CS 369M: Metric Embeddings and Algorithmic Applications 10/30/2018

Lectures 11-13: Tree Embeddings for Problems with Capacities

Lecturer: Moses Charikar *Scribe: Margalit Glasgow*

1 Introduction

In Lecture 9, we proved that any n-point metric can be embedded into a distribution on hierarchical well-separated tree metrics (HSTs) with distortion $O(\log(n))$ [2]. In today's lecture, we discuss how this result can be used to approximate not only shortest path *lengths* in graphs, but also *capacities* across cuts. This technique will ultimately yield $O(\log(n))$ approximation algorithms for the Oblivious Routing Problem and the Minimum Bisection Problem.

2 Two Motivating Problems

We begin by motivating this lecture with two main problems, Oblivious Routing and Minimum Bisection.

2.1 The Oblivious Routing Problem

Let $G = (V, E)$ be a graph with n vertices and symmetric capacities $c : E \to \mathbb{R}^+$. Given a set of demand pairs $d : V \times V \to \mathbb{R}^+$, we would like to route $d(u, v)$ units of flow between each demand pair (u, v) while minimizing the congestion, $\max_{e \in E} \dfrac{\text{flow}(e)}{c(e)}$. In the oblivious version of this problem, we have to commit to a predetermined flow route between each pair before learning what the demands are.

Ultimately we will show that it is possible to commit to flow routes such that for any set of demands, the congestion is no more than $O(\log(n))$ times the optimal congestion, i.e., compared to when we know the demands beforehand. We will accomplish this by committing to route a single unit of flow from u to v on a weighted combination of flow paths. Each path will be given by the single path from u to v in one tree in the support of our distribution, and weighted according to the probability of that tree in the distribution.

2.2 The Minimum Bisection Problem

Let $G = (V, E)$ be a graph with n vertices. We want to partition G into S and $\overline{S} := V \setminus S$ with $|S| = |V|/2$ so as to minimize the total capacity of edges between S and $\overline{S}$.

For this problem, we will show that we can achieve a polynomial-time $O(\log(n))$ approximation by choosing a bisection of the vertices that is optimal for one tree T in the support of the probabilistic tree embedding.

2.3 Analysis

Both of the algorithms above use the following generic procedure:

1. Create a probabilistic embedding from G to a distribution over trees, which preserves some notion of cut capacities up to a ρ-approximation, formally defined later (see Definition 3.2 and Corollary 5.2).

2. Given a problem instance, solve it optimally on each tree T in the support of the distribution, producing a solution OPT_T. By exploiting the simple structure of trees, finding an optimal solution to NP hard problems can often be done in polynomial time.

3. Map the optimal tree solutions to a solution on G, creating a feasible solution A_G.

Our analysis will rely on the fact that because our distribution over trees preserves cut capacities up to a factor of ρ, the feasible solution A_G only looses this factor when compared to the optimal solution on G. The specifics of this analysis will depend on the problem we are solving, but the main ingredients are in Lemma 4.3 and Lemma 4.4, which bound the congestion of flows mapped between decomposition trees and G.

3 Mappings between Trees and Graphs

In this section, we describe how to map edges from a graph G to a decomposition tree T (defined below) and back to G. This mapping will in turn define a way to maps flows from G to T and back to G, which will be useful in the analysis of our approximation algorithms.

Let $G = (V, E)$ be a graph with capacities given by $c : E \to R^+$. and lengths given by $\ell : E \to R^+$. A *decomposition tree* $T = (V_T, E_T)$ is a rooted tree whose leaves correspond to the vertices of G. The tree T may have additional vertices. For clarity, we will index edges and vertices in T by the subscript t.

