Tree-Approximations for the Weighted Cost-Distance Problem (extended abstract)

Christian Schindelhauer[1] and Birgitta Weber[2]

[1] Department. of Mathematics and Computer Science and Heinz-Nixdorf-Institute, Paderborn University, 33095 Paderborn, Germany,
`schindel@upb.de`,
[2] Institute of Theoretical Computer Science, ETH Zentrum, CH-8092 Zürich, Switzerland,
`weberb@inf.ethz.ch`

Abstract. We generalize the Cost-Distance problem: Given a set of n sites in k-dimensional Euclidean space and a weighting over pairs of sites, construct a network that minimizes the cost (i.e. weight) of the network and the weighted distances between all pairs of sites. It turns out that the optimal solution can contain Steiner points as well as cycles. Furthermore, there are instances where crossings optimize the network.

We then investigate how trees can approximate the weighted Cost-Distance problem. We show that for any given set of n sites and a non-negative weighting of pairs, provided the sum of the weights is polynomial, one can construct in polynomial time a tree that approximates the optimal network within a factor of $O(\log n)$. Finally, we show that better approximation rates are not possible for trees. We prove this by giving a counter-example. Thus, we show that for this instance that every tree solution differs from the optimal network by a factor $\Omega(\log n)$.

1 Introduction

1.1 Problem and Motivation

Given n terminal points in the Euclidean space we investigate the problem of constructing a network with small cost and short distances. This research is motivated by a number of practical problems arising in network design for real traffic, as well as traffic in communication networks. It is often observed that the cost of networks can be described by a component depending only on the size of the network and by a component growing with the demand of certain connections. Consider a street network: if one minimizes only the network size to cover cost for building and maintenance, the connections between terminals can grow by the diameter of the network. Then, additional costs caused by detours outweigh the fixed costs.

In practice network designers model the demand in a network by a so-called *origin-destination matrix* $w(u, v)$. For sites u, v it describes the traffic starting at u with destination v. We model the cost of the network for each edge by a linear function $c_1||e||_2 + c_2 \sum_{(u,v)\in P(e)} w(u, v)||e||_2$ for $c_1, c_2 > 0$, where $||e||_2$ denotes the Euclidean length of the edge and $P(e)$ is the set of all pairs (u, v) such that the shortest path between u and v contains e. By summing over all edges we define the *Weighted Cost-Distance* (WCD) of a network N and a weighting w:

$$\mathrm{WCD}_w(N) := \sum_{e \in E(N)} \left(c_1 ||e||_2 + c_2 \sum_{(u,v) \in P(e)} w(u, v) ||e||_2 \right) . \tag{1}$$

So, for a pair u, v with large weight $w(u, v)$ (frequent traffic) a detour between u and v implies higher costs than between pairs with smaller weight.

There is a trade-off between cost and weighted distance. If we choose $c_2 = 0$ we face the intensively studied *minimum network problem*. If we choose $c_1 = 0$, the optimal solution is a complete network. For sites in general position and positive weights. As we scale the parameter c_1/c_2 from 0 to ∞, we see a gradual transformation from the Steiner tree to the complete network. We are interested in the structure of the intermediate states.

For simplicity we replace the above definition by the following. Since we only consider $c_2 > 0$, we can set $c_1 = c_2 = 1$ if we simultaneously modify the weighting by $w'(u, v) = \frac{c_1}{c_2} w(u, v)$. This results in the following equivalent version of the Weighted Cost-Distance:

$$\mathrm{WCD}_w(N) := \sum_{e \in E(N)} c(e) + \sum_{u,v \in V(N)} w(u, v) L_N(u, v) , \tag{2}$$

where $c(e)$ denotes the cost of an edge and $L_N(u, v)$ the length of the shortest path from u to v in the network N. We use this notation throughout this paper. The corresponding optimization problem is defined as follows.

