Report on Algorithmic Tight Bound Analysis of Tree Metrics Approximation

Xiaojun Sun
A Final Project Report
as a satisfaction of requirement by Advanced Algorithms
Fall Semester 2013

Abstract

This report is discussing the key techniques of J. Fakcharoenphol el.'s journal paper[1]. First, we learn about the importance of adopting tree metric approximating algorithms. Then two key techniques are found: the CKR like algorithm to decompose a graph, as well as a region-growing like algorithm to derandomize the probabilistic approximation. Former algorithm guarantees the expected distortion among a set of trees has an upper bound $O(logn)$, latter one tells the existence of a single tree whose average distortion also satisfies upper bound $O(logn)$. At last this paper concludes that the $O(logn)$ distortion is a tight bound, thus can improve the performance of all tree metric approximation applications.

I. Problem Description

The problem this paper addressing is as its title states: to find a tight bound for the algorithm which approximate arbitrary metrics with tree metrics.

A. Importance of Tree Metrics Approximation Problem

For a lot of problems in both algorithmic research and real life, there exists a metric representation. One example is the Traveling Salesman Problem (TSP). This problem can be represented by a weighted graph, as long as specifying a metric. If we want to solve this problem, we will need to run over all possible tours and find out the optimal, because this is currently proved to be a NP-hard problem. However, since metric space satisfies triangle inequality (will be discussed later in section II-A), a Spanning Tree Heuristic[2] can be adopted, the minimum spanning tree is taken, and bunch of polynomial-time algorithms can be applied on this spanning tree to provide a n approximation to TSP.

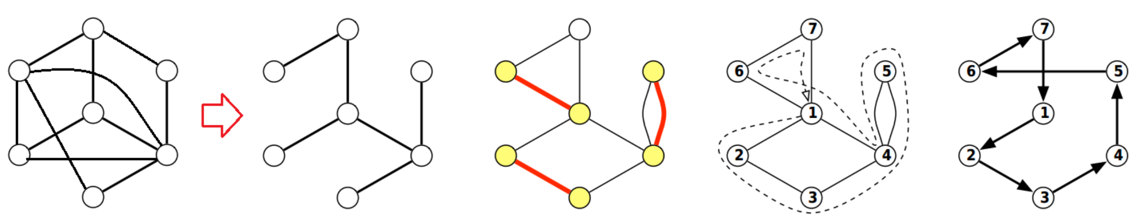

Fig. 1: A spanning tree heuristic for TSP (revising from [3])

In nature, the spanning tree heuristic is a deterministic α-approximation of a graph (without loss of generality, we call it an arbitrary metric space) using tree metric. This problem shows the importance of developing tree metrics approximation algorithms. Furthermore, if we could embed an arbitrary metric into a tree with distortion D, then the optimal solution to this TSP on the tree would be within D of the optimal TSP tour on original input metric. As the distortion be well-bounded at a lower level, the approximation algorithm will be further improved. More compelling algorithmic applications of low distortion embeddings will be listed in section I-C.

B. History

Approximations of metric spaces by "simpler" metric spaces has been intensively studied, both in mathematics and computer science. "Embedding" is an important concept in these research areas, it stands for a map from one metric space to another one, also can be defined in topology with geometry, algebra and domain theory with different sets/domains/fields. Johnson el.(1984)[4] discussed embeddings in Hilbert spaces at Functional theory point of view; Graham el.(1985)[5] did some work about embeddings in $\mathbb{Z}^d$ in graph theory; moreover, Linial el.(1994)[6] gave a research on low-distortion embeddings in low-dimensional real normed spaces.

As to research about tree metrics approximation, it origins from the paper by Cai el.(1995)[7] who gave out a tree spanner approximation to solve K-server problem. Soon, Bartal(1996)[8] introduced an important structure called Hierarchically well Separated Tree (HST), and proved an arbitrary metric space can be α-probabilistically-approximated with distortion $\alpha = O(log^2 n)$, and lower bound is $\Omega(logn)$(this will be extensively discussed in section II-C). Later, Bartal(1998)[9] developed Hierarchical Partition Metrics (HPM) based on HST fashion, and improved the bound of distortion to $O(lognloglogn)$.