Any decomposition tree T induces the following four mappings:

1. A map $P : E \to E_T^*$ which maps a graph edge (u, v) to the unique tree path from leaf u to leaf v in T.

2. A vertex mapping $m_V : V_T \to V$ maps a vertex v_t in V_T to an arbitrary leaf in the subtree rooted at v_t.

3. An edge mapping $m_E : E_T \to E^*$ maps edges in T to paths in G. For any edge $e_t = (u_t, v_t) \in E_T$, $m_E(e_t)$ will be some arbitrarily chosen shortest path between $m_V(u_t)$ and $m_V(v_t)$.

4. A mapping $M : E \to E^*$ from each edge $e_i = (u, v)$ in E to multisets of E. This mapping is defined as the union of all paths $m_E(e_t)$ for each edge e_t in the path $P(e_i)$ from leaves u to v in T. We will represent this mapping M by a matrix, where M_{ij} is the number of times e_j lies on a path $m_E(e_t)$ for e_t in the unique simple tree path from u to v. Formally,

$$M_{ij} = \sum_{e_t \in P(e_i)} \mathbb{1}_{e_j \in m_E(e_t)}. \tag{1}$$

Example 1. Consider the graph G and a decomposition tree, T in Figure 1. We have labeled each vertex v_t of T with the label $m_V(v_t)$. Note that the values on the edges are their labels, not lengths or capacities.

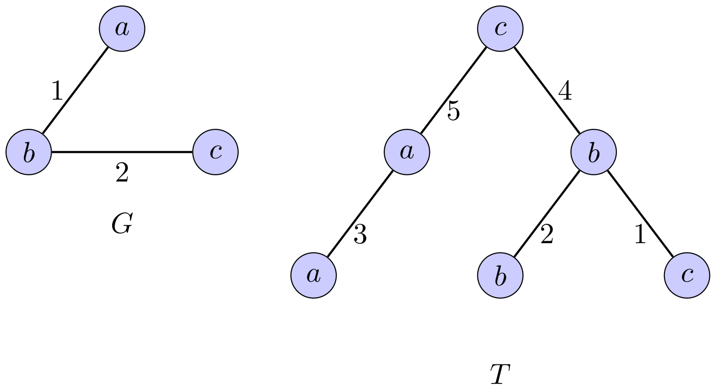

Figure 1: A graph and decomposition tree

The following matrix represents the mappings m_E for each edge of the tree:

	1	2
1	0	1
2	0	0
3	0	0
4	0	1
5	1	1

The follow matrix represents the mapping M,

	1	2
1	1	0
2	2	1

Let $\mathcal{M}$ be the set of admissible mappings, that is, the set of mappings induced by some decomposition tree of G.

A probabilistic mapping is given by weights $\lambda_M \geq 0$ for each mapping $M \in \mathcal{M}$ where $\sum \lambda_M = 1$.

Remark. A probabilistic mapping can equivalently be defined by a distribution $\{\lambda_T\}$ over decomposition trees, where for each mapping M induced by a tree T, we put $\lambda_M = \lambda_T$. In the rest of these notes, we will sometimes use the weights $\{\lambda_T\}$ and $\{\lambda_M\}$ interchangeably.

We are now ready to define capacity distortion (*relative load*) of an probabilistic mapping, in analogy with our previous notion of length distortion, or *stretch* of a probabilistic embedding:

Capacity : Length :: Relative Load : Stretch

The stretch of an edge is a measure of how much the length of an edge $(u, v) \in E$ is stretched when it is mapped by M to a multiset of E.

Definition 3.1 (Distance Mapping)**.** For any edge $e_j \in E$, define

$$\text{dist}_M(e_i) = \sum_j M_{ij}\ell_j.$$

and

$$\text{stretch}_M(e_i) = \frac{\text{dist}_M(e_i)}{\ell_i}.$$

The average stretch of a probabilistic mapping is

$$\mathbb{E}[\text{stretch}_M(e_i)] = \sum \frac{\lambda_M \text{dist}_M(e_i)}{\ell_i}.$$

Given a distribution over mappings, the *stretch* of the distribution is the maximum over all edges of the average stretch of an edge.

