2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS) | 979-8-3503-1894-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/FOCS57990.2023.00012

One Tree to Rule Them All: Poly-Logarithmic Universal Steiner Tree

Costas Busch*, Da Qi Chen†, Arnold Filtser‡,
Daniel Hathcock§, D Ellis Hershkowitz¶‖ and Rajmohan Rajaraman**
*School of Computer and Cyber Sciences
Augusta University, Augusta, Georgia
Email: kbusch@augusta.edu
†Biocomplexity Institute and Initiative
University of Virginia, Charlottesville, Virginia
Email: daqichen02@gmail.com
‡Department of Computer Science
Bar-Ilan University, Ramat Gan, Israel
Email: arnold.filtser@biu.ac.il
§Department of Mathematical Sciences
Carnegie Mellon University, Pittsburgh, Pennsylvania
Email: dhathcoc@andrew.cmu.edu
¶Computer Science Department
Brown University, Providence, Rhode Island
‖Department of Mathematics
ETH Zürich, Zürich, Switzerland
Email: delhersh@gmail.com
**Khoury College of Computer Sciences
Northeastern University, Boston, Massachusetts
Email: r.rajaraman@northeastern.edu

***Abstract*—A spanning tree T of graph G is a ρ-approximate *universal Steiner tree* (UST) for root vertex r if, for any subset of vertices S containing r, the cost of the minimal subgraph of T connecting S is within a ρ factor of the minimum cost tree connecting S in G. Busch et al. (FOCS 2012) showed that every graph admits $2^{O(\sqrt{\log n})}$-approximate USTs by showing that USTs are equivalent to strong sparse partition hierarchies (up to poly-logs). Further, they posed poly-logarithmic USTs and strong sparse partition hierarchies as open questions.**

We settle these open questions by giving polynomial-time algorithms for computing both $O(\log^7 n)$-approximate USTs and poly-logarithmic strong sparse partition hierarchies. We reduce the existence of these objects to the previously studied cluster aggregation problem and a class of well-separated point sets which we call dangling nets. For graphs with constant doubling dimension or constant pathwidth we obtain improved bounds by deriving $O(\log n)$-approximate USTs and $O(1)$ strong sparse partition hierarchies. Our doubling dimension result is tight up to second order terms.

Busch supported by National Science Foundation grant CNS-2131538. Filtser supported by the Israel Science Foundation (grant No. 1042/22). Hathcock supported by the National Science Foundation Graduate Research Fellowship under grant No. DGE-2140739. Hershkowitz funded by the SNSF, Swiss National Science Foundation grant 200021_184622. Rajaraman supported by National Science Foundation grant CCF-1909363.

I. Introduction

Consider the problem of designing a network that allows a server to broadcast a message to a single set of clients. If sending a message over a link incurs some cost then designing the best broadcast network is classically modelled as the Steiner tree problem [HR92]. Here, we are given an edge-weighted graph $G = (V, E, w)$, terminals $S \subseteq V$ and our goal is a subgraph $H \subseteq G$ connecting S of minimum weight $w(H) := \sum_{e \in H} w(e)$. We let OPT_S be the weight of an optimal solution.

However, Steiner trees fail to model the fact that a server generally broadcasts different messages to different subsets of clients over time. If building network links is slow and labor-intensive, we cannot simply construct

979-8-3503-1894-4/23/$31.00 ©2023 IEEE

DOI 10.1109/FOCS57990.2023.00012

Family	Approximation	Ref.
Complete Graphs		
General	$O(\log^2 n)$	[GHR06]
	$\Omega(\log n)$	[JLN+05]
Planar	$O(\log n)$	[BLT14]
	$\tilde{\Omega}(\log n)$	[JLN+05]
Doubling	$\tilde{O}(d^3) \cdot \log n$	[Fil20]
	$\tilde{\Omega}(\log n)$	[JLN+05]
Pathwidth	$O(\mathrm{pw} \cdot \log n)$	[Fil20]
General Graphs		
General	$2^{O(\sqrt{\log n})}$	[BDR+12]
	$O(\log^7 n)$	Thm. VI.2
Planar	$O(\log^{18} n)$	[BDR+12]
Doubling	$\tilde{O}(d^7) \cdot \log n$	Full Ver. [BCF+23]
Pathwidth	$O(\mathrm{pw}^8 \cdot \log n)$	Full Ver. [BCF+23]

Fig. 1: A summary of UST work.

new links each time a new broadcast must be performed. Rather, in such situations we must understand how to construct a *single network* in which the broadcast cost from a server is small for every subset of clients. Ideally, we would like our network to be a tree since trees have a simple routing structure. Our goals are similar if our aim is to perform repeated aggregation of data of different subsets of clients. Motivated by these settings, Jia et al. [JLN+05] introduced the idea of universal Steiner trees (USTs), defined below and illustrated in Figure 2.

