6 Randomized metric reduction

In this chapter we are going to examine a randomized technique to embed an arbitrary metric into a tree-metric with low distortion. The technique presented here, which is based on Bartals work, was developed by Fakcharoenphol, Rao and Talwar [4] and is suitable for a large class of combinatorial optimization problems. For all of the applications presented here, no better approximation algorithms are known so far.

6.1 Notation

A *metric* (V, d) is defined by a set of points V (also called *nodes*) and a distance measure d with the following properties

1. $d(v, v) = 0$ for all $v \in V$,
2. $d(v, w) > 0$ for all $v, w \in V$ with $v \neq w$,
3. $d(v, w) = d(w, v)$ for all $v, w \in V$ (symmetry), and
4. $d(u, w) \leq d(u, v) + d(v, w)$ for all $u, v, w \in V$ (triangle inequality).

W.l.o.g. let the minimum distance of two nodes be 1, and let Δ be the *diameter* of the metric (i.e., the maximum distance of all pairs of nodes). Further, we assume w.l.o.g. that $\Delta = 2^\delta$ for some $\delta \in \mathbb{N}$.

A metric (V, d') *dominates* another metric (V, d) if for all $v, w \in V$, $d'(v, w) \geq d(v, w)$. The goal is to find a dominating tree metric for any given metric.

Let $\mathcal{S}$ be a family of metrics over V, and let $\mathcal{D}$ be a probability distribution over $\mathcal{S}$. We say that $(\mathcal{S}, \mathcal{D})$ *approximates* metric (V, d) *α-probabilistically* if every metric in $\mathcal{S}$ dominates (V, d) and for every pair u, v of nodes in V it holds that $\mathbb{E}_{d' \in (\mathcal{S}, \mathcal{D})}[d'(u, v)] \leq \alpha \cdot d(u, v)$.

An *r-decomposition* of (V, d), with $r \in \mathbb{N}$, is a partition of V into groups such that for every group G there is a node $v \in V$ with $d(v, w) < r$ for all $w \in G$ (i.e., the *radius* of the group is less than r and therefore its diameter is less than $2r$). A *hierarchical decomposition* of (V, d) is a series of $\delta + 1$ decompositions $D_0, D_1, \ldots, D_\delta$ with the property that

- $D_\delta = \{V\}$ is the trivial partition (all nodes are in one group), and
- D_i is a 2^i-decomposition and refinement of D_{i+1} (i.e., groups in D_{i+1} are divided into further subgroups).

Each group in D_0 has radius less than 1 and therefore consists of a single node.

6.2 From decompositions to trees

A hierarchical decomposition defines a laminar family (i.e., a set of subsets $\mathcal{F} \subseteq 2^V$ with the property that for all $A, B \in \mathcal{F}$, $A \subseteq B$ or $B \subseteq A$ or $A \cap B = \emptyset$) and can be represented by a *decomposition tree* as follows. For every i, every group $G \in D_i$ represents a node in that tree and the children of G are all groups $G' \in D_{i-1}$ that are contained in G. The root is the node representing V while the leaves are formed by groups containing only a single node (cf. Fig. 1).

Let the edges of a node $S \in D_i$ to any of its children in the decomposition tree T have length 2^i (which is an upper bound for the radius of S). This induces a distance function $d_T(\cdot, \cdot)$ on V with $d_T(v, w)$ being equal to the length of the unique path from the node $\{v\} \in D_0$ to the node $\{w\} \in D_0$ in T. It is not difficult to check that d_T is a metric. Further, $d_T(v, w) \geq d(v, w)$ for all $v, w \in V$ since the least common ancestor of v and w in T must represent a set with diameter at least $d(v, w)$. In the following we will prove upper bounds for $d_T(v, w)$ as well. A pair (v, w) is *at level i* if v and w appear the last time together in a group $G \in D_i$. If (v, w) is at level i, then $d_T(v, w) = 2 \sum_{j=1}^{i} 2^j \leq 2^{i+2}$.

6.3 Decomposition of the set of nodes

Consider the following random experiment to create a hierarchical decomposition of (V, d), where $V = \{v_1, \ldots, v_n\}$. Choose a permutation π uniformly at random out of the set of all permutations of $\{1, \ldots, n\}$, and choose β uniformly at random in $[1, 2]$. Then, for every i, we compute D_i out of D_{i+1} as follows.