Definition 3.2. [Capacity Mapping] For any edge $e_j \in E$, define

$$\text{load}_M(e_j) = \sum_i M_{ij} c_i.$$

Define the relative load

$$\text{rload}_M(e_j) = \frac{\text{load}_M(e_j)}{c_j}.$$

The average relative load of a probabilistic mapping is

$$\mathbb{E}[\text{rload}_M(e_j)] = \sum \frac{\lambda_M \text{load}_M(e_j)}{c_j}.$$

Given a distribution over mappings, the *relative load* of the distribution is the maximum over all edges of the average relative load of an edge.

In our analysis of our approximation algorithms, it will be convenient to map flows from G to a decomposition tree T and vice versa. We will do this by mapping the flow on each edge e in G to a flow of that same value on the entire path $P(e)$ in T associated with that edge. A flow in T can be mapped to a flow in G by mapping the flow on each edge e_t of T to a flow of the same value on the path $m_E(e_t)$ in G.

Formally, given a multi-commodity flow $f : E \to R^+$ on G, and a decomposition tree T with associated mappings P, m_V, m_E, and M, define the linear mapping $m' : R^{|E|} \to R^{|E_T|}$ by

$$m'(f)_{e_t} = \sum_{e \in E : e_t \in P(e)} f_e.$$

Similarly, define the linear mapping $m : R^{|E_T|} \to R^{|E|}$ by

$$m(f)_e = \sum_{e_t \in E_T : e \in m_E(e_t)} f_{e_t}.$$

It is easy to check that

$$m(m'(f)) = Mf.$$

We also need to define capacities $c' : E_T \to R^+$ in each decomposition tree so that we can measure the congestion of a flow in T. For a tree edge $e_t = (v_t, u_t)$, define

$$c'(e_t) := \sum_{e \in E : e_t \in P(e)} c_e. \tag{2}$$

4 Analysis

In this section we will prove the following two theorems.

Theorem 4.1. *Given a graph G and distribution $\{\lambda_M\}$ over $\mathcal{M}$ with average relative load ρ, we can find a predetermined routing which, given any set of demands, achieves at most ρ times the congestion of the optimal routing of the demands in retrospect.*

Theorem 4.2. *Given a graph G and a distribution $\{\lambda_M\}$ over $\mathcal{M}$ with average relative load ρ, in polynomial time, we can find a bisection of weight at most ρ times the optimal bisection of G.*

In the next section, we will show that we can satisfy the hypothesis of these two theorems for $\rho = O(\log(n))$. The proof of these theorems rely on the next two lemmas:

Lemma 4.3. *For any flow f of congestion $C_G(f)$ in G, the congestion $C_T(m'(f))$ of $m'(f)$ in T is at most $C_G(f)$.*

Proof. This follows directly from our definition of tree capacities in Equation (2). For any tree edge e_t, we have

$$m'(f)_{e_t} = \sum_{e \in E : e_t \in P(e)} f_e \leq \sum_{e \in E : e_t \in P(e)} C_G(f) c_e = C_G(f) c'(e_t).$$

Hence for every tree edge, the congestion is no more than $C_G(f)$. □

Lemma 4.4. *Given a distribution $\{\lambda_T\}$ over decomposition trees, with average relative load ρ, for any flows f_T on each tree of congestion at most C, the congestion of the expected graph flow, $C_G(\mathbb{E}_T[m(f_T)])$, is at most $C\rho$.*

Proof. Given a flow f_T, we have $C_G(\mathbb{E}_T[m(f)]) = \max_e \frac{\mathbb{E}_T[m(f_T)_e]}{c_e}$. For every graph edge e_j and a fixed tree T, we have

$$m(f_T)_{e_j} = \sum_{e_t : e_j \in m_E(e_t)} f_{T e_t} \leq \sum_{e_t : e_j \in m_E(e_t)} C c'(e_t) = C \sum_{e_t : e_j \in m_E(e_t)} \sum_{e_i : e_t \in P(e_i)} c_i$$

