by $\mathcal{B}'(e') = \frac{\mathcal{B}(e')}{1 - \mathcal{B}(e_1)}$ for every $e' \in E'$. Note that with this scaling, $\mathcal{B}'$ is a valid budget allocation to 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 $e_1 = (r, r')$, it follows that

$$\mathrm{wr}_{\mathcal{B}}(r) = \frac{d_1}{b_1} + \frac{\mathrm{wr}_{\mathcal{B}'}(r')}{1 - b_1}.$$

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 finding b_1 that minimizes the function $\mathrm{wr}_{\mathcal{B}}(r) = \frac{d_1}{b_1} + \frac{R'}{1-b_1}$. Therefore, it follows that $\mathrm{BR}(T, r) = \frac{d_1}{b_1} + \frac{R'}{1-b_1}$ for $b_1 = \frac{\sqrt{d_1}}{\sqrt{R'}+\sqrt{d_1}}$. $\square$

We next consider a more general tree structure (Figure 2.1-b). 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 the 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. Formally, $V'_i = V_i \cup \{r\}$, and $E'_i = E_i \cup \{(r, r_i)\}$. Clearly, the edges sets E'_is are 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.



(a) (b)

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

Lemma 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 $\delta_{\mathcal{B}^*}(r, \cdot)$). It then holds for all $1 \le i, j \le k$ that $\delta_{\mathcal{B}^*}(r, l_i) = \delta_{\mathcal{B}^*}(r, l_j)$.*

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, in turn, leads to 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 were assigned a ρ fraction of its current budget, i.e. $\rho \cdot \mathcal{B}^*(e)$ (while assignment to all other edges stays 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, $1 - (1 - \rho) \cdot \mathcal{B}^*(E'_j)$. Turning it into a valid budget allocation by dividing the remaining $(1-\rho)\mathcal{B}^*(E'_j)$ 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 $\mathcal{B}'(e) = \rho \cdot \mathcal{B}^*(e) + \frac{(1-\rho)\mathcal{B}^*(E'_j)}{|E|}$ if $e \in E'_j$, and $\mathcal{B}'(e) = \mathcal{B}^*(e) + \frac{(1-\rho)\mathcal{B}^*(E'_j)}{|E|}$ otherwise. We note that $\mathcal{B}'$ may not be an optimal allocation, however it is a contradiction to the optimality of $\mathcal{B}^*$. $\square$

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 4 *Let T be a tree as above. Then in any optimal budget allocation $\mathcal{B}^*$ to 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 $\mathrm{BR}(T, r) = \sum_{j=1}^{k} \mathrm{BR}(T'_j, r)$.*

Theorem 5 *Given a tree T rooted at r, it is possible to find an optimal valid budget allocation for T and r, in linear time in the size of T.*

Proof. T is a rooted tree, thus an inductive construction is only natural. 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, and no T'_i, T'_j for $i \ne j$ share any vertex other than r and E'_i, E'_j are disjoint for all $i \ne j$.

By Corollary 4, if we know $\mathrm{BR}(T'_i, r)$ for all $1 \le i \le k$, 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 2).

Note, that we evaluate the optimal solution for every subtree of every vertex in T exactly once and thus the procedure requires in linear time. $\square$

The following lemma proves helpful in the sequel, 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 6 *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., $\mathrm{BR}(T, r) \ge \sum_{e \in E} \ell(e)$.*

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 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 $(r_1 \dots r_k)$ and denote T_i the subtree induced by r and the vertices of the subtree of r_i. By Corollary 4 $\mathrm{BR}(T,r) = \sum_{i=1}^{k}\mathrm{BR}(T_i,r)$. Hence, by the induction hypothesis, the lemma follows. □

2.2 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 5 with every vertex v as a candidate center vertex r, and select the vertex v for which $\mathrm{BR}(T,v)$ is minimal as the ultimate center. However, this naive algorithm requires $O(n^2)$ time. We next show how to construct a linear time algorithm for this problem (indeed, for a tree T, our algorithm computes $\mathrm{BR}(T,v)$ for every v in T). Intuitively, this protocol uses the fact that given $\mathrm{BR}(T,r)$ and the partial computations made by algorithm of Theorem 5, applied to the T and r, it possible to compute in constant time $\mathrm{BR}(T,v)$ for every neighbor v of r. This intuition is formalized in Lemma 7.

Lemma 7 *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 Figure 2.2.*

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 4 that $\mathrm{BR}(T,v) = \mathrm{BR}(T_v,v) + \mathrm{BR}(\hat{T},v)$. By Lemma 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 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 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. Thereafter, we 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.