Later, people began to focus on how to find a decomposition of graph which can benefit the tree metrics approximation. Calinescu el.(2001)[10] gave an $O(logk)$ approximation algorithm based on a linear programming relaxation for 0-extension problem (k is number of terminals), which is known as CKR algorithm (will be introduced in section II-B). The partition algorithm of CKR was improved by Fakcharoenphol el.(2003, actually authors of this paper)[1], which is adopted finally in this paper to provide a probabilistic graph decomposition algorithm which guarantees the expected distortion have a upper bound $O(logn)$.

C. Potential Applications

Tree metrics approximation can help solve bunch of applications in a variety of fields. The following problems already have reasonable approximation algorithm on trees, so if our goal (to approximate arbitrary metrics with tree metrics) is attained, these problems can be easily solved for much broader inputs.

The group Steiner tree problem. Given a weighted graph and a collection of k sets, find a Steiner tree connecting at least one element from each set. Garg el.[11] give an $O(logklogn)$ approximation algorithm for trees.

Buy-at-bulk network design. Given a weighted graph and a sub-linear function of cost describing the edge cost as a function of the load on it. If given a sequence of pairs of nodes, the objective is to find a path connecting each pair, and minimize the total load. Awerbuch el. [12] give an $O(1)$-approximation algorithm on trees.

The communication spanning tree problem[13]. Given a metric space network with costs on edges, the communication spanning tree problem is to find a spanning tree and minimize a weighted sum of tree distances over all pairs of nodes. Graph decomposition fashion approach can be applied trivially to find desired spanning tree.

Meantime, probabilistic approximation of metric spaces is of particular importance in the case of on-line problems where randomization against oblivious adversaries is very powerful.

Metric task systems consists of a set of states forming a metric space and a set of tasks associated with costs in the different states. The goal is to schedule state transitions in order to minimize the total move and task costs. This problem has deterministic competitive ratio of $2n-1$, and proved to have same upper and lower bound to the $(n-1)$-server problem on n points.[14]

The K-server problem consists of K-servers in a metric space, points are requested over time and a server must be moved to the request location. The memoryless K-server algorithm for trees developed by Chrobak[15] is very efficient randomized algorithm comparing to traditional work function algorithm.

II. Techniques from the Paper

A. Introduction to Basic Concepts

There are some definitions to make, which is necessary to understand the argument of this paper.

Definition 2.1: Given a set X of points, a *distance function* on X is a map $d : X \times X \to \mathbb{R}_+$ that is symmetric, and satisfies $d(i,i) = 0\ \forall i \in X$. The distance is said to be a *metric* if the triangle inequality holds, i.e.

$$d(i, j) \le d(i,k) + d(k, j),\ \forall\ i, j,k \in X.$$

Furthermore, a *metric space* is a set where metric for every element pair is defined. In graph theory, this distance is weight of a path between arbitrary vertices. So any undirected weighted graph (UWG) could be interpreted as a metric space. Given metric space $G = (V,d)$ as an UWG, E is a set including all edges in G, $|V| = n$, pick arbitrary vertices u, v from V, denote the distance between them as $d(u,v)$, The property of metric space can also be written in 3 specifications:

(a) $d(u,v) \ge 0$, and $d(u,v) = 0$ if and only if $u = v$;

(b) $d(u,v) = d(v,u)$ (symmetry);

(c) $d(u,v) \le d(u,w) + d(w,v)$ (triangle inequality).

We also mentioned the "embedding" and "distortion" in section I, they can be formalized as following:

Definition 2.2: Given metric spaces (X,d) and (X',d'), a map $f : X \to X'$ is called an *embedding*.

Definition 2.3: Given two metrics (X,d) and (X',d') and a map $f : X \to X'$, the *contraction* of f is the maximum factor by which distances are shrunk, i.e.,

$$\max_{x,y \in X} \frac{d(x,y)}{d'(f(x), f(y))}$$

the *expansion* or *stretch* of f is the maximum factor by which distances are stretched:

$$\max_{x,y \in X} \frac{d'(f(x), f(y))}{d(x,y)}$$

and the *distortion* of f , denoted by $||f||_{dist}$, is the product of the contraction and the expansion.