By definition of the matrix M in Equation (1), we have

$$M_{ij} = \sum_{e_t \in P(e_i)} \mathbb{1}_{e_j \in m_E(e_t)} = \sum_{e_t : e_j \in m_E(e_t)} \mathbb{1}_{e_t \in P(e_i)},$$

so

$$m(f_T)_{e_j} = C \sum_i M_{ij} c_i = \mathrm{load}_M(e_j).$$

Hence for any edge,

$$\frac{\mathbb{E}_T[m(f_T)_e]}{c_e} = C \mathbb{E}_T[\mathrm{rload}_M(e_j)] \leq C\rho,$$

because the average relative load of our distribution of mappings is ρ. It follows that

$$C_G(\mathbb{E}_T[m(f)]) \leq C\rho.$$

□

We are now ready to prove our main results for oblivious routing and minimum bisection. We begin with oblivious routing, which is more straightforward.

Proof. (Theorem 4.1) We can construct a predetermined routing from the distribution over mappings in the following way. For each tree T in the support of the distribution of mappings, let $r_T(u, v)$ be the tree flow routing one unit of flow from the leaf u to the leaf v in T on the unique tree path from u to v. Now for any pair (u, v), to route one unit of flow from u to v in G, we will use the flow

$$r(u, v) = \sum_T \lambda_T m(r_T(u, v)).$$

For demands $\{d(u, v)\}$, the final flow f in G is given by

$$f = \sum_{(u,v)} d(u, v)r(u, v) = \sum_T \lambda_T m(f_T),$$

where

$$f_T = \sum_{(u,v)} d(u, v)r_T(u, v).$$

Suppose the optimal routing in hindsight, f^*, has congestion at most $C_G(f^*)$ in G. Recall that we want to show that

$$C_G(f) \leq \rho C_G(f^*).$$

For each tree T in the support of the distribution of mappings, let $f_T^* = m'(f^*)$, where m' is the mapping given by T. Then by Lemma 4.3, we have

$$C_T(f_T^*) \leq C_G(f^*). \tag{3}$$

Because f_T is the optimal way to route the demands in T, and f_T^* and f_T route the same demands, we have

$$C_T(f_T) \leq C_T(f_T^*). \tag{4}$$

Finally, by Lemma 4.4, combining Equation (3) and Equation (4), we have

$$C_G\left(\sum_T \lambda_T m(f_T)\right) = C_G\left(\mathbb{E}_T m(f_T)\right) \leq \rho C_G(f^*). \tag{5}$$

□

The approximation ratio for minimum bisection follows from a slightly more complicated argument.

Proof. (Theorem 4.2)

Given a distribution over tree mappings with average relative load at most ρ, we will produce a bisection in the following way. For each tree T in the support of this distribution[1], we compute the minimum cost leaf bisection $(S_T, \overline{S}_T)$. The cost of a leaf bisection $(S, \overline{S})$ in a tree is defined as the minimum capacity of edges that must be removed to separate S from $\overline{S}$. Computing the optimal (minimal cost) leaf bisection is possible in polynomial time using dynamic programming, see [3] for

[1]By a problem on the homework, the support of this distribution has size $O(n^2)$, see also [4].

example. We then output the minimum bisection on G given by one of the $(S_T, \overline{S}_T)$ which achieves the minimum capacity in G.

We will use the notation $\text{cost}_T(S, \overline{S})$ to denote the capacity of the leaf bisection $(S, \overline{S})$ in T, and $\text{cost}_G(S, \overline{S})$ to denote the cost of the bisection $(S, \overline{S})$ in G.