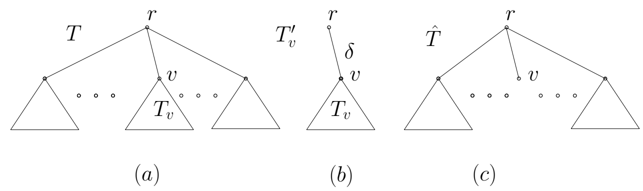


Figure 4: (a) The original tree rooted at r. (b) Considering v as the root of T'_v. (c) The tree $\hat{T}$.

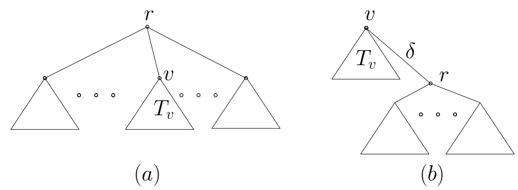


Figure 5: (a) The original tree rooted at r. (b) Considering v as the root of the tree. Computation of an optimal budget allocation for the tree rooted at v can be done in constant time, given an optimal budget allocation for the tree rooted at r.

Theorem 8 *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.*

3 Budget Radius – The General Case

In this section we consider the general case problem of optimizing the budget radius for a complete graph over n vertices, induced by some metric space $M = (V,d)$. We present an $O(\log^2(n))$ approximation algorithm for this problem. We start by showing that a naive Minimum Spanning Tree (MST) heuristic may lead to an $O(n^{0.5})$ approximation factor. 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 $O(n^{1.5})$.

3.1 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

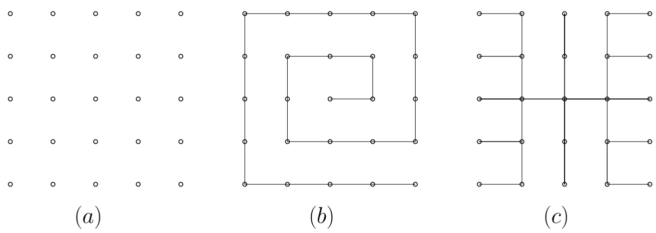


Figure 6: Using an MST-like heuristic may lead to an $O(n^{0.5})$ approximation ratio with respect to the *center point* problem (minimum radius). (a) A grid based set of points. (b) A path-like MST. (c) A solution with radius $O(n^{0.5})$.

valid budget allocation $\mathcal{B}$ to E with budget radius at most $\log^2 n$ and such that the graph induced by $\{e : \mathcal{B}(e) > 0\}$ is a tree spanning V.

Lemma 9 *Let $G = (V, E)$ be the complete graph described above, then* $\mathrm{BR}(G) \leq \log^2 n$.

3.2 General 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 $2 \cdot \mathrm{LB}$.
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 5 to BT. Assign the appropriate budget to all edges in BT and 0 to all other edges in E.

Theorem 10 *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* $\mathrm{BR}(G)$ *by a* $2\log^2(n)$ *factor.*

Proof. First note that finding an Hamiltonian path (HP) with weight no more than $2 \cdot \mathrm{LB}$ is feasible using an MST for G. More importantly, note that by Lemma 6, it holds that LB is a lower bound on the optimal solution (i.e., on $\mathrm{BR}(G)$). This is true since an optimal budget allocation defines a tree (see Lemma 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 9 yields a budget radius of at most $2 \cdot \mathrm{LB} \cdot \log^2(n) \leq 2 \cdot \mathrm{BR}(G) \cdot \log^2(n)$. $\square$

4 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 general 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) Hardness: is the budget center point problem on general graphs NP-hard[1]. ii) Facility location: Find approximation algorithms for the k-center, k-median, and 1-median on general graphs. iii) Graph optimization: minimizing the diameter of the graph.

References

[1] A. M. Campbell, T. J. Lowe, and L. Zhang. Upgrading arcs to minimize the maximum travel time in a network. *Netw.*, 47(2):72–80, 2006.

[2] V. Chepoi, H. Noltemeier, and Y. Vaxès. Upgrading trees under diameter and budget constraints. *Networks*, 41(1):24–35, 2003.

[3] Victor Chepoi and Yann Vaxès. Augmenting trees to meet biconnectivity and diameter constraints. *Algorithmica*, 33(2):243–262, 2002.

[4] M. S. Daskin. *Network and Discrete Location: Models, Algorithms, and Applications.* Wiley-Interscience, 1995.

[5] S. T. McCormick Li, C. L. and D. Simchi-Levi. On the minimum-cost-bounded diameter and the fixed-budget-minimum-diameter edge addition problems. *Operations Research Letters*, 11:303–308, 1992.

[6] N. Megiddo. The weighted Euclidean 1-center problem. *Math. Oper. Res.*, 8(4):498–504, 1983.

[7] G. Narasimhan and M. Smid. *Geometric Spanner Networks.* Cambridge University Press, 2007.