Easy to understand that "distortion" is the maximum difference between 2 metrics. Next it is necessary to define the most important concept in the tree approximation algorithm: "α-probabilistic approximation".

Definition 2.4: A metric space N over V, *dominates* a metric space M over V, if for every $u, v \in V$, $d_N(u,v) \ge d_M(u,v)$.

Definition 2.5: A metric space N over V, α-*approximates* a metric space M over V, if it dominates M and for every $u, v \in V$, $d_N(u,v) \le \alpha \cdot d_M(u,v)$.

Definition 2.6: A set of metric spaces $\mathcal{S}$ over V, α-*probabilistically-approximates* a metric space M over V, if every metric space in $\mathcal{S}$ dominates M and $\exists$ a probability distribution over metric space $N \in \mathcal{S}$ such that for every $u, v \in V$, $Ex(d_N(u,v)) \le \alpha \cdot d_M(u,v)$.

Here Ex means expectations. Consider an optimization problem $\mathcal{P}$ defined on metric spaces, the cost of solution for $\mathcal{P}$ can be expressed as a linear combination of distances between vertices in metric space.[9] Use the linearity of expectations, if the probability distribution is known, α-probabilistic approximation is a good approximation when deterministic approximation is unavailable, just like the problem we concern: approximate an arbitrary metric with a single tree metric.

There is an example to explain how to calculate distortion α, for an α-probabilistic approximation for a tree to a graph.

Example 2.1: Suppose we have a n-cycle graph, remove one edge randomly to get a tree. For each edge $(u,v) \in E$ on this tree, the probability of distant change is $\frac{1}{n}$, while probability for unchange is $\frac{n-1}{n}$. So the expectation of distortion on one edge is (assume initial distance in n-cycle for neighbors are all 1, the distortion is then equal to distance on tree metric space):

$$Ex(d_T(u,v)) = \frac{1}{n} \cdot (n-1) + \frac{n-1}{n} \cdot 1 = 2\frac{n-1}{n} \le 2$$

We can assert that this n-cycle can be *2-probabilistically-approximated* by a tree with uniform probability distribution.

B. Graph Decomposition and CKR Algorithm

Example 2.1 is a good explanation on how to build a tree out of a graph by cutting some edges. The graph decomposition techniques reply on a CKR-like procedure. The CKR algorithm is shown below: In this algorithm

ALGORITHM 1: The CKR Algorithm

Input: A semi-metric space (where distinct vertices may have 0 distance) $G=(V,d)$, while every edge has corresponding distance greater than 1
Output: A tree semi-metric space dominating original semi-metric space
Choose a random permutation σ of terminals(for 0-extension problem);
Choose a real number α uniformly at random from $[1,2)$;
for *each vertex* $v \in V$ **do**
    $l \leftarrow 1$;
    while *v has not yet been assigned a terminal* **do**
        if $d(v,\sigma(l)) \leq \alpha R(v)$ **then**
            assign v to terminal $\sigma(l)$;
        end
        $l \leftarrow l+1$;
    end
end

$R(v)$ is the distance of *closest terminal of v* to *this terminal's closest terminal*. Terminal is a special set of vertices in multi-way cut problem which can be used as roots of tree. More details such as 0-extension problem and semi-metric are reasoned in [10], they do not make any difference to the new algorithm. Here, the authors make some modifications to this CKR algorithm to achieve graph decomposition.

The procedure can be briefly described like below: use balls of different sizes to cut this graph, the first ball's radius is D_i which can contain all vertices. Then recursively choose vertices as centers to draw balls in half of previous radius $D_{i-1} = \frac{D_i}{2}$. Once the circle (bound of ball) cuts any edges, delete them, repeat drawing until every vertex is sitting within a ball, at this time we only keep edges from higher layer's center to lower layer's centers. Now doing same thing recursively for each smaller ball with smaller radius $\frac{D_{i-1}}{2}$ (or in other words, go into lower layer $i-2$). Repeat this until every vertex is a center of a ball, all remaining edges and all vertices form a tree.