Definition 1. *Let $L_G(u, v)$ denote the minimum length of a path of vertex u to v in graph G.*

- Weighted Cost-Distance Network problem (CDN): *Given a set of sites V in Euclidean space and a weighting $w : V \times V \mapsto \mathbb{R}^+$, find a network $N = (V, E)$ that optimizes the Cost-Distance WCD$_w(N)$ (according to equation (2)).*
- Weighted Cost-Distance Tree problem (CDT): *Given V and $w : V \times V \mapsto \mathbb{R}^+$, find a tree $T = (V, E)$ that optimizes the Cost-Distance WCD$_w(T)$.*

In addition to the sites we allow the use of a non-terminal vertex set, if not explicitly stated otherwise.

1.2 Previous Work

If the weights are set to zero, and no resctrictions for the non-terminals are given the Weighted Cost-Distance problem reduces to the *Euclidean Steiner Tree problem*. It was shown to be NP-hard by Garey, Graham and Johnson [GGJ76]. However, in his groundbreaking paper Arora [Aro98] showed that this problem admits a polynomial time approximation scheme.

In [KRY95] the *Balanced Spanning Tree* problem was introduced. Here, the task is to find a tree which optimizes the term

$$\sum_{e \in E(T)} c(e) + \sum_{s \in V} L_T(s,r)$$

for a given root r under a metric c (not necessarily Euclidean). Non-terminal sites are not available.

The authors prove the existence of trees, where the dilation of all vertices' distances from the root is bounded by any $\alpha > 1$ and the trees cost is at most β times the cost of the minimum spanning tree, where $\beta = 1 + \frac{2}{\alpha - 1}$. This leads to a constant polynomial time bounded approximation algorithm.

The Balanced Spanning Tree problem is a variant of the Weighted Cost-Distance Network problem, if we allow general metrics and exclude non-terminal vertices. The weighting is limited to $w(r,u) = 1$ and $w(u,v) = 0$ for $u, v \in V \setminus \{r\}$. For this problem in [KRY95] it is shown that a tree is always part of the optimal solution and approximating networks can be pruned to trees. Hence, here the Cost-Distance Network problem reduces to the Cost-Distance Tree problem.

Meyerson et al. [MMP00] generalize this problem by introducing a positive vertex weighting, and by allowing two different metrics for cost and distance: the length metric ℓ and the cost metric c. The Cost-Distance measure is given by

$$\sum_{e \in E(T)} c(e) + \sum_{s \in V} w(s) L_T(s,r)$$

for a root r. They present a polynomial time bounded randomized algorithm that approximates the problem within a factor of $O(\log n)$. Furthermore, they show that the optimal solution is always a tree.

A t-spanner is a connected partial graph of a given graph G such that for all vertices $u, v \in V(G)$ the corresponding shortest path in the t-spanner is at most t times longer than in G. There exist t-spanners in Euclidean space, whose sizes are bounded linearly by the size of the minimum spanning tree [ADD$^+$95]. It turns out that these spanning networks already allow us to state constant factor approximation algorithms for the Weighted Cost-Distance Network problem.

Theorem 1 ([ADD$^+$95]). *In k-dimensional Euclidean space, for any $t > 1$ there exists a t-spanner with size $O(c(MST))$, which can be computed in time $O(n \log n)$.*

This immediately implies that t-spanners allow constant factor approximation for the CDN-problem.

Corollary 1. *For Euclidean space the Weighted Cost-Distance Network problem can be approximated by a constant factor within time $O(n \log n)$.*

For the two-dimensional Euclidean space we can pin down the constant very accurately by using the result of [LL89].

Lemma 1. *[LL89] For $r > 0$, there exists a $(1 + \frac{1}{r})\frac{2\pi}{3\cos(\pi/6)}$-spanner of the complete graph, whose size is at most $2r + 1$ times the costs of the minimal spanning tree.*

Optimizing the choice of t leads to the following result:

Theorem 2. *For the Euclidean plane there exists a polynomial time approximation of the Weighted Cost-Distance Network problem, where we do not allow non-terminal vertices, by a factor of* $\frac{2\pi+3+\sqrt{4\pi^2+36\pi+9}}{3\sqrt{3}} \approx 4.23\ldots$.

A complete proof can be found in [Web01].

Using the results in [Bar98] and [CCG$^+$98] one can transfer the t-spanner result of [ADD$^+$95] to arbitrary metrics. However the cost is increased by a logarithmic term. Such t-spanners give an approximative solution for CDN:

Corollary 2. *For metric costs and distances the Weighted Cost-Distance Network problem can be approximated in polynomial time within a factor of* $O(\log n)$.