Definition I.1 (ρ-Approximate Universal Steiner Tree)**.** *Given an edge-weighted graph $G = (V, E, w)$ and root $r \in V$, a ρ-approximate universal Steiner tree is a spanning tree $T \subseteq G$ such that for every $S \subseteq V$ containing r, we have*

$$w(T\{S\}) \leq \rho \cdot OPT_S$$

where $T\{S\} \subseteq T$ is the minimal subtree of T connecting S, and OPT_S is the minimum weight Steiner tree connecting S in G.

Known UST results are given in Figure 1. Surprisingly, it is known that every n-vertex graph admits a $2^{O(\sqrt{\log n})}$-approximate and poly-time-computable UST, as proven by [BDR+12] more than a decade ago. On the other hand, the best known lower bound is $\rho \geq \Omega(\log n)$ [JLN+05]. In fact, even when G is the complete graph whose distances are induced by an $\sqrt{n} \times \sqrt{n}$ grid, there is an $\Omega(\log n / \log \log n)$ lower bound. Improved upper bounds are known for several special cases: fixed minor-free (e.g. planar) graphs admit $O(\log^{18} n)$-approximate USTs [BDR+12]. Complete graphs induced by a metric admit $O(\log^2 n)$-approximate USTs [GHR06]. If the inducing metric has doubling dimension d, then the complete graph admits $O(d^3 \cdot \log n)$-approximate USTs [Fil20]. Furthermore, better bounds are known for complete graphs when the inducing metric is the shortest path metric of a restricted graph H: if H is planar or has pathwidth pw then the complete graph admits $O(\log n)$- [BLT14] and $O(\mathrm{pw} \cdot \log n)$- [Fil20] approximate USTs, respectively.

Thus, for general graphs, there is a huge gap between the upper and lower bounds of $2^{O(\sqrt{\log n})}$ and $\Omega(\log n)$. Closing this gap has been posed as an open question [BDR+12, JLN+05, Fil20].

Our main result is a poly-time $O(\log^7 n)$-approximate UST, settling this open question.

Furthermore, if G has constant doubling dimension or pathwidth, we provide an $O(\log n)$-approximate UST. The doubling dimension result is tight up to second order terms [JLN+05].

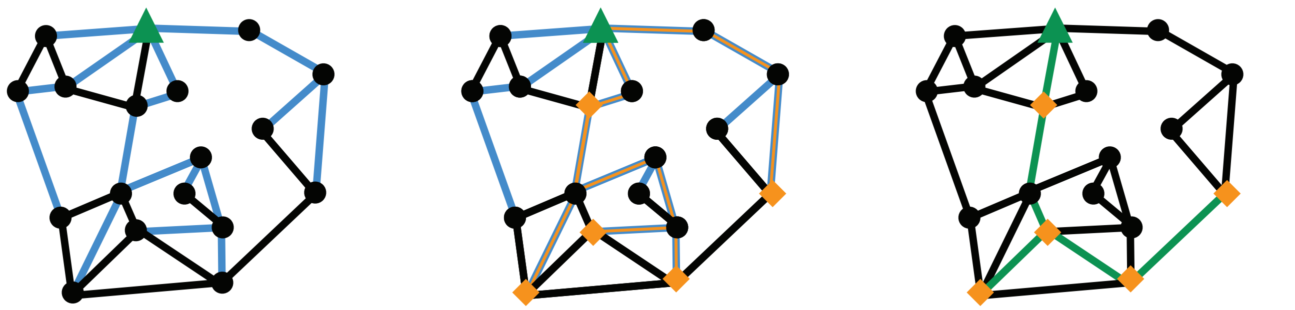
We obtain our results by proving the existence of certain graph hierarchies—strong sparse partition hierarchies— and leveraging a previously-established connection between these hierarchies and USTs. We prove the existence of these hierarchies, in turn, by reducing their existence to two objects: (1) low distortion solutions to the (previously studied) cluster aggregation problem, and (2) a certain kind of net which we call dangling nets that provide *additive* sparsity guarantees. The existence of these nets can be inferred from an analysis in [Fil20] of the random-shift techniques of [MPX13]. For cluster aggregation, we improve the best bounds in general graphs from $O(\log^2 n)$ [BDR+12] to $O(\log n)$ and prove $O(1)$ bounds for trees, constant doubling dimension and constant pathwidth graphs. Our results are summarized in Table I. We spend the rest of this section describing them in greater detail.[1]