Set $\beta_i := 2^{i-1}\beta$. Let S be a group in D_{i+1}. Every node $u \in S$ gets assigned to the first node $v \in V$ (regarding π) which is closer than β_i to u. This node is declared as u's *center*. In this way, S is cut into several groups in D_i. Note that the center of a group S does not have to be part of S and that there might be several groups in D_i with the same center,

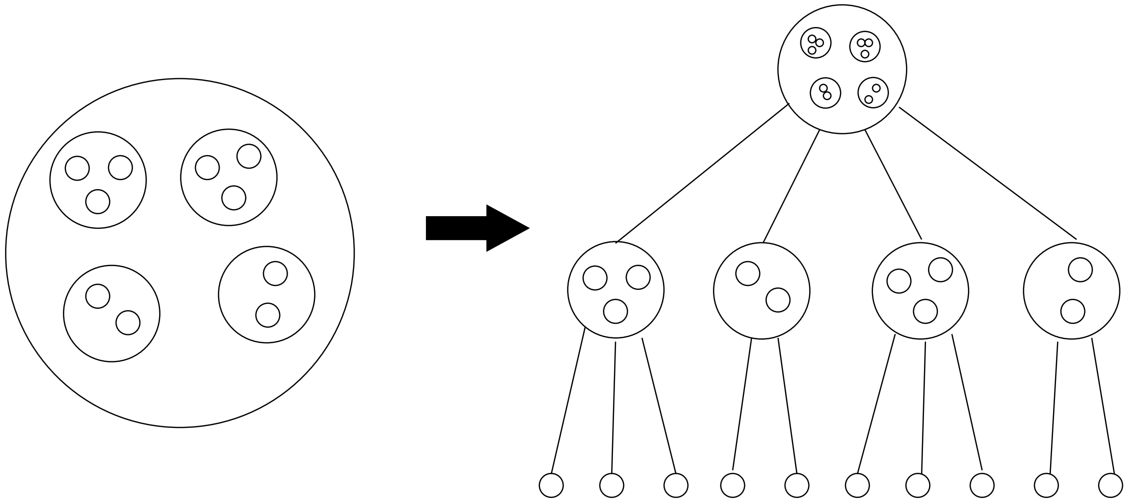

Figure 1: From a laminar family to a decomposition-tree.

which is the case if the nodes already belong to different groups in D_{i+1}. Furthermore, $\beta_i \leq 2^i$ and therefore the radius of all groups in D_i is less than 2^i which leads to a 2^i-decomposition. The formal decomposition algorithm is shown in Figure 2. We note that a group may stop participating in lower levels once it has reached a size of 1, so it will be a leaf of the decomposition tree representing the node it consists of.

Algorithm Partition(V, d)**:**
choose a random permutation π of $\{1, \dots, n\}$
choose β uniformly at random from $[1, 2]$
$D_\delta := \{V\}$; $i := \delta - 1$
while D_{i+1} contains a group with more than one node **do**
    $\beta_i := 2^{i-1}\beta$
    for $\ell := 1$ **to** n **do**
        for every $S \in D_{i+1}$ **do**
            create a new group with all thus far unassigned nodes in S
            which are closer to $v_{\pi(\ell)}$ than β_i
    $i := i - 1$

Figure 2: The partitioning algorithm

Algorithm 2 can be implemented in a straight-forward way with runtime $O(n^3)$. With specific data structures one can decrease the runtime to $O(n^2)$, which is linear in the input size since d usually needs complexity $\Theta(n^2)$ to be described properly.