Let $(S^*, \overline{S}^*)$ be the optimal bisection in G. We want to show that there exists some T in the support such that

$$\text{cost}_G(S_T, \overline{S}_T) \leq \rho \text{cost}_G(S^*, \overline{S}^*). \tag{6}$$

Claim 4.5. *For any tree T, and any bisection $(S, \overline{S})$, we have*

$$\text{cost}_T(S, \overline{S}) \geq \text{cost}_G(S, \overline{S}).$$

Proof. Let s be a super-source in G, connected by edges of infinite capacity to each vertex in S, and let t be a super-sink in G, connected by edges of infinite capacity to each vertex in $\overline{S}$. Without exceeding edge capacities, we can send at most $\text{cost}_G(S, \overline{S})$ units of flow from s to t in G, because any flow must cross the cut from S to $\overline{S}$. Analogously adding a super-source s and super-sink t to T, by Lemma 4.3, we can send a flow of congestion at most 1 from s to t in T. Hence the cost of the bisection in the tree is at least $\text{cost}_G(S, \overline{S})$. □

To show the existence of a tree satisfying Equation (6), it suffices to prove the following claim:

Claim 4.6.

$$\mathbb{E}_T \left[\text{cost}_G(S_T, \overline{S}_T)\right] \leq \rho \text{cost}_G(S^*, \overline{S}^*).$$

Proof. First note that for any T, by Claim 4.5 and optimality of the bisection $(S_T, \overline{S}_T)$, we have

$$\mathbb{E}_T \left[\text{cost}_G(S_T, \overline{S}_T)\right] \leq \mathbb{E}_T \left[\text{cost}_T(S_T, \overline{S}_T)\right] \leq \mathbb{E}_T \left[\text{cost}_T(S^*, \overline{S}^*)\right] \tag{7}$$

By Claim 4.5, in each tree T, there exists a flow f_T of value $\text{cost}_T(S^*, \overline{S}^*)$ from S^* to $\overline{S}^*$ with congestion at most 1 in T.

We can map these flows f_T to a flow f of value $\mathbb{E}_T[\text{cost}_T(S^*, \overline{S}^*)]$ in G, where

$$f = \mathbb{E}_T[m(f_T)].$$

Then by Lemma 4.4, the congestion of f in G is at most ρ. Scaling down by ρ, we get a flow of congestion at most 1 in G and value $\dfrac{\mathbb{E}_T[\text{cost}_T(S^*, \overline{S}^*)]}{\rho}$.

It follows that any cut between S^* and $\overline{S}^*$ in G must have capacity at least $\dfrac{\mathbb{E}_T[\text{cost}_T(S^*, \overline{S}^*)]}{\rho}$, and hence

$$\frac{\mathbb{E}_T[\text{cost}_T(S^*, \overline{S}^*)]}{\rho} \leq \text{cost}_G(S^*, \overline{S}^*).$$

Combining this with Equation (7) yields the claim. □

Using Claim 4.6, we see that there must exist some tree T in the support such that

$$\text{cost}_G(S_T, \overline{S}_T) \leq \rho \text{cost}_G(S^*, \overline{S}^*).$$

By this claim, there must exist a tree T satisfying Equation (6), so our algorithm can find a bisection of cost at most ρ times the cost of the optimal bisection of G. □

5 Low Relative Load Probabilistic Mappings

In this section, we will prove the following result.

Theorem 5.1 (Räcke [4]). *For every family of admissible mappings $\mathcal{M}$, the following are equivalent:*

1. *For every collection of lengths, there exists a distribution $\{\lambda_M\}$ over mappings with stretch at most ρ.*

2. *For every collection of capacities, there exists a distribution $\{\lambda_M\}$ over mappings with relative load at most ρ.*

Remark. The distribution over tree mappings which yields low stretch may not be the same as the distribution which yields low distortion. For a discussion of distributions which yield low stretch *and* distortion, see Feige-Andersen [1], section 3.4.