A. Poly-Logarithmic USTs

As mentioned, our main result is a polynomial-time computable $O(\log^7 n)$-approximate UST in general graphs. Not only is this an exponential improvement for general graphs[2], it significantly improves upon the best bounds known for planar graphs (previously $O(\log^{18} n)$ [BDR+12]).

[1] We make use of standard graph notation and concepts throughout this work; see Section III for definitions.

[2] Following the conventions in theoretical computer science, we call $f = \mathrm{polylog}(g)$ an exponential improvement over g.



(a) UST T.

(b) $T\{S\} \subseteq T$.

(c) Optimal Steiner tree for S.

Fig. 2: 2a is a UST (blue) in unit-weight G with root r (green triangle). 2b is induced subtree $T\{S\}$ (orange) of weight 11 for S (orange diamonds). 2c is weight 6 optimal Steiner tree (green).

Problem	Param.	General	Thm.	Doubling	Ref.	Pathwidth	Ref.
UST	ρ	$O(\log^7 n)$	(VI.2)	$\tilde{O}(d^7) \cdot \log n$	[BCF+23]	$O(\mathrm{pw}^8 \cdot \log n)$	[BCF+23]
Strong Sparse Hierarchy	α	$O(\log n)$	(VI.1)	$O(d)$	[BCF+23]	$O(\mathrm{pw})$	[BCF+23]
	τ	$O(\log n)$		$\tilde{O}(d)$		$O(\mathrm{pw}^2)$	
	γ	$O(\log^2 n)$		$\tilde{O}(d^3)$		$O(\mathrm{pw}^2)$	
Dangling Nets [Fil20]	α	$O(\log n)$		$O(d)$		$O(\mathrm{pw})$	
	τ	$O(\log n)$		$\tilde{O}(d)$		$O(\mathrm{pw}^2)$	
Cluster Agg.	β	$O(\log n)$	(V.1)	$\tilde{O}(d^2)$	[BCF+23]	$O(\mathrm{pw})$	[BCF+23]

TABLE I: A summary of our results. For details on the results for bounded doubling dimension graphs and bounded pathwidth graphs, we refer the reader to the full version of the paper [BCF+23]. Our solutions for cluster aggregation on doubling dimension d only work for the instances of cluster aggregation we must solve to compute hierarchies of strong sparse partitions. $\tilde{O}$ notation hides poly$(\log d)$ factors. The results for dangling nets are proven implicitly in [Fil20]. Our algorithms for general graphs and bounded doubling dimension graphs are randomized and succeed with high probability $(1 - n^{-\Omega(1)})$, while our result for pathwidth graphs and trees are deterministic.

We also give improved UST bounds for graphs with doubling dimension d and graphs with pathwidth pw: poly$(d) \cdot \log n$ and poly$(\mathrm{pw}) \cdot \log n$ respectively. Bounded doubling dimension graphs are a well-studied graph class that generalizes the "bounded growth" of low-dimensional Euclidean space to arbitrary graphs [FLL06, AGGM06, ACGP16, KRX08, FS16, FKT19]. Bounded pathwidth graphs are a fundamental graph class that plays a key role in the celebrated graph minor theorem [RS86]. As discussed above, it was previously known that $O(\log n)$-approximate USTs are possible if G is a complete graph whose edge lengths are induced by either a constant doubling dimension metric or the shortest path metric of a constant pathwidth graph. Our results strengthen this, showing $O(\log n)$-approximate USTs are possible for these two cases without the additional assumption that G is the complete graph.

B. *Strong Sparse Partitions via Cluster Aggregation and Dangling Nets*

As mentioned, we achieve our UST algorithm by way of new results in graph hierarchies.

We build on works over the past several decades on efficiently decomposing and extracting structure from graphs and metrics. Notable examples of this work are ball carvings, low-diameter decompositions (LDDs), network decompositions, padded decompositions and sparse neighborhood covers, all of which have numerous algorithmic applications, especially in parallel and distributed computing [AP90, LS93, KPR93, FG19, BGK+11, Fil19a, FS10, EHRG22, ABCP96, ABN08, CG21, Bar04, KK17, RG20, FL22]. Generally speaking, these constructions separate a graph into clusters of nearby vertices while respecting graph distances.