Fix a pair (u, v). Now, we show that the expectation of $d_T(u, v)$ is bounded by $O(d(u, v) \log n)$. Considering the discussion above we get

$$\mathbb{E}[d_T(u, v)] \leq \sum_{i=0}^{\delta} \mathbb{P}[(u, v) \text{ is at level } i] \cdot 2^{i+2}.$$

Certainly, if $d(u, v) \geq 2^{i+1}$, nodes u and v cannot be contained in the same group in D_i. In other words, (u, v) cannot be at level i. Let i^* be the smallest i with $d(u, v) < 2^{i+1}$. Then $\mathbb{P}[(u, v) \text{ is at level } i] = 0$ for all $i < i^*$. Thus, it remains to bound this probability for $i \geq i^*$. For any $i^* \leq j \leq \delta$ let K_j^u be the set of nodes in V which are closer than 2^j to node u. Further, let $k_j^u = |K_j^u|$. (We set $k_j^u = 0$ for $j < i^*$.)

Consider some fixed $i \geq i^*$. We say that $v_{\pi(\ell)}$ *decides* the pair (u, v) at level i if it is the first center that node u or v is assigned to at level i. Note that once π and β are fixed, this center is unique and well defined. Further, we say that

$v_{\pi(\ell)}$ *cuts* the pair (u, v) at level i if it decides (u, v) at level i and exactly one node from u and v gets assigned to $v_{\pi(\ell)}$. Obviously, if (u, v) is at level $i + 1$, then there must be a node w that cuts (u, v) in level i. Therefore it holds

$$\begin{aligned}\mathbb{P}[(u,v) \text{ is at level } i+1] &= \mathbb{P}[\exists w : w \text{ cuts } (u,v) \text{ at level } i] \\ &\leq \sum_w \mathbb{P}[w \text{ cuts } (u,v) \text{ at level } i].\end{aligned}$$

We say that a center w *cuts* node u from (u, v) at level i if w cuts the pair (u, v) and u is being assigned to w. For each center w we limit the probability for w to cut u from (u, v) at level i. For this we order the centers in K_i^u in ascending distance to u. Suppose this order is given by $w_1, w_2, \ldots, w_{k_i^u}$. In this case, a center w_s is able to cut u from (u, v) only if the following holds:

1. $d(u, w_s) < \beta_i$,
2. $d(v, w_s) \geq \beta_i$, and
3. w_s decides (u, v).

From the first two requirements it follows that β_i must be in the interval $[d(u, w_s), d(v, w_s)]$. Due to the triangle inequality it holds $d(v, w_s) \leq d(v, u) + d(u, w_s)$ and therefore the length of the interval $[d(u, w_s), d(v, w_s)]$ is at most $d(u, v)$. Since β_i is chosen uniformly at random from $[2^{i-1}, 2^i]$, the probability for β_i to lie in the said interval is at most $d(u, v)/2^{i-1}$.

Next, we can deduce a probability from requirement (3). Due to the definition of K_i^u it holds that $d(u, w_s) < \beta_i$ and therefore $d(u, w_{s'}) < \beta_i$ for all $s' \leq s$. The probability that (u, v) is decided by center w_s is at most $1/s$ since π is a random permutation.

Note that the first probability bound only depends on β while the second one only depends on the choice of π. Thus, both probability bounds hold independently and we obtain the following inequalities.

$$\begin{aligned}\mathbb{P}[(u,v) \text{ is at level } i+1] &\leq \sum_{s=1}^{k_i^u} (d(u,v)/2^{i-1}) \cdot \frac{1}{s} + \sum_{s=1}^{k_i^v} (d(u,v)/2^{i-1}) \cdot \frac{1}{s} \\ &\leq \frac{d(u,v)}{2^{i-1}} (\ln k_i^u + 1 + \ln k_i^v + 1) \leq \frac{d(u,v)(\ln n + 1)}{2^{i-2}}\end{aligned}$$

Hence,

$$\begin{aligned}\mathbb{E}[d_T(u,v)] &\leq \sum_{i=0}^{\delta} \mathbb{P}[(u,v) \text{ is at level } i] \cdot 2^{i+2} \\ &\leq \sum_{i=i^*}^{\delta} \frac{d(u,v)(\ln n + 1)}{2^{i-3}} \cdot 2^{i+2} = O(\delta \log n \cdot d(u,v)).\end{aligned}$$

Thus, the expected length of $d_T(u, v)$ is in $O(\log \Delta \cdot \log n \cdot d(u, v))$.

To show the bound of $O(\log n)$ we observe that the amount of centers over all δ levels is n. A more detailed analysis of the procedure above will then provide the desired result, as shown next.