1.3 The Optimal Network is not a Tree

For the minimum network problem it is known that introducing non-terminal vertices helps to reduce the network costs (i.e. size) by a constant factor. The optimal choice of such vertices are Steiner points.

Many properties are known for these Steiner networks. First of all minimum networks are trees. Further, in the plane Steiner points have degree three and the angle of neighbored edges is $120°$. The number of these non-terminal points is bounded by $n-2$.

A complete analysis of even small graphs shows that non-terminal sites also allow an improvement of a constant factor for the CDN-problem. Nevertheless, the angles between the adjacent edges may differ from $120\,°$.

In contrast to the Cost-Distance Problems investigated so far, it turns out that the optimal solution is not a tree. We will prove in section 3 that a tree can differ by at least a factor of $\Omega(\log n)$ from the optimal network. Even more surprisingly, non-terminal (quasi-Steiner points) may be involved in cycles and there may be cycles that connect only quasi-Steiner points.

Another interesting observation is that the optimal network may include crossing edges, where the placement of a quasi-Steiner point onto the crossing point does not improve the solution. This reminds of the open problem whether optimal dilation trees contain crossings.

For a more detailed discussion of the topics, addressed in this introductory section we refer to [Web01]. Examples for crossing and quasi-Steiner points can be seen in figures 1, 2 and 3. In the following section we will prove that the optimal Cost-Distance network can be approximated by a tree within a factor of $O(\log n)$. Furthermore, there is a polynomial time bounded algorithm that computes such a tree, given the weighting and the sites in Euclidean k-dimensional space. In section 3 we prove the optimality of this approximation factor. We finally conclude these results and present some open problems for further research.

2 A Tree-Approximation by Factor of $O(\log n)$

Note that for k-dimensional Euclidean space the quality of the minimum networks differs from the minimum spanning tree only by a constant factor. For the Cost-Distance

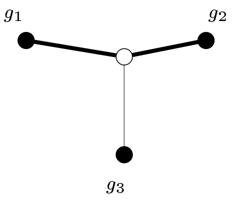

Fig. 1. The optimal WCD-network contains a quasi-Steiner point. $w(g_1, g_2) \gg w(g_1, g_3) \wedge w(g_1, g_2) \gg w(g_1, g_2)$

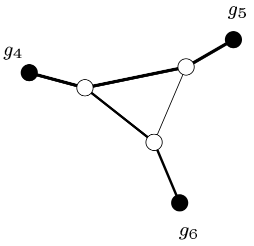

Fig. 2. The optimal WCD-network contains a cycle. $w(g_4, g_5) \approx w(g_4, g_6) \wedge w(g_4, g_5) > w(g_5, g_6)$

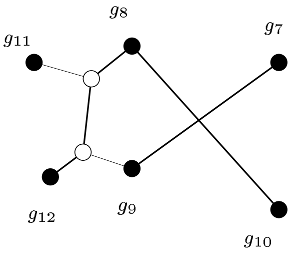

Fig. 3. An instance where a crossing is part of the optimal solution. $w(g_8, g_{10}) \approx w(g_7, g_9) \wedge w(g_8, g_{12}) > w(g_9, g_{11}) \wedge w(g_9, g_{10}) = w(g_8, g_7) = 0$

problem the situation is similar. Therefore we will not use any non-terminals in the following construction.

We use the notion of a *split tree* [ADD$^+$95]. A *split tree* is a tree that stems from a hierarchical decomposition of a point set into k-dimensional rectangles of bounded aspect ratio, say in the range $[\frac{1}{3}, 3]$. We start with the smallest possible rectangle, $R_0 = R(V)$, including the point set V. Let r_0 be the root of the split tree. This rectangle R_0 is split into two smaller rectangles R_1 and R_2. Let $V(R)$ be the subset of vertices in rectangle R. The split tree of R_1 is the split tree for the vertices $V(R_1)$, and similary for R_2 and $V(R_2)$. These subtrees are connected to the root r_0.