A corollary of this theorem proves that our algorithms in the previous section achieve approximations of $O(\log(n))$:

Corollary 5.2. *For every collection of capacities, there exists a distribution over decomposition trees which induces a distribution $\{\lambda_M\}$ over mappings with relative load at most $O(\log(n))$.*

The proof of the corollary relies on the result of FRT:

Theorem 5.3 (FRT [2]). *For any graph $G = (V, E)$ with shortest path distances $d(u, v) \geq 1$, there exists some distribution over HSTs T, such that*

- *For every tree T in the support, $d(u, v) \leq d_T(u, v)$.*
- *On average, $\mathbb{E}_T[d_T(u, v)] \leq O(\log(n))d(u, v)$.*

Tree distances $d_T(u, v)$ are measured by the weight of the least common ancestor of u and v in T. The weight of any node is 8^h, where h is its height from the bottom of the tree. Each leaf has height 0, and each path from the root to a leaf is of equal length.

Proof. (Corollary 5.2) We will show how Theorem 5.3 implies the first statement in Theorem 5.1. For a graph G with lengths ℓ_i, scale the lengths so that $\ell_i \geq 1$. Let λ_T be the probability of the tree T in the distribution given by Theorem 5.3. For each T in the support, let M be the mapping induced by this tree, and choose $\lambda_M = \lambda_T$.

For any edge $(u, v) = e_i \in E$, let $h_T(e_i)$ denote the height of the least common ancestor of u and v in T.

Consider the unique path $P(e_i)$ in T from u to v which traverses exactly two edges $e_t^{h,1}$ and $e_t^{h,2}$ between every consecutive pair of heights $(h, h+1)$, $h < h_T(e_i)$. Let $(u_t^{h,k}, v_t^{h,k}) = e_t^{h,k}$ for $k \in \{1, 2\}$.

By definition of the distance mapping,

$$\text{dist}_M(e_i) = \sum_{e_t \in P} \sum_{e_j \in m_E(e_t)} \ell_j.$$

Summing over tree edges, we have

$$\text{dist}_M(e_i) = \sum_{h=0}^{h_T(e_i)-1} \sum_{k=1,2} \sum_{e_j \in m_E(e_t^{h,k})} \ell_j = \sum_{h=0}^{h_T(e_i)-1} \sum_{k=1,2} d(m_V(u_t^{h,k}), m_V(v_t^{h,k})).$$

Because $m_V(u_t^{h,k})$ and $m_V(v_t^{h,k})$ have a least common ancestor at height at most $h+1$, by the dominating property of the FRT embeddings, we have

$$d(m_V(u_t^{h,k}), m_V(v_t^{h,k})) \leq 8^{h+1}.$$

It follows that the average distance of e_i,

$$\mathbb{E}_M\left[\mathrm{dist}_M(e_i)\right] \leq \mathbb{E}_T\left[\sum_{h=0}^{h_T(e_i)-1} \sum_{k=1,2} d(m_V(u_t^{h,k}), m_V(v_t^{h,k}))\right] \leq \mathbb{E}_T\left[\sum_{h=0}^{h_T(e_i)-1} 2\cdot 8^{h+1}\right]$$
$$\leq \mathbb{E}_T\left[8^{h_T(e_i)+1}\right] = 8\mathbb{E}_T\left[d_T(u,v)\right] \leq O(\log(n))d(u,v) \leq O(\log(n))\ell_i.$$

It follows that the average stretch of e_i is

$$\mathbb{E}_M\left[\frac{\mathrm{dist}_M(e_i)}{\ell_i}\right] \leq O(\log(n)),$$

which proves that the probabilistic mapping given by $\{\lambda_M\}$ has stretch at most $O(\log(n))$. □

To prove Theorem 5.1, we will use the following two lemmas relating minimum stretch (resp. relative load) with a two player game.

Lemma 5.4. *For every family of admissible mappings $\mathcal{M}$, there exists a distribution over mappings with stretch at most ρ if and only if for every set of $\alpha_i \geq 0$ where $\sum \alpha_i = 1$, there exists an $M \in \mathcal{M}$ such that*

$$\sum \alpha_i \text{ stretch }_M(e_i) \leq \rho. \tag{8}$$

The following lemma is the analogous result for relative load.