The decompositions of our focus are *strong sparse* *partitions*, first defined by [JLN+05] (in their weak diameter version) and studied in several later works [BDR+12, CJK+22, Fil20].

Definition I.2 (Strong Δ-Diameter (α, τ)-Sparse Partitions)**.** *Given edge-weighted graph $G = (V, E, w)$, a strong Δ-diameter (α, τ)-sparse partition is a partition $\mathcal{C}$ of V such that:*

- ***Low (Strong) Diameter:*** *$\forall C \in \mathcal{C}$, the induced graph $G[C]$ has diameter at most Δ;*
- ***Ball Preservation:*** *$\forall v \in V$, the ball $B_G(v, \frac{\Delta}{\alpha})$ intersects at most τ clusters from $\mathcal{C}$.*

Sparse partitions with *weak diameter* and poly-logarithmic parameters can be constructed directly from classic sparse covers [AP90, JLN+05, Fil20] or ball carving techniques [Bar96, CJK+22]. However, to date, the only known techniques for poly-logarithmic sparse partitions with *strong diameter* guarantees in general graphs are the $(O(\log n), O(\log n))$-sparse partitions of [Fil20], constructed using exponentially-shifted starting times.[3] These start times were first used by [MPX13] to compute low diameter decompositions and spanners.

The simple graph class of trees cannot do much better than the strong $(O(\log n), O(\log n))$-sparse partitions in general graphs: both α and τ have to be essentially $\Omega(\log n)$. As such, bounded pathwidth and doubling dimension graphs are of particular interest. In particular, graphs with bounded pathwidth are exactly the graph family that excludes a fixed tree as a minor, circumventing this barrier with constant parameter strong sparse partitions.Conversely, trees that do not have good sparse partitions have doubling dimension $\Omega(\log n)$.

Little is known about graph decompositions in hierarchical settings; in particular, if our goal is a series of decompositions of increasing diameter where each decomposition coarsens the previous. One notable such hierarchy introduced by [BDR+12] is a hierarchy of strong sparse partitions.[4]

Definition I.3 (γ-Hierarchy of Strong (α, τ)-Sparse Partitions)**.** *Given edge-weighted graph $G = (V, E, w)$, a γ-hierarchy of strong (α, τ)-sparse partitions consists of vertex partitions $\{\{v\} : v \in V\} = \mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_k = \{\{V\}\}$ such that:*

- ***Strong Partitions:*** *$\mathcal{C}_i$ is a strong γ^i-diameter (α, τ)-sparse partition for every i;*
- ***Coarsening:*** *$\mathcal{C}_{i+1}$ coarsens $\mathcal{C}_i$, i.e. for each $U \in \mathcal{C}_i$ there is a $U' \in \mathcal{C}_{i+1}$ such that $U \subseteq U'$.*

If we did not enforce the above coarsening property, we could trivially compute the above partitions with poly-logarithmic parameters by using the strong sparse partitions from [Fil20] independently for each level of the hierarchy. However, the coarsening property renders computing such hierarchies highly non-trivial as it prevents such independent construction. Indeed, while hierarchies with poly-logarithmic parameterizations have been stated as an open question (see, e.g. [Fil20]), the previous best bounds known for such hierarchies are $\gamma = \alpha = \tau = 2^{O(\sqrt{\log n})}$ [BDR+12]. Thus, there is an exponential gap between the bounds known for "one-level" and hierarchical strong sparse partitions.

Nonetheless, previous work has demonstrated that these hierarchies can serve as the foundation of remarkably powerful algorithmic result such as USTs.

Theorem I.4 ([BDR+12])**.** *Given edge-weighted graph $G = (V, E, w)$ and a γ-hierarchy of strong (α, τ)-sparse partitions, one can compute an $O(\alpha^2\tau^2\gamma \log n)$-approximate UST in polynomial time.*

[BDR+12] gave $2^{O(\sqrt{\log n})}$-approximate USTs by combining the above theorem with their hierarchies.

Our second major contribution is a reduction of such hierarchies to the previously-studied cluster aggregation problem [BDR+12] and what we call dangling nets. Informally, cluster aggregation takes a vertex partition and a collection of portal vertices and coarsens it to a partition with a portal in each coarsened part. The goal is to guarantee that the portal in each vertex's coarsened cluster is nearly as close in its cluster as its originally-closest portal. Crucially for our purposes, the distortion of a solution is measured *additively*. See Figure 3 for an illustration of cluster aggregation.