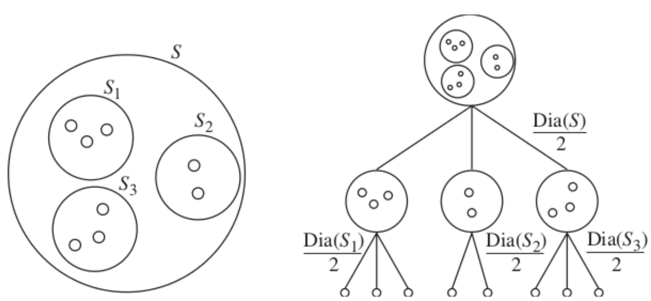

Fig. 2: Brief procedure of graph decomposition by ball cutting[1]

We call this algorithm "FRT algorithm" in the same fashion of "CKR algorithm"(C,K,R are initials of authors). We can see one main difference is the probability to choose ball radiuss, this will be important in the tight bound analysis later in section III-A.

A graphical example of FRT decomposition is attached in APPENDIX section.

C. k-Hierarchically well Separated Trees

The definition of k-hierarchically well separated tree (k-HST) was first state in Bartal's paper[8].

ALGORITHM 2: The FRT Algorithm

Input: A metric space $G = (V, d)$, while every edge has corresponding distance strictly greater than 1
Output: A tree metric space dominating original metric space
Choose a random permutation π of all vertices;
Choose a real number β randomly from $[1,2]$ at distribution $p(x) = \frac{1}{xln2}$;
$D_\delta \leftarrow V$; $i \leftarrow \delta - 1$; /* D means set of vertices in the ball, largest ball has radius 2^δ */
while *D_{i+1} has non-singleton clusters* **do**
  $\beta_i \leftarrow 2^{i-1}\beta$; /* Choose radius not exactly on powers of 2 */
  for $l = 1, 2, \ldots, n$ **do**
    for *every cluster S in D_{i+1}* **do**
      Create a new cluster consisting of all unassigned vertices in S closer than β_i to $\pi(l)$
    end
  end
  $i \leftarrow i - 1$;
end

Definition 2.7: A *k-hierarchically well separated tree* is defined as a rooted weighted tree with following properties:

(a) The edge weight from any node to each of its children is the same.

(b) The edge weights along any path from the root to a leaf are decreasing by a factor of at least k.

Obviously in FRT graph decomposition, the tree we construct is 2-HST, because: weights on tree edge in the same layer are all 2^i (upper bound), and the radius shrinks by factor 2 when going down one layer. Actually in [9], 2-HST can be converted to any k-HST with distortion $O(k/logk)$, this proves the FRT algorithm is able to be better utilized on more general applications even taking another factor rather than 2.

III. Computational Analysis

A. Graph Decomposition Bound Analysis

In section II-B we mentioned the brief procedure of graph partition method (FRT algorithm) used in this paper. In this section, we will prove that the expectation of tree metric distortion compared to original graph has a $O(logn)$ bound.

The computation of expectation of distortion follows the method in example 2.1. As the proof in this paper, we define to formalize:

Definition 3.1: Center w *settles* the edge (u, v) at level/layer i if it is the first center to which at least one end of this edge (u and v) get assigned.

From this definition, there is exactly one center settles any edge (u, v) at any particular level. Note that edge (u, v) is in level i means u and v first get separated to different clusters at level i. Also a formal definition of "cut" can be made here:

Definition 3.2: Center w *cuts* edge (u, v) at level i if it settles (u, v) at this level, but exactly one of u, v is assigned to (included in ball) w at level i.

One example of "cut" is in fig 3, $\pi(1)$ is the center of grey shaded ball, it has radius $2^{i-1}\beta$. Assume this ball is the first ball we drew in this level, and there exists an edge $(\pi(10), \pi(2))$. Since $\pi(10)$ is first time to be assigned to $\pi(1)$, we call $\pi(1)$ cuts this edge at level $i-1$.

Then we can review the proof in this paper (part 2.3, p 490-492) step by step. First, it says

Theorem 3.1: If w cuts edge (u, v), the tree length of this edge (u, v) is about 2^{i+2}.