Let us fix a $i \geq i^* + 3$. Due to the definition of i^* it follows that $d(u, v) < 2^{i-2}$. Additionally, for any $w \in K_{i-2}^u$ it holds $d(v, w) \leq d(v, u) + d(u, w) < 2^{i-2} + 2^{i-2} = 2^{i-1} \leq \beta_i$. Hence, w cannot be the center cutting u from (u, v) since this would require the three requirements above to be fulfilled. Therefore, no center of $w_1, w_2, \ldots, w_{k_{i-2}^u}$ is able to cut u from (u, v) at level i. It follows that the probability for u to be cut from (u, v) is at most

$$\sum_{s=k_{i-2}^u+1}^{k_i^u} (d(u,v)/2^{i-1}) \cdot \frac{1}{s} = (d(u,v)/2^{i-1}) \cdot (H_{k_i^u} - H_{k_{i-2}^u})$$

where $H_n = \sum_{i=1}^{n} \frac{1}{i}$ is the harmonic number. Since (u, v) is cut if either u or v gets cut from (u, v), the probability for the pair (u, v) to be cut in level i is upper bounded by

$$\frac{d(u,v)}{2^{i-1}} \cdot [H_{k_i^u} + H_{k_i^v} - H_{k_{i-2}^u} - H_{k_{i-2}^v}].$$

For $i \in \{i^*, \ldots, i^*+2\}$ we can bound this probability by the formula

$$\frac{d(u,v)}{2^{i-1}} \cdot (H_{k_i^u} + H_{k_i^v}) \leq \frac{d(u,v)}{2^{i-1}} \cdot 2H_n.$$

The expectation of $d_T(u,v)$ is therefore

$$\begin{aligned}
\mathbb{E}[d_T(u,v)] &\leq \sum_{i=0}^{\delta} \mathbb{P}[(u,v) \text{ is at level } i] \cdot 2^{i+2} \\
&\leq \sum_{i=i^*}^{i^*+2} 2H_n \cdot \frac{d(u,v)}{2^{i-1}} \cdot 2^{i+2} \\
&\quad + \sum_{i=i^*+3}^{\delta} [H_{k_i^u} + H_{k_i^v} - H_{k_{i-2}^u} - H_{k_{i-2}^v}] \cdot \frac{d(u,v)}{2^{i-1}} \cdot 2^{i+2} \\
&\leq 8d(u,v)(3 \cdot 2H_n + H_{k_\delta^u} + H_{k_\delta^v} + H_{k_{\delta-1}^u} + H_{k_{\delta-1}^v}) \\
&\leq 8d(u,v) \cdot 10H_n \\
&\leq 80(\ln n + 1) \cdot d(u,v).
\end{aligned}$$

This shows that the expected value of $d_T(u,v)$ is at most $O(d(u,v) \cdot \log n)$ for any pair (u,v). Hence, it holds:

Theorem 6.1 *The probability distribution over the tree metric defined by the partitioning algorithm $O(\log n)$-probabilistically approximates metric d.*

6.4 Applications

Many problems are much easier to solve in tree metrics than in others. A few of these are presented below.

The k-median problem

An instance of the k-median problem consists of a set of points $V = \{v_1, \ldots, v_n\}$ and a metric d. The goal is to find a set $M \subseteq V$ of k median points such that the sum of the distances of all nodes to its closest median-points is minimal, i.e.

$$\sum_{i=1}^{n} \min_{w \in M} d(v_i, w).$$

For trees we know optimal algorithms. In the case of a tree-metric we assume that we are given an undirected graph $G = (V, E)$ with edge lengths $c : E \to \mathbb{R}_+$, where G represents a tree, and the distance $d(u,v)$ for an arbitrary pair $u, v \in V$ is defined as the length of the unique path from u to v in G. For this case Tamir [6] presented a precise algorithm, which is based on dynamic programming and runs in time $O(k \cdot n^2)$. If k is constant, even precise algorithms with runtime $O(n \cdot polylog(n))$ are known [2]. Hence, we obtain the following result.