Definition I.5 (Cluster Aggregation)**.** *An instance of cluster aggregation consists of an edge-weighted graph $G = (V, E, w)$, a partition $\mathcal{C}$ of V into clusters of strong diameter Δ and a set of portals $P \subseteq V$. A β-distortion solution is an assignment $f : \mathcal{C} \to P$ such that for every $v \in V$*

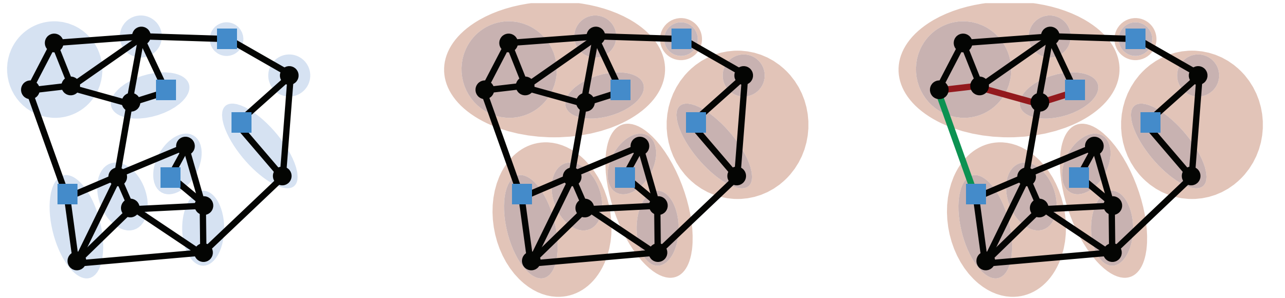
$$d_{G[f^{-1}(f(v))]}(v, f(v)) \leq d_G(v, P) + \beta \cdot \Delta$$

where $C_v \in \mathcal{C}$ is the cluster containing v and we let $f(v) := f(C_v)$ and $f^{-1}(p) := \{v : f(v) = p\}$.

In other words, a β-distortion cluster aggregation solution f requires that the distance from v to $p = f(v)$ in the cluster induced by p, is at most $\beta \cdot \Delta$ larger than

[3] Worse partitions with $2^{O(\sqrt{\log n})}$ parameters are possible by adapting the greedy approach of [BDR+12].

[4] We assume that the minimal pairwise distance in G is 1. Otherwise, we can scale all distances accordingly.



(a) Cluster aggregation instance. (b) Cluster aggregation solution. (c) Solution distortion.

Fig. 3: A cluster aggregation instance with unit-weight edges. 3a gives the instance; portals P are blue squares and the input partition parts $\mathcal{C}$ are blue ovals. 3b gives solution where each red oval is the pre-image of some portal. 3c illustrates why the solution is 2-distortion with the path of a vertex to its nearest portal in green and to its nearest portal in its coarsened cluster in red.

the distance from v to it's closest portal in G. Observe that any solution f on input cluster aggregation partition $\mathcal{C}$ naturally corresponds to a coarser partition $\mathcal{C}'$. Also, observe that, in general, we have that $\beta \geq 1$ by Figure 5.

Informally, a dangling net is a collection of net vertices we "dangle" off of a graph so that every vertex is close to a net vertex but no vertex has too many net vertices nearby. Crucially, the sense of "nearby" is also measured *additively*. See Figure 7b for an illustration.

Definition I.6 (Δ-Covering (α, τ)-Sparse Dangling Net)**.** *A dangling net for graph $G = (V, E, w)$ consists of vertices N where $N \cap V = \emptyset$ and a matching M with edge weights w_M from N to V. We let $G + N := (V \sqcup N, E \sqcup M, w \sqcup w_M)$ be the resulting graph. N is Δ-covering (α, τ)-sparse if*

- ***Covering:*** *$d_{G+N}(v, N) \leq \Delta$ for every $v \in V$;*
- ***Additive Sparsity:*** *for all $v \in V$ we have* $\left|\left\{t \in N : d_{G+N}(v,t) \leq d_{G+N}(v,N) + \frac{\Delta}{\alpha}\right\}\right| \leq \tau$.

While not explicitly stated in terms of dangling nets, the random shift analysis of [Fil20] implicitly prove the existence of good parameter dangling nets: e.g. $\alpha = \tau = O(\log n)$ for general graphs; see Theorem III.1.