We will construct a *fair split tree* (FST) where each sub-tree with vertex set V' has a diameter of $O(kd(V'))$, where $d(V') := \max_{u,v \in V'} ||u, v||_2$. Let $\ell(R)$ be the length of the longest edge of a rectangle R. We will use the following recursive construction given a rectangle R, a root $r \in V(R)$ and a weighting w such that for some $c > 1$: $W := \sum_{u,v} w(u, v) = O(n^c)$.

1. If $\ell(R) \leq \frac{d(V)}{n^c}$, then we choose an arbitrary vertex $r \in R$ and connect all vertices $V(R)$ directly to r.
2. Otherwise, we partition the rectangle R by a hyperplane orthogonal to an edge e with length $\ell(R_0)$. The distance between the hyperplane and the ends of the longest edge is at least $\frac{1}{3}\ell(R_0)$. The exact position depends on the weighting and will be described in the proof of Theorem 3.
   The resulting two axis-parallel adjacent rectangles partitioning R are called R_1 and R_2.
   (a) If r is in R_1 let $r_1 = r$ and take an arbitrary vertex $r_2 \in R_2$ and vice versa if $r \in V(R_1)$. Insert the edge $\{r_1, r_2\}$.
   (b) Recursively, proceed with R_1, r_1 and R_2, r_2.

Note that $d(V) \leq \ell(R_0)$ and observe that after k rounds the length of the longest edge is reduced by at most a factor of $\frac{2}{3}$. So there are only $O(k \log n)$ rounds until the size of the rectangles is bounded by $\frac{\ell(R_0)}{n^c}$. The length of every path in the resulting tree is bounded by $3k\ell(R_0)$: starting from the vertex of the path closest to the root, following the path downwards in both directions, the lengths of the edges $e_1, e_2, \ldots$ and $e'_1, e'_2, \ldots$ are upper bounded by $||e_i||_2, ||e'_i||_2 \leq (\frac{2}{3})^{\lfloor i/k \rfloor} d(V)$.

Lemma 2. *Fair split trees have diameter $3kd(V)$ and weight $O(s(MST(V))k \log n)$.*

Proof. We apply the Lemma of [Epp00,DHN93] using the isolation property. If we add non-intersecting cylinders to all edges with radius $r/3$ and distance $r/3$ to the end points, then the cost of the corresponding network is linearly bounded by the cost of the MST. (The isolation property also holds if the cylinder is replaced by other geometric objects). Note that for the edges of each recursion step, we can attach such a cylinder to an edge such that the cylinder is completely in the corresponding rectangle. Since there are at most $O(k \log n)$ recursion steps this implies the claim. □

We have not presented where we place the split. The following Lemma helps us to make a good selection.

Lemma 3. *Given rectangle R_0 and a weighting $w : V \times V \mapsto \mathbb{R}_0^+$. There exists partition of V into rectangles R_1 and R_2 with vertex sets V_1, V_2 such that*

$$\sum_{(u,v)\in\ V_1\times V_2 \cup V_2 \times V_1} w(u,v) \ \leq \ \frac{3D}{\ell(R_0)}$$

where $D := \sum_{u,v\in V} w(u,v)||u,v||_2$.

Proof. Define $p := \left\lfloor \frac{\ell(R_0)}{3\Delta} \right\rfloor$ adjacent parallel rectangles R_i of thickness $\Delta := \frac{D}{W}$, where $W := \sum_{u,v\in V} w(u,v)$. These rectangles have distance of at least $\ell(R_0)/3$ to the left and right end of the longest edge of R_0. We will partition between a pair R_i and R_{i+1}.

Next consider pair a pair of vertices u, v with $u \in R_i$ and $v \in R_j$. Then, we have $||u, v||_2 \geq \Delta \cdot |i - j|$. Measure v_i which is the weight of all connections crossing the right border between R_i and R_{i+1}:

$$v_i = \sum_{j\leq i<k} \sum_{u\in R_j} \sum_{v\in R_k} w(u,v) + w(v,u) \ .$$

Let $i = I(u)$ denote the index of the rectangle R_i with $u \in R_i$. Note that

$$\sum_i v_i \ \leq \ \sum_{u,v\in \bigcup_i R_i} w(u,v)|I(u) - I(v)| \ \leq \ \sum_{u,v} \frac{w(u,v)||u,v||_2}{\Delta} \ = \ W \ .$$

Hence, for at least one of the rectangles R_i we have $v_i \leq \frac{W}{p} \leq \frac{3C}{\ell(R_0)}$. □

Of course, this split can be found in polynomial time if the number of partitions is not too high. If we use $2p$ rectangles, then a random partition fullfills this property with probability of at least $\frac{1}{2}$. However, the number of sites n is a lower bound of the number of different values v_i. Using this observation one can find an algorithm that always determines such a split in polynomial time, even if Δ is arbitrarily small.