Theorem 6.2 *With Tamir's algorithm one can solve the k-median problem for arbitrary metrics in time $O(k \cdot n^2)$ with an expected approximation ratio of $O(\log n)$.*

Proof. Consider the following algorithm:

Given an arbitrary instance (V, d) where d is a metric, reduce d to a tree metric d' using algorithm Partition(V, d), solve the problem on d' using Tamir's algorithm, and return the objective value obtained by that algorithm.

As we will show, this algorithm has an expected approximation ratio of $O(\log n)$, which proves the theorem. For a given metric d let

$$OPT_d = \min_{M \subseteq V, |M|=k} \sum_{i=1}^{n} \min_{w \in M} d(v_i, w)$$

be the optimal value of the k-median problem regarding this metric. Let $\mathcal{B}$ be a family of tree metrics over V and $\mathcal{D}$ a probability distribution over $\mathcal{B}$. Assume $(\mathcal{B}, \mathcal{D})$ approximates (V, d) α-probabilistically. Then it holds for any $d' \in \mathcal{B}$ that

(V, d') dominates (V, d) and thus $OPT_{d'} \geq OPT_d$. Furthermore, for the optimal set of medians M concerning d it holds that $OPT_{d'} \leq \sum_{i=1}^{n} \min_{w \in M} d'(v_i, w)$. Hence,

$$\begin{aligned} \mathbb{E}[OPT_{d'}] &\leq \mathbb{E}\left[\sum_{i=1}^{n} \min_{w \in M} d'(v_i, w)\right] \\ &= \sum_{i=1}^{n} \mathbb{E}[\min_{w \in M} d'(v_i, w)] \\ &\overset{(*)}{\leq} \sum_{i=1}^{n} \min_{w \in M} \mathbb{E}[d'(v_i, w)] \\ &\leq \sum_{i=1}^{n} \min_{w \in M} \alpha \cdot d(v_i, w) = \alpha \cdot OPT_d. \end{aligned}$$

Inequality $(*)$ follows since it is known that for any matrix $A = (a_{i,j}) \in \mathbb{R}^{(m,k)}$,

$$\sum_{i=1}^{m} \min\{a_{i,1}, \ldots, a_{i,k}\} \leq \min\left\{\sum_{i=1}^{m} a_{i,1}, \ldots, \sum_{i=1}^{m} a_{i,k}\right\}.$$

Hence, $\mathbb{E}[OPT_{d'}] \in [OPT_d, \alpha \cdot OPT_d]$. Therefore, the expected approximation ratio of our algorithm is $\alpha = O(\log n)$.

If a k-median set is required instead of a number, we can just output the median set M' found for d', because due to the fact that d' dominates d it holds that

$$\sum_{i=1}^{n} \min_{w \in M'} d(v_i, w) \leq \sum_{i=1}^{n} \min_{w \in M'} d'(v_i, w) = OPT_{d'}$$

so the objective value for M' w.r.t. d is at most as high as the objective value for M' w.r.t. d', which means that on expectation, it is still at most $O(OPT_d \log n)$. □

The group-Steiner-tree problem

An instance of the group-Steiner-tree problem consists of a connected undirected graph $G = (V, E)$ with edge costs given by $c : E \to \mathbb{R}_+$ and k subsets $V_1, \ldots, V_k \subseteq V$. The goal is to find a tree $T = (V', E')$ in G containing at least one element of each subset and having minimum edge costs $\sum_{e \in E'} c(e)$.

Garg, Konjevod and Ravi [5] presented a $O(\log k \log n)$-approximation algorithm for trees, which implies the following result for arbitrary graphs.

Theorem 6.3 *Using the GKR-algorithm one can solve the group-Steiner-tree problem for arbitrary graphs in polynomial time with an expected approximation ratio of* $O(\log k \log^2 n)$.

Proof. Let us use the same approach as in the previous problem:

Given an arbitrary instance $(G, c, V_1, \ldots, V_k)$, define $d(v, w)$ as the length of the shortest path from v to w in G with respect to the edge costs c. Then reduce d to a tree metric d' using algorithm Partition(V, d), where d' represents the shortest path metric in the decomposition tree $DT = (V', E')$. Let $c' : E' \to \mathbb{N}$ denote the costs of the edges of DT as defined in Section 5.2. Then we use the GKR-algorithm to solve the group-Steiner-tree problem for $(DT, c', V_1, \ldots, V_k)$ where the sets V_i refer to the singletons at level D_0 in DT, and return the objective value obtained by that algorithm.