We state our reduction of strong sparse hierarchies to cluster aggregation and dangling nets.

Theorem I.7. *Fix edge-weighted graph G and $\alpha, \beta, \tau \geq 0$. If for every $\Delta > 0$:*

- ***Dangling Net:*** *there is a dangling net N that is Δ-covering (α, τ)-sparse and;*
- ***Cluster Aggregation:*** *$G + N$ cluster aggregation on portals N is always β-distortion solvable;*

then, G has a $2\beta \cdot (2\alpha + 1)$-hierarchy of strong $(8\alpha + 4, \tau)$-sparse partitions. Furthermore, if each N and cluster aggregation solution is poly-time computable then the hierarchy is poly-time computable.

C. Improved Cluster Aggregation

The connection we establish between cluster aggregation, strong sparse partition hierarchies and USTs—as well as the fact that [BDR+12] posed improvements on their $O(\log^2 n)$-distortion cluster aggregation solutions as an open question—motivates further study of cluster aggregation.

Our third major contribution is an improvement to cluster aggregation distortion in a variety of graph classes. Notably, we improve the $O(\log^2 n)$-distortion solutions of [BDR+12] to $O(\log n)$-distortion for general graphs and give improved bounds for trees, bounded pathwidth and bounded doubling dimension graphs. For bounded doubling dimension graphs we must make assumptions on the input (see the full version [BCF+23]). We know of no bounds prior to our work for cluster aggregation other than the previous $O(\log^2 n)$ of [BDR+12] for general graphs. We summarize our cluster aggregation results in Figure 4 ($\kappa \leq n$ is the number of clusters in the input partition).

Combining our reduction (Theorem I.7) with the above cluster aggregation algorithms and dangling nets (Theorem III.1 for general graphs) gives our strong sparse partition hierarchies. Combining these hierarchies with Theorem I.4 gives our UST solutions. See Section VI for proof details and again, see Table I for an overview of the resulting bounds.

As the notation we use is quite standard, we defer a description of it and our (mostly) standard preliminaries to Section III.

Family	Distortion	Ref.
General	$O(\log^2 n)$	[BDR+12]
	$O(\log \kappa)$	Theorem V.1
Trees	4	Full ver. [BCF+23]
Doubling	$O(d^2 \cdot \log d)$	Full ver. [BCF+23]
Pathwidth	$8(\mathrm{pw}+1)$	Full ver. [BCF+23]

Fig. 4: Our cluster aggregation results.

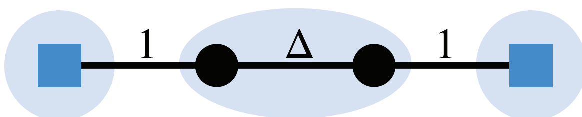


Fig. 5: Why $\beta \geq 1$ for cluster aggregation. One vertex in the center cluster must traverse its Δ-diameter cluster to get to a portal in any cluster aggregation solution.

D. Additional Related Work

We review additional related work not discussed earlier.

1) (Online and Oblivious) Steiner Tree: As it is an elementary NP-hard problem [GJ79], there has been extensive work on polynomial-time approximation algorithms for Steiner tree and related problems [AKR91, BGRS13, BGRS10, RZ05, HHZ21, Fil22, GKR00].

The subset of this work most closely related to our own is work on online and oblivious Steiner tree. In online Steiner tree the elements of $S \setminus \{r\}$ arrive one at a time and the algorithm must add a subset of edges to its solution so that it is feasible and cost-competitive with the optimal Steiner tree for the so-far arrived subset of $S \setminus \{r\}$. Notably, the greedy algorithm is a tight $O(\log n)$-approximation [IW91], though improved bounds are known if elements of $S \setminus \{r\}$ leave rather than arrive [GK14, GGK13]. See [AA92, NPS11, Ang07, XM22] for further work. Even harder, in oblivious Steiner tree, for each possible vertex $v \in V \setminus \{r\}$, the algorithm must pre-commit to a path P_v from r to v. Then, a subset S containing r is revealed and the algorithm must play as its solution the union of its pre-commited-to paths for S, namely $\bigcup_{v \in S \setminus \{r\}} P_v$. The goal of the algorithm is for its played solution to be cost-competitive with the optimal Steiner tree for S for every S. Notably, unlike USTs, the union of the paths played by the algorithm need not induce a tree. [GHR06] gave an $O(\log^2 n)$-approximate polynomial-time algorithm for this problem and its more general version "oblivious network design."