Proof Tree metric we need always dominates original metric, so we need to estimate the maximum; current radius that cut (u, v) is 2^i maximum (please refer to FRT algorithm, $2^{i-1}\beta \leq 2^i$), so the distance between u, v is 2 times of current diameter, i.e. $4 \times 2^i = 2^{i+1}$. Fig 4(a) illustrates this assert. In that figure, u is assigned to center w,

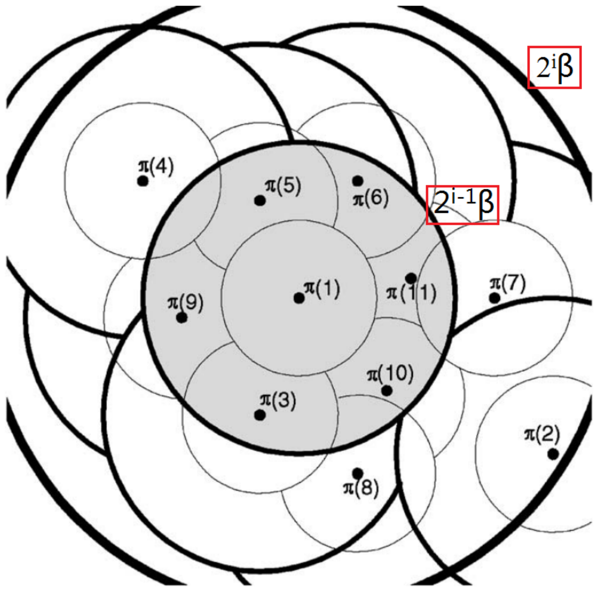

Fig. 3: Sample for hierarchically ball cutting (revised from [1])

and there must exist a center w' comes later (in the same level or lower levels) to which v is assigned. We know center w cuts edge (u,v) at level i, then the radius of ball w is r, the radius of ball w' must satisfy $r' \leq r$. Note that the distance between w, w' should be less than $2r$, otherwise (u,v) is located already in a higher level. So the largest possible distance of u,v is $4r$. $\square$

Then the new distance counting on the tree metric space associated with center w is denoted with $d_w^T(u,v)$. Consider the center w is arbitrarily taken from a set of vertices, it is necessary to apply the upper bound on an arbitrary partition. Arrange a sequence of these centers $\{w_1, w_2, \ldots, w_s, \ldots, w_n\}$ in the order of $d_{w_1}^T(u,v) \leq d_{w_2}^T(u,v) \leq \cdots d_{w_s}^T(u,v) \leq \cdots d_{w_t}^T(u,v)$, w_s is picked arbitrarily. Without loss of generality, assume $d(w_s,u) \leq d(w_s,v)$. To satisfy w_s cuts edge (u,v), 2 conditions must be fulfilled:
(a) $d(w_s,u) \leq \beta_i < d(w_s,v)$ for some i;
(b) w_s settles (u,v) at level i.

The expectation of tree metric distance when condition (a) is fulfilled is:

$$\begin{aligned} Ex\left(d_{w_s}^T(u,v) \mid d(w_s,u) \leq \beta_i < d(w_s,v)\right) &= \sum Pr(\beta_i \ lies \ in \ [\ d(w_s,u), d(w_s,v)\ )) \cdot d_{w_s}^T(u,v) \\ &= \int_{d(w_s,u)}^{d(w_s,v)} \frac{1}{x ln 2} \cdot d_{w_s}^T(u,v) \cdot dx \end{aligned}$$

note that the value of $d_{w_s}^T(u,v)$ has upper bound $2^{i+2} = 8 \cdot 2^{i-1} \leq 8 \cdot 2^{i-1}\beta = 8\beta_i$.

Condition (b) is fulfilled when w_s is the first in sequence of $< w >$ which can settle (u,v) at level i. Since w_j defines a tree distance smaller than w_s when $j < s$, with the same radius, w_j can surely "settle" this edge at level i, an example is fig 4(b). In this example we assume tree distance is $d(w_s,u) + d(w_s,v)$, easy to observe that any center w_j configuring a shorter tree distance is sitting at a location satisfying the definition of "settle" edge (u,v) when applying radius β_i.