[8] S. Nickel and J. Puerto. *Location Theory: A Unified Approach.* Springer, 2005.

[9] A. Tamir. Improved complexity bounds for center location problems on networks by using dynamic data structures. *SIAM J. Discret. Math.*, 1(3):377–396, 1988.

[10] J. Z. Zhang, X. G. Yang, and M. C. Cai. A network improvement problem under different norms. *Comput. Optim. Appl.*, 27(3):305–319, 2004.

[1]The discrete version of the center point budget problem is weakly NP-complete even on a path like graph (reduced to *perfect partition problem*).

Appendix

Proof for Corollary 4

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, 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 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, $\hat{\mathcal{B}}$ is not optimal and hence, $\mathcal{B}^*$ is not optimal either – 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, 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 $\mathrm{BR}(T,r) = \sum_{j=1}^k \mathrm{BR}(T'_j, r)$. □

Proof for Theorem 8

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 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 7, 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. □

Proof for Lemma 9

Proof. Let $P = \{p_1, p_2, ..., p_n\}$ be a set of n points on the interval $[0,1]$ (in increasing order). Next, 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, $\mathrm{wr}_{\mathcal{B}}(r') \le \log^2(n)$. 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 $\alpha_e \overset{\text{def}}{=} \frac{1}{\log(n)} \frac{\ell(e)}{\sum_{e' \in S_i} \ell(e')}$. 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 $\sum_{e' \in S_i} \ell(e') \le 1$. Thus, for every edge e of T, we have $\alpha_e \ge \frac{\ell(e)}{\log(n)}$. 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 $\sum_{e\in P} \omega_{\mathcal{B}}(e) = \sum_{e \in P} \frac{\ell(e)}{\alpha_e} \le \sum_{e \in P} \log(n)$, which is at most $\log^2(n)$ since P consists of at most $\log(n)$ edges. □

Generalization to Median Point

In this section we generalize the center point algorithms to the *median point* case[2]. In this case we would like to find the node (M) and its corresponding budget allocation which minimizes the average (or sum) weight of all shortest path from M to all the graph nodes. Following the same framework as in the center point we now would like to find the optimal allocation for $\mathcal{B}^*$, for a fixed node r as a median point.

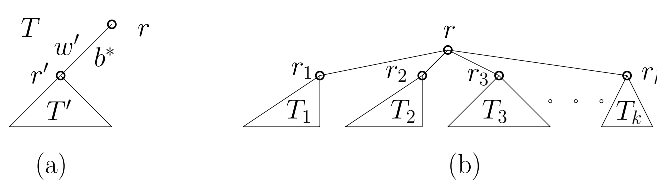


Figure 7: Following the same framework as in the center point we now would like to find the optimal allocation for $\mathcal{B}^*$. (a) Single subtree case: b^* minimizes the value of: $\frac{sum(T',r')}{1-b} + n\frac{w'}{b}$, where $n = |T'|$. (b) Multiple subtrees case: can be solved as an LP minimization problem.

In case there is a single child to the root r (see Figure 4a) we need to minimize the following term: $sum(T,r) = \frac{sum(T',r')}{1-b^*} + n\frac{w'}{b^*}$, where $n = |T'|$. Observe that $sum(T,r) =$

[2]The complete version of the budget median problems on trees can be found in the full version of this paper.

$\frac{c_1+\frac{c_2}{x}}{1-x}$ where $c_1 = sum(T', r'), c_2 = nw'$ therefore $\mathcal{B}^*$ can be computed in constant time, and so is $sum(T, r)$.

If the root r has more than a single subtree (as shown in Figure 4b), we can apply the above computation on each subtree independently, calculating the optimal budget allocation for each edge $e_i = (r_i, r)$, then the budget between all subtrees can be normalized using the following minimization problem: $min(\frac{X_1}{B_1} + \frac{X_2}{B_2} + ... + \frac{X_k}{B_k})$ where $B_i > 0$, X_i is the optimal sum of T_i with the root r, and $B_1+B_2+...+B_k = 1$. This problem can be solved in $O(k)$ time.

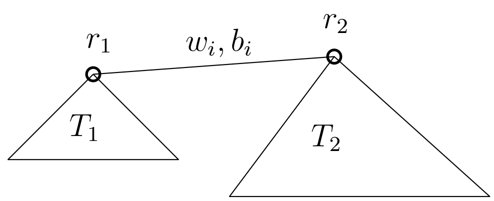


Figure 8: Assuming $|T_1| < |T_2|$ the median cannot be in T_1, regardless of w_i and b_i.

Corollary 11 *The location of the median in the unrooted tree case depends only on the tree structure, i.e., it is in the regular unweighted tree median node. This is due to the convex nature of the median see figure 4. Therefore, the optimal root for the median point can be found in linear time.*