Observe that any ρ-approximate UST immediately gives a ρ-approximate oblivious Steiner tree algorithm which, in turn, gives a ρ-approximate online Steiner tree algorithm. Thus, in this sense UST is at least as hard as both online and oblivious Steiner tree.

2) Tree Embeddings and (Hierarchical) Graph Decompositions: There has been extensive work on approximating arbitrary graphs by distributions over trees by way of so-called probabilistic tree embeddings [Bar98, DGR06, AN12, BGS16, FRT03, ACE+20, FL21, HHZ21, Fil22]. Notably, any graph admits a distribution over trees that $O(\log n)$-approximate distances in expectation [FRT03] and a distribution over subtrees that $O(\log n \log \log n)$-approximate distances in expectation [AN12].

USTs and probabilistic tree embeddings both attempt to flatten the weight structure of a graph to a tree. However, tree embeddings only aim to provide pairwise guarantees in expectation, while USTs provide guarantees for every possible subset of vertices deterministically. While one can always sample many tree embeddings to provide pairwise guarantees with high probability, the corresponding subgraph will not be a single tree, unlike a UST.

As mentioned in Section I-B, decompositions of graphs into nearby vertices that respect distance structure have been extensively studied. See, for example, [CJK+22] for a recent application of sparse partitions in streaming algorithms. The graph decomposition most similar to sparse partitions are the scattering partitions of [Fil20]. Informally, scattering partitions provide the same guarantees as sparse partitions but with respect to shortest paths rather than balls.

These sorts of decompositions (and, in particular, hierarchies of them) are intimately related to tree embeddings. For example, the tree embeddings of [Bar98] can be viewed as a hierarchy of low-diameter decompositions. However, we note that, unlike strong sparse partition hierarchies, these hierarchies generally do not provide deterministic guarantees and, for example, [Bar98]'s hierarchy only provides weak diameter guarantees. Somewhat similarly, [ACE+20] produce a strong diameter padded decomposition hierarchy.

3) Universal Problems: In addition to Steiner tree, there are a number of problems whose universal versions have been studied. For example, the universal travelling salesman problem has been extensively studied [SS08, GKSS10, HKL06, BCK11, JLN+05, PBI89, BG89]. There are also works on universal set cover [JLN+05, GGL+08] and universal versions of clustering problems [GMP23].

II. Overview of Challenges and Intuition

Before moving on to our formal results, we give a brief overview of our techniques.

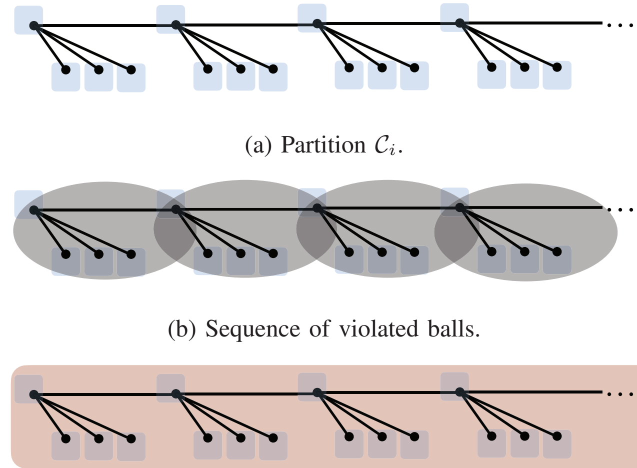
A. Reducing Hierarchies to Cluster Aggregation and Dangling Nets

Similarly to previous work, we take a bottom-up approach to compute strong sparse partition hierarchies. We begin with the singleton partition $\mathcal{C}_0 = \{\{v\} : v \in V\}$ and then compute each $\mathcal{C}_{i+1}$ using $\mathcal{C}_i$. Recall that our goal is a strong γ^{i+1}-diameter partition $\mathcal{C}_{i+1}$ which coarsens $\mathcal{C}_i$ and which guarantees that any ball of radius γ^{i+1}/α intersects at most τ clusters of $\mathcal{C}_{i+1}$.

Previous Approach: A natural strategy for computing the cluster $C'_j \in \mathcal{C}_{i+1}$ containing cluster $C_j \in \mathcal{C}_i$ is to start with C_j and expand it whenever it intersects a "violated" ball. Namely, if this cluster is incident to a diameter γ^{i+1}/α ball B intersecting more than τ clusters, grow this cluster to contain all clusters intersecting B. The issue with this is that we may end up with a very long sequence of violated balls, each of which forces us to grow C'_j further. See Figures 6a and 6b.