Concluding above arguments, in a consecutive sequence $\{w_1, w_2, \ldots, w_s\}$ every element can settle (u,v) if it is the first center selected at level i under condition (a). Considering that w_j may also settle (u,v) even $j > s$ and w_s is randomly chosen, probability

$$Pr(\text{condition (b)} \mid \text{condition (a)}) \leq \frac{1}{s}$$

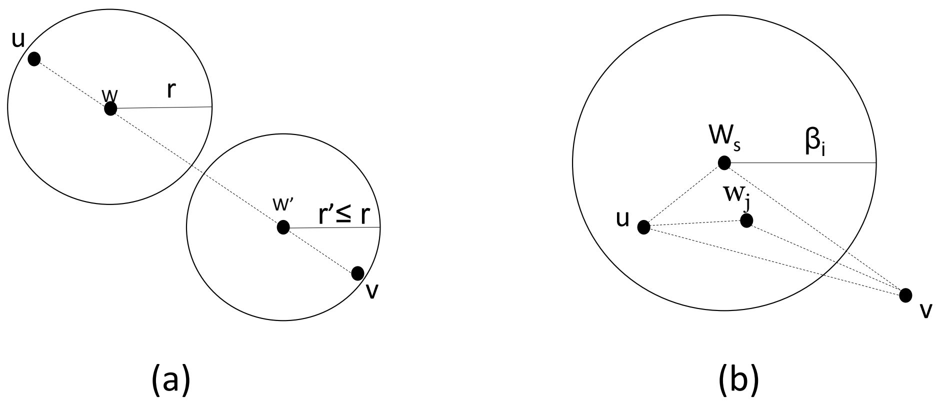

Fig. 4: Examples for partition upper bound proof

this is a conditional probability. Following assertion $Pr(A \text{ and } B) = Pr(A) \cdot Pr(B|A)$, probability for center w_s to cut (u,v) is

$$\frac{1}{s}\int_{d(w_s,u)}^{d(w_s,v)} \frac{1}{xln2} \cdot d^T_{w_s}(u,v) \cdot dx$$

since $\frac{1}{s}$ is irrelevant with variable x, it can be directly multiplied into expectation calculation, so the expected tree metric distance between u, v through w_s is

$$\begin{aligned}
Ex\left(d^T_{w_s}(u,v)\right) &\leq \int_{d(w_s,u)}^{d(w_s,v)} \frac{1}{xln2} \cdot 8x \cdot \frac{1}{s} dx \\
&= \frac{8}{sln2}(d(w_s,v) - d(w_s,u)) && (1) \\
&\leq \frac{8d(u,v)}{sln2} \quad \text{(triangle inequality)} && (2)
\end{aligned}$$

w_s is randomly chosen from sequence containing n vertices. Using linearity of expectation, the expected tree metric distance is bounded by

$$Ex(d_T(u,v)) \leq \sum_{s=1}^{n} \frac{8d(u,v)}{sln2} = O(logn) \cdot d(u,v)$$

The conclusion is

Theorem 3.2: The distribution over tree metrics resulting from FRT algorithm $O(logn)$-probabilistically-approximates the original metric d.

B. Derandomization Bound Analysis

Derandomization means that FRT algorithm not only provide a set of tree metrics with desired expectation on distortion, but also could actually give a single tree metric such that its distortion is bounded by $O(logn)$.

To generalize, assume for each edge (u,v) there is weight c_{uv} assigned on it. We want to prove that

Proposition 3.1: The FRT algorithm can find a tree metric space d^T such that (V,d^T) dominates original metric space $G=(V,d)$ and satisfies

$$\sum_{u,v\in V} c_{uv}\cdot d^T(u,v)\le O(logn)\sum_{u,v\in V} c_{uv}\cdot d(u,v)$$

In order to formalize, we can define

Definition 3.3: Given an edge $e=(u,v)$ of length d_e and weight c_e, the *volume* of the edge is $d_e\cdot c_e$, the *sum of volume* $W=\sum_e c_e d_e$ corresponding to a set of qualified edges.

Using the definition of volume, it is easy to further define the following concepts vital in our proof.