Lemma 5.5. *For every family of admissible mappings $\mathcal{M}$, there exists a distribution over mappings with relative load at most ρ if and only if for every set of $\beta_j \geq 0$ where $\sum \beta_j = 1$, there exists a mapping $M \in \mathcal{M}$ such that*

$$\sum \beta_j \mathrm{rload}_M(e_j) \leq \rho \sum \beta_j.$$

The proof of these lemmas is application of von Neumann's minimax Theorem:

Theorem 5.6 (Minimax)**.** *Given a two player game where Player 1 chooses an allowed strategy M, and Player 2 chooses an allowed strategy i, where the payoff of the game is $p(i, M)$, we have*

$$\min_{\{\lambda_M\}:\sum \lambda_M=1} \max_i \sum_M \lambda_M\, p(i,M) \tag{9}$$
$$= \max_{\{\alpha_i\}:\sum \alpha_i=1} \min_M \sum_i \alpha_i\, p(i,M). \tag{10}$$

We call this value the value of the game.

Proof. (Lemma 5.4) Consider a zero sum game in which Player 1, MAP, is trying to find a good mapping, and Player 2, EDGE, tries to find an edge on which this mapping has high average stretch. The payoff $p(M, i)$ of this game is the stretch of e_i in mapping M.

Suppose there exists a distribution λ_M on $\mathcal{M}$ with stretch at most ρ. Then by Equation (9) the value of the game is at most ρ because for any edge, the average stretch is less than ρ. It follows by Theorem 5.6 that for any α_i where $\sum \alpha_i = 1$, there exists some mapping M such that the sum in Equation (8) is at most the value of the game, ρ. This yields one direction of the lemma.

Now suppose for every set α_i, Equation (8) holds. It follows that the value of the game is at most ρ, meaning the value in Equation (9) is at most ρ, and hence there exists a distribution of mappings λ_M with stretch at most ρ. □

Lemma 5.5 follows by the same reasoning.

We are now ready to prove Theorem 5.1.

Proof. (Theorem 5.1) We first show that the first statement implies the second. By Lemma 5.4, the first statement implies that for any set of lengths ℓ_i and α_i with $\sum \alpha_i = 1$, there exists a mapping M such that

$$\sum_i \alpha_i \frac{\mathrm{dist}_M(e_i)}{\ell_i} \leq \rho. \tag{11}$$

Recall that we want to show that for any β_i which sum to 1, we have

$$\sum_j \beta_j \frac{\mathrm{dist}_M(e_j)}{c_j} \leq \rho. \tag{12}$$

Given some β_j, choose $\alpha_i = \beta_i$, and $\ell_i = \dfrac{\beta_i}{c_i}$. Employing Equation (11), it follows that

$$\sum_i \beta_i \frac{c_i \mathrm{dist}_M(e_i)}{\beta_i} \leq \rho. \tag{13}$$

Now

$$\sum_i \beta_i \frac{c_i \mathrm{dist}_M(e_i)}{\beta_i} = \sum_i c_i \sum_j M_{ij} \ell_j = \sum_{i,j} M_{ij} \frac{c_i \beta_j}{c_j} = \sum_j \beta_j \frac{\mathrm{load}_M(e_j)}{c_j},$$

yielding Equation (12). By Lemma 5.5, this implies that there exists a distribution on $\mathcal{M}$ with relative load at most ρ.

We can similarly show that the second statement implies the first by choosing $\beta_i = \alpha_i$, and $c_i = \alpha_i / \ell_i$.

□

References

[1] Reid Andersen and Uriel Feige. Interchanging distance and capacity in probabilistic mappings. *CoRR*, abs/0907.3631, 2009.

[2] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *Journal of Computer and System Sciences*, 69(3):485–497, 2004.

[3] Marek Karpinski, Andrzej Lingas, and Dzmitry Sledneu. Optimal cuts and partitions in tree metrics in polynomial time. *Information Processing Letters*, 113(12):447–451, 2013.

[4] Harald Räcke. Optimal hierarchical decompositions for congestion minimization in networks. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 255–264. ACM, 2008.