As we will show, this algorithm has an expected approximation ratio of $O(\log k \log^2 n)$, which proves the theorem. Let $T = (U, F)$ be the optimal group-Steiner-tree in G, and let T be organized in a unique way from some fixed node $r \in U$, which we declare as its root. For every $i \in \{1, \ldots, k\}$, let $v_i \in U$ be the first node in V_i encountered in T when performing an inorder traversal of T. Certainly, there must be such a node for each i, otherwise T would not be a group-Steiner-tree. Also, all leaves in T must be one of the v_i's because otherwise T would be reducible. Suppose for simplicity that the v_i's are visited by the inorder traversal in the order $v_1, v_2, \ldots, v_k$. Let $p(v, w)$ be the unique path from v to w in T, and let

$c(p(v,w))$ be sum of the costs of the edges in p. Since the paths $p(v_1,v_2)$, $p(v_2,v_3)$, …, $p(v_{k-1},v_k)$, $p(v_k,v_1)$ stitched together give an Euler tour of T, it holds for $v_{k+1}=v_1$ that

$$\sum_{i=1}^{k} c(p(v_i,v_{i+1})) = 2\sum_{e\in F} c(e)$$

On the other hand, $c(p(v_i,v_{i+1})) \geq d(v_i,v_{i+1})$, so

$$\sum_{i=1}^{k} c(p(v_i,v_{i+1})) \geq \sum_{i=1}^{k} d(v_i,v_{i+1})$$

which implies that

$$\sum_{i=1}^{k-1} d(v_i,v_{i+1}) \leq 2\sum_{e\in F} c(e).$$

Moreover, the union of the edges on the shortest paths for the pairs (v_i,v_{i+1}) results in a connected subgraph of G with costs at least equal to the ones of T. Hence,

$$\sum_{e\in F} c(e) \leq \sum_{i=1}^{k-1} d(v_i,v_{i+1})$$

Therefore, altogether,

$$\sum_{e\in F} c(e) \leq \sum_{i=1}^{k-1} d(v_i,v_{i+1}) \leq 2\sum_{e\in F} c(e).$$

Now, let $T'=(U',F')$ be the optimal group-Steiner-tree in the decomposition tree DT, and let $w_1,\ldots,w_\ell$ be its leaves. Obviously, each leaf must belong to some group V_i, and each group V_i has at most one leaf in T because otherwise T' can be reduced. Also, there cannot be any inner nodes of DT that belong to some V_i since the nodes in V are mapped to the leaves of DT. Hence, $\ell=k$. For simplicity, suppose that $w_i\in V_i$.

Using the inequalities for T and the fact that d' dominates d, it holds that

$$\begin{aligned}\sum_{e\in F'} c'(e) &\geq \frac{1}{2}\sum_{i=1}^{k-1} d'(w_i,w_{i+1}) \geq \frac{1}{2}\sum_{i=1}^{k-1} d(w_i,w_{i+1})\\ &\geq \frac{1}{2}\sum_{e\in F} c(e).\end{aligned}$$

Thus, the cost of T' regarding d' is at least as high as half the cost of an optimal group-Steiner-tree in G. Furthermore, for the unique minimum tree $T''=(U'',F'')$ connecting the nodes $v_i,\ldots,v_k$ in DT it holds that

$$\begin{aligned}\mathbb{E}[\sum_{e\in F''} c'(e)] &\leq \mathbb{E}[\sum_{i=1}^{k-1} d'(v_i,v_{i+1})]\\ &= \sum_{i=1}^{k-1}\mathbb{E}[d'(v_i,v_{i+1})]\\ &\leq \sum_{i=1}^{k-1}\alpha d(v_i,v_{i+1})\\ &\leq 2\alpha\sum_{e\in F} c(e).\end{aligned}$$

Now, let T_{GKR} be the tree obtained by the GKR-algorithm in DT. Since the GKR-algorithm ensures that for the optimal tree T' in DT, $\sum_{e\in T_{\mathrm{GKR}}} c'(e) \leq \beta\sum_{e\in F'} c'(e) \leq \beta\sum_{e\in F''} c'(e)$, where $\beta = O(\log k\log n)$, we observe that

$$\mathbb{E}[\sum_{e\in T_{\mathrm{GKR}}} c'(e)] \in [\frac{1}{2}\sum_{e\in F} c(e),\ 2\alpha\beta\sum_{e\in F} c(e)].$$

Therefore, we obtain a $O(\log k \log^2 n)$-approximation.