Theorem 3. *Given a set of sites V in k-dimensional Euclidean space and a non-negative weighting w such that the sum of all weights is polynomial in $n = |V|$; there exists a tree with a weighted distance that differs by the optimal Weighted Cost-Distance by at most a factor of $O(k \log n)$. Such a tree has size $O(c(MST(V))k \log n)$ and can be computed in polynomial time.*

Proof. We construct a fair split tree using the partition introduced in Lemma 3. We consider the vertex pair sets $P_1 := V_1^2$, $P_2 := V_2^2$, and $Q := V \times V \setminus (P_1 \cup P_2)$.

It holds for pairs in Q:

$$\sum_{(u,v)\in Q} w(u,v)L_T(u,v) \;\leq\; \frac{3D}{\ell(R)} L_T(u,v) \;\leq\; \frac{3D}{\ell(R)} 3k\ell(R) \leq 9kD \,,$$

where $D := \sum_{u,v} w(u,v)||u,v||_2$ is a lower bound for the weighted distance of the optimal network. For the disjoint pair sets P_1 and P_2 we apply this technique recursively for at most $O(ck \log n)$ rounds. As we have already observed, the length of the longest edge of the sub-rectangles is at most $\ell' := \frac{\ell(R_0)}{n^c}$. Then we face partitions $P_1, \ldots, P_m$ with partial weight sums $W_1, \ldots, W_m$ ($W_i := \sum_{(u,v)\in P_i} w(u,v)$). The sum of all weights $W := \sum_{u,v} w(u,v)$ is bounded by a polynomial $O(n^c)$. Therefore, $\sum_i W_i \leq W = O(n^c)$. The corresponding normalized weighted distances $\Delta_i := \sum_{u,v \in P_i} \frac{w(u,v)}{W_i} ||u,v||_2$ are bounded by ℓ_i, which is the length of the longest edge of the partition P_i's rectangle. Note that

$$\begin{aligned}\sum_{i\in P_i} L_T(u,v)w(u,v) \;&\leq\; 2\sum_i \ell_i w(u,v) \;\leq\; 2\sum_i \ell_i W_i \\ &\leq\; 2\ell' W \;\leq\; c'\ell(R_0) \;\leq\; c's(\mathrm{MST}(V))\end{aligned}$$

for a suitable constant c'. This and the recurrency over $O(k \log n)$ rounds imply

$$\begin{aligned}\sum_{u,v} w(u,v)L_T(u,v) &\leq c's(\mathrm{MST}(V)) + c''k(\log n)D \\ &\leq c'''k(\log n)\mathrm{WCD}_w(N)\end{aligned}$$

for a suitable constant c'' and c''' and every network N. □

3 A Lower Bound for Tree-Approximations

Trees cannot approximate the optimal Weighted Cost-Distance graph better than stated in Theorem 2. To show this, we construct a counter-example where the sites are uniformly distributed and the weighting supports only neighbored sites.

In particular, we consider an $n \times n$ unit square grid G and the following weighting function:

$$w(u,v) = \begin{cases} 1 : ||u,v||_2 = 1 \\ 0 : ||u,v||_2 \neq 1 \,. \end{cases}$$

Clearly, the weighted Cost-Distance of the grid consisting of all positive weighted edges is $O(n^2)$ and since the minimum spanning tree has at least cost $n^2 - 1$, this network is optimal up to a constant factor. We will show that every spanning tree T has weighted distance $\Omega(n^2 \log n)$ even if we allow T to use non-terminal vertices.