The main observation of [BDR+12] was that if the number of clusters each violated ball is incident to is at least $2^{O(\sqrt{\log n})}$, this sequence of violated balls can have length at most $2^{O(\sqrt{\log n})}$, which gives strong sparse partition hierarchies with $\alpha = \tau = \gamma = 2^{O(\sqrt{\log n})}$. Notably, the approach of [BDR+12] is "all or nothing" in that if there is a violated ball incident to more than τ clusters of $\mathcal{C}_i$, then *all* of these clusters are forced to be in the same cluster of $\mathcal{C}_{i+1}$. See Figure 6c.

Our Approach: Our approach uses dangling nets to coordinate cluster aggregation in a way that coarsens $\mathcal{C}_i$ without being all or nothing. On one hand, a dangling net respects balls but not in a way that has anything to do with $\mathcal{C}_i$ or coarsening it. In particular, a dangling net N corresponds to a natural sparse Voronoi partition (where each vertex goes to the closest net vertex in N) whose sparsity properties are robust to small (additive) changes. On the other hand, cluster aggregation provides a principled way of coarsening a partition $\mathcal{C}_i$ but does not necessarily respect balls. In particular, it coarsens a partition at the cost of small (additive) changes. We use dangling nets as portals for cluster aggregation to get the best of both techniques: cluster aggregation ensures that we coarsen with small additive costs while dangling nets ensure that these additive costs do not negatively impact sparsity. See Figure 7 for an illustration of our approach (and its later analysis).



(a) Partition $\mathcal{C}_i$.

(b) Sequence of violated balls.

(c) [BDR+12] all or nothing C'_j.

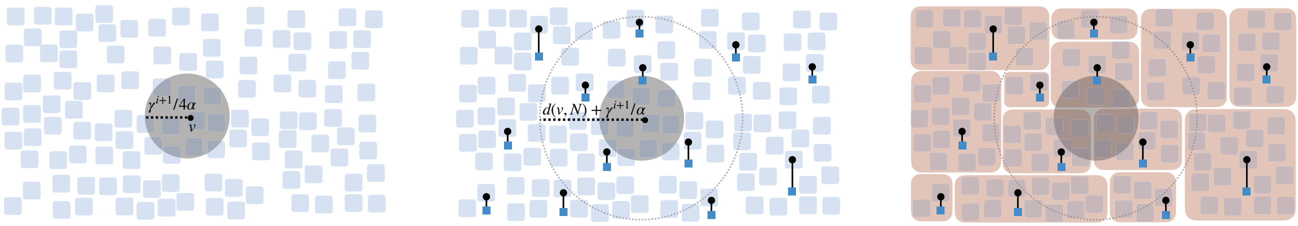
Fig. 6: The challenge of a sequence of violated balls when constructing the cluster in $\mathcal{C}_{i+1}$ containing $C_j \in \mathcal{C}_i$. 6a gives $\mathcal{C}_i$ as blue squares with C_j upper-left. 6b shows the "violated balls" of diameter γ^{i+1}/α. 6c shows the solution computed by [BDR+12] assuming that $\tau < 5$.

B. Improved Cluster Aggregation

We now briefly discuss our techniques for producing improved cluster aggregation solutions.

Our General Graphs Approach: Our approach for achieving an $O(\log n)$-distortion cluster aggregation is a round-robin process of $O(\log n)$ phases. In each phase, each unassigned cluster has a constant probability of merging with a cluster containing a portal. We accomplish this as follows. Define the *maximal internally disjoint* (MID) path of an unassigned cluster C as the maximal prefix of the shortest path from some representative node in C to a portal which is disjoint from all assigned clusters. In each phase we iterate through the clusters with portals. For each cluster C'_i with a portal we repeatedly flip a fair coin until we get a tails at which point we move on to the next cluster with a portal. Each time we get a heads we do an "expansion iteration", merging C'_i with all clusters incident to a MID path that ends at C'_i. Intuitively, this is a sort of geometric ball growing where MID paths are always treated as having weight 0. See Figure 8 for an illustration.

Every unassigned cluster is assigned in each phase with constant probability. Therefore with high probabil-



(a) $\mathcal{C}_i$ and one γ^{i+1}/α ball. (b) Dangling net N. (