Definition 3.4: Ball is a set of vertices located in a ball with center t and radius r, denoted by $B(t,r)$. The *volume of neighborhood* $W(t,r)$ is computed by following criteria:

(a) edge $e=(u,v)$ with both ends in $B(t,r)$ contributes $c_e d_e$ to $W(t,r)$;

(b) edge $e=(u,v)$ with exactly one end such as u in $B(t,r)$ contributes $c_e(r-d(t,u))$ to $W(t,r)$;

(c) edge $e=(u,v)$ with no ends in $B(t,r)$ contributes 0 to $W(t,r)$;

Cut volume is denoted as $c(t,r)=\sum_{u\in B(t,r),v\notin B(t,r)} c_{uv}$ representing total weight of edges cut by $B(t,r)$.

Based on above definitions, the proof in this paper can be reproduced. First, this assertion is proved:

Theorem 3.3: There are radii $r_i: 2^{i-1}\le r_i<2^i$ such that $\sum_i \frac{c(t,r_i)}{W(t,r_i)}\cdot 2^i\le O(logn)$ if W is polynomial function of size n.

The proof is very detailed in this paper already (part 3.1, p491), using region growing lemma from Garg el.[16] and contradiction. Then the author relaxed pre-condition that W has to be polynomial function of size n by find the actual upper bound of W. The largest edge length is $O(n)$, and unit volume also affects the total volume linearly ($O(n)$), so the unit neighborhood volume on unit edge length $W_0=W(t,1)$ is at least $(O(n\cdot n))^{-1}=\Omega(1/n^2)$ times total volume W, ratio W/W_0 is bounded by $O(n^2)$, then $ln(n^2)=O(logn)$, the results with assumption still stand.

Above theorem is used to calculate the deterministic upper bound distortion of FRT algorithm. Let t be the center that maximizes $W(t,2^i)$ (2^i is highest level maximum radius as mentioned in section III-A). The FRT algorithm cut out $B(t,r_i)$ (r_i is chosen in this level w.r.t. t), and the new tree metric edge has length upper bound 2^{i+2} as discussed.

Recall the inequality in our proposition, $c_{uv}\cdot d^T(u,v)\le c(t,r_i)\cdot 2^{i+2}$ for one edge at level i, $\sum_{u,v\in V} c_{uv}\cdot d(u,v)=W(t,r_i)$ according to definition. Sum up for all possible reached levels:

$$\frac{\sum_{u,v\in V} c_{uv}\cdot d^T(u,v)}{\sum_{u,v\in V} c_{uv}\cdot d(u,v)}=\sum_i \frac{c(t,r_i)}{W(t,r_i)}\cdot 2^{i+2}\le 4ln\frac{W(t,2^{i+2})}{W(t,2^i)}\le 8ln\frac{W}{W(t,1)}=O(log(n)),$$

our proposition is proved. Note here t is chosen to guarantee it can still cut some edge with unit(minimum) length 1 to form unit volume $W(t,1)$.

In conclusion:

Theorem 3.4: FRT algorithm returns a 2-hierarchically well separated tree metric d_T such that

(a) $\forall u,v\in V, d_T(u,v)\ge d(u,v)$;

(b) $\sum_{u,v\in V} c_{uv}\cdot d^T(u,v)\le O(logn)\sum_{u,v\in V} c_{uv}\cdot d(u,v)$.

C. Computational Complexity Analysis

In algorithm 2, there are 3 cycles exploiting every vertices, the time complexity is $O(n^3)$, a polynomial time complexity.

IV. Further Developments on this Paper

A. Improvements on Relevant Applications

All applications we mentioned in I-C and listed in Part 4 in this paper (we are discussing) get an improvement on their performance since the upper bound of distortion is optimized. Moreover, its graph decomposition approach also inspires some now approximations like spatial approximation in metric labeling problem in image processing[17] and trip planing about big data[18].

In graph theory the low-stretch spanning tree is also discussed as an important topic[19].

B. Formalization and Refinement on Theory

Some refinements are made on the FRT algorithm, especially on the probability distribution to select ball radii, such as Cai[20]. However, since the tight bound can be applied generally in similar approximation algorithms, there is no improvement on further constraining the distortion.

And Shalekamp relaxed the distribution assumption made in this FRT algorithm, and directly gave a proof for the tight bound, this is a big progress to formalize the tree approximation algorithms[21].