Let G_i be the set of vertices with distance $i-1$ to the convex hull of the grid, i.e. G_1 is the convex hull and G_{i+1} is the convex hull of $G \setminus \bigcup_{j \leq i} G_j$.

Lemma 4. *For every spanning tree T of the grid and for all $i \leq n/2$ there exist two grid neighbors $u, v \in G_i$ such that the connecting path in T has at least length $\frac{n}{2}$.*

Proof. Assume the contrary and consider the upper row of G_i. Note that neighbored vertices (in the grid) are connected by a path which is too short to reach the other half of the grid. Therefore in the upper row the leftmost and the rightmost vertex must be connected by a path, which is completely in the upper half of the rectangle.

For symmetry reasons the analogous property is true for the the left column, the lower row, and the right column. Therefore there exists a cycle that encloses the center of the grid, contradicting the tree property. □

Definition 2 (spanning cut). *A spanning cut splits a tree $T = (V, E)$ by a straight line s into trees $T_1 = (V_1 \cup S_1, E_1)$ and $T_2 = (V_2 \cup S_2, E_2)$. These sub-trees are entirely in the left or right half-space defined by s. All vertices in V_2 (resp. V_1) are orthogonally projected onto s and will be used as non-terminals S_1 in T_1 (resp. S_2 in T_2). All edges in trees T_1 and T_2 are copied from the original tree.*

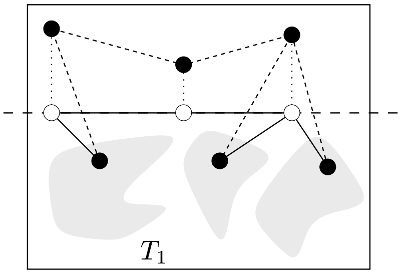

Fig. 4. A spanning cut and the resulting sub-tree in the lower halfspace

So, we copy every tree into both half spaces without increasing any edge length, for an example see Fig. 4.

Lemma 5. *For a spanning cut of T in to T_1 and T_2 we have for all $u_1, u_2 \in V(T_1)$ and $v_1, v_2 \in V(T_2)$:*

$L_G(u_1, u_2) \geq L_{G_1}(u_1, u_2, sincetheyreachedtheothersideofthegrid)$ *and* $L_G(v_1, v_2) \geq L_{G_2}(v_1, v_2)$

Theorem 4. *For every spanning tree T of the $n \times n$-grid, where $w(u, v) = 1$ if u and v are neighbored vertices and $w(u, v) = 0$ elsewhere, the weighted Cost-Distance is at least $\Omega(n^2 \log n)$, while the optimal Cost-Distance network has cost and weighted distance $O(n^2)$.*

Proof. We will split this grid into 16 sub-grids of size $\frac{n}{4} \times \frac{n}{4}$ by 15 spanning cuts (Fig. 5). By Lemma 5 the sum of the weighted distances of the sub-grids is a lower bound for the over-all grid (We also split the weightings into 16 local weightings).

Lemma 4 implies that in every subset G_i there are paths $p_1, \ldots, p_{n/2}$ between neighboring vertices with length of at least $n/2$. Furthermore, we can choose these paths such that the spanning cut reduces the lengths of all of them by at least $\frac{n}{4}$, since they reach the other side of the grid.

This way, we can account the length $\frac{n}{4}$ of these $\frac{n}{2}$ paths for this recursion level. This leads to the following recurrency for the weighted distance $W(n)$ of spanning trees of a $n \times n$-grid:

$$\mathrm{W}(n) \geq \frac{n^2}{8} + 16\,\mathrm{W}(n/4)\;.$$

Resolving this recurrency proves the claim. □

Applying the algorithm of Section 2 to this instance produces trees structured similar to the U-Layout shown in Fig. 6. Such trees optimize the weighted Cost-Distance of an $n \times n$ grid by a factor of $\Theta(\log n)$.