If instead of the objective value we want the group-Steiner-tree as output of our algorithm, we simply output any tree $\hat{T} = (\hat{U}, \hat{F})$ in G containing the k leaves $w_1, \ldots, w_k$ of T_{GKR}, where $\hat{T}$ can be obtained from the subgraph resulting from the union of the shortest paths for the pairs (w_1, w_2), (w_2, w_3), …, (w_{k-1}, w_k), (w_k, w_1) in G. For this tree we get

$$\sum_{e \in \hat{F}} c(e) \leq \sum_{i=1}^{k} d(w_i, w_{i+1}) \quad \leq \quad \sum_{i=1}^{k} d'(w_i, w_{i+1}) \leq 2 \sum_{e \in T_{\mathrm{GKR}}} c'(e).$$

So on expectation, the cost of $\hat{T}$ is at most $O(OPT_d \log k \log^2 n)$. □

Buy en bloc network design

A problem instance consists of an undirected graph $G = (V, E)$ with edge lengths $\ell : E \to \mathbb{R}_+$ and a set of source-target-pairs (s, t) with flow demands $d(s, t)$. For each source-target-pair a path through G must be chosen that can accommodate the demand. One achieves this by buying/renting cables along the edges. Exactly k types of cable exist, where type i has capacity u_i and cost c_i per unit of length. The goal is to buy/rent enough cable such that a flow of $d(s, t)$ is possible for every source-target-pair (s, t) with costs as low as possible.

The problem can easily be solved within an $O(1)$-factor in a tree since the optimal strategy is to route the flow along the unique path from s to t for every source-target-pair (s, t), so it only remains to find the best setup of cables for every edge, which can be seen as a variant of the knapsack problem that can be solved within an $O(1)$-factor. One can even design $O(1)$-competitive online algorithms for this problem, as shown by Awerbuch and Azar [1]. Consequently, we obtain the following theorem.

Theorem 6.4 *By using the Awerbuch-Azar algorithm one can solve the buy en bloc network design problem for arbitrary graphs in polynomial time with an expected approximation ratio of* $O(\log n)$.

Vehicle routing

A problem instance consists of a metric (V, d). In this metric, n objects are placed which need to be transported to n target points. This is done by a waggon driving from point to point in V with a cargo capacity of k objects. The goal is to minimize the overall path length of the waggon needed to deliver all objects.

Charikar et al. [3] presented an $O(1)$-approximation algorithm for trees. Consequently, we obtain the following theorem.

Theorem 6.5 *By using the CCGG-algorithm one can solve the vehicle routing problem for arbitrary graphs in polynomial time with an expected approximation ratio of* $O(\log n)$.

References

[1] B. Awerbuch and Y. Azar. Buy-at-bulk network design. In *Proc. of the 38th IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 542–547, 1997.

[2] R. Benkoczi and B. Bhattacharya. A new template for solving p-median problems for trees in sub-quadratic time. In *Proc. of the European Symposium on Algorithms (ESA)*, pages 271–282, 2005.

[3] M. Charikar, C. Chekuri, A. Goel, and S. Guha. Rounding via trees: Deterministic approximation algorithms for group Steiner trees and k-median. In *Proc. of the 30th ACM Symp. on Theory of Computing (STOC)*, pages 114–123, 1998.

[4] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proc. of the 35th ACM Symp. on Theory of Computing (STOC)*, pages 448–455, 2003.

[5] N. Garg, G. Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group Steiner tree problem. *Journal of Algorithms*, 37:66–84, 2000.

[6] A. Tamir. An $o(pn^2)$ algorithm for the p-median and related problems on tree graphs. *Operations Research Letters*, 19(2):59–64, 1996.