V. My Comments

The tree approximation algorithm is a good approach to solve some NP-hard problems in graph theory, thus to find a tight bound of the tree metric distortion is important to optimize lots of approximation algorithms. The derandomization process guarantees its correctness for on-line problems. In conclusion, this paper is innovative and vital enough and worth 400+ citations.

References

[1] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar, “A tight bound on approximating arbitrary metrics by tree metrics”, *Journal of Computer and System Sciences*, vol. 69, pp. 485–497, 2004.

[2] Daniel J Rosenkrantz, Richard E Stearns, and Philip M Lewis, II, “An analysis of several heuristics for the traveling salesman problem”, *SIAM journal on computing*, vol. 6, pp. 563–581, 1977.

[3] Jeff Erickson, “Algorithms course materials”.

[4] William B Johnson and Joram Lindenstrauss, “Extensions of lipschitz mappings into a hilbert space”, *Contemporary mathematics*, vol. 26, pp. 1, 1984.

[5] RL Graham and PM Winkler, “On isometric embeddings of graphs”, *Transactions of the American mathematical Society*, vol. 288, pp. 527–536, 1985.

[6] Nathan Linial, Eran London, and Yuri Rabinovich, “The geometry of graphs and some of its algorithmic applications”, *Combinatorica*, vol. 15, pp. 215–245, 1995.

[7] Leizhen Cai and Derek G Corneil, “Tree spanners”, *SIAM Journal on Discrete Mathematics*, vol. 8, pp. 359–387, 1995.

[8] Yair Bartal, “Probabilistic approximation of metric spaces and its algorithmic applications”, *in Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium on*, pp. 184–193. IEEE, 1996.

[9] Yair Bartal, “On approximating arbitrary metrices by tree metrics”, *in Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pp. 161–168. ACM, 1998.

[10] Gruia Calinescu, Howard Karloff, and Yuval Rabani, “Approximation algorithms for the 0-extension problem”, *in In Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, 2001.

[11] Naveen Garg, Goran Konjevod, and R Ravi, “A polylogarithmic approximation algorithm for the group steiner tree problem”, *in Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms*, pp. 253–259. Society for Industrial and Applied Mathematics, 1998.

[12] Baruch Awerbuch and Yossi Azar, “Buy-at-bulk network design”, *in Foundations of Computer Science, 1997. Proceedings., 38th Annual Symposium on*, pp. 542–547. IEEE, 1997.

[13] Te C Hu, “Optimum communication spanning trees”, *SIAM Journal on Computing*, vol. 3, pp. 188–195, 1974.

[14] Mark Manasse, Lyle McGeoch, and Daniel Sleator, “Competitive algorithms for on-line problems”, *in Proceedings of the twentieth annual ACM symposium on Theory of computing*, pp. 322–333. ACM, 1988.

[15] Marek Chrobak and Lawrence L Larmore, “An optimal on-line algorithm for k servers on trees”, *SIAM Journal on Computing*, vol. 20, pp. 144–148, 1991.

[16] Naveen Garg, Vijay V Vazirani, and Mihalis Yannakakis, “Approximate max-flow min-(multi) cut theorems and their applications”, *SIAM Journal on Computing*, vol. 25, pp. 235–251, 1996.

[17] Matthieu Guillaumin, Thomas Mensink, Jakob Verbeek, and Cordelia Schmid, “Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation”, *in Computer Vision, 2009 IEEE 12th International Conference on*, pp. 309–316. IEEE, 2009.

[18] Feifei Li, Dihan Cheng, Marios Hadjieleftheriou, George Kollios, and Shang-Hua Teng, “On trip planning queries in spatial databases”, *in Advances in Spatial and Temporal Databases*, pp. 273–290. Springer, 2005.

[19] Michael Elkin, “Low stretch spanning trees”, *in Encyclopedia of Algorithms*, pp. 1–99. Springer, 2008.

[20] Yu-hua Cai and Feng-ying Wei, “Probabilistic approximation algorithm of metric spaces”, *Journal of Shandong University*, vol. 48, pp. 51–55, 2013.

[21] Schalekamp F., “Probabilistic tree embedding”, *in http://fransschalekamp.com/research/teaching/FRT.pdf*, 2012.