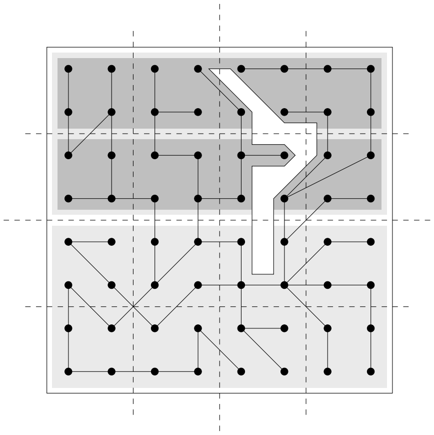

Fig. 5. The white marked p̀-shaped area induces long paths for a number of neighbored pairs. For the lower bound the grid is tiled into 16 sub-grids

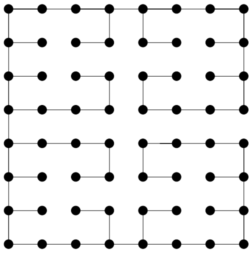

Fig. 6. The U-Layout approximates the Cost-Distance of this instance by a factor of $\Theta(\log n)$

4 Conclusions and Future Research

As an immediate implication of Theorem 3 we can state the following approximation result:

Corollary 3. *For polynomial weights the Weighted Cost-Distance-Tree problem can be polynomially approximated within a factor of $O(\log n)$.*

There is some hope that the approximation techniques introduced by Arora [Aro98] may lead to a polynomial time approximation scheme. Another follow-up result may be the extension to general metrics. We conjecture that the results of [Bar98] lead to an $O(\log^2 n)$ approximation.

An interesting open question is: if W, the sum of all weights, is super-polynomial, does the upper bound of section 3 also apply? Or can the lower bound factor be increased for such weightings? This mirrors the case in the original setting (Equation (1)) that the fixed costs are sub-polynomial compared to the linear costs.

Another extension of these results may be to consider different metrics for cost and distance as introduced in [MMP00]. They proved a $O(\log n)$-approximation for the two-metrics Cost-Distance problem with weights only on the root-vertex pairs. We have shown that for pairwise weight trees do not approximate better than $\Theta(\log n)$, while for vertex-root weightings Meyerson et al. [MMP00] showed that a tree is always part of the optimal solution. It is an interesting open question whether trees approximate this Weighted Cost-Distance problem with different metrics within a factor of $O(\log n)$.

References

[ADD$^+$95] S. Arya, G. Das, D.M.Mount, J.S. Slowe, and M. Smid. Euclidean spanners: Short, thin, and lanky. In *Proc. 27th ACM Symp. Theory of Computing*. ACM, 1995.

[Aro98] Sanjeev Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45(5):753–782, 1998.

[Bar98] Bartal. On approximating arbitrary metrics by tree metrics. In *ACM Symposium on Theory of Computing*, 1998.

[CCG$^+$98] M. Charikar, C. Chekuri, A. Goel, S. Guha, and S. Plotkin. Approximating a finite metric by a small number of tree metrics. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science*, 1998.

[DHN93] Gautam Das, Paul J. Heffernan, and Giri Narasimhan. Optimally sparse spanners in 3-dimensional euclidean space. In *Symposium on Computational Geometry*, pages 53–62, 1993.

[Epp00] David Eppstein. Spanning trees and spanners. In Jörg-Rudiger Sack and Jorge Urrutia, editors, *Handbook of Computational Geometry*, chapter 9, pages 425–461. Elsevier Science Publishing, 2000.

[GGJ76] M. R. Garey, R. L. Graham, and David S. Johnson. Some NP-complete geometric problems. In *ACM Symposium on Theory of Computing*, pages 10–22, 1976.

[KRY95] S. Khuller, B. Raghavachari, and N. Young. Balancing minimum spanning trees and shortest path trees. In *Algorithmica, 14*, pages 305–321, 1995.

[LL89] C. Levcopoulos and A. Lingas. There are planar graphs almost as good as the complete graphs and as short as minimum spanning trees. In *Proc. Internat. Symp. on Optimal Algorithms, volume 401 of LNCS*, pages 9–13. Springer-Verlag, 1989.

[MMP00] A. Meyerson, K. Munagala, and S. Plotkin. COST-DISTANCE: Two metric network design. In *Proc. 41st Symp. Foundations of Computer Science*. IEEE, 2000.

[Web01] Birgitta Weber. Netzwerke optimiert für lokale Kosten-Distanzen. Lübeck University, Diploma Thesis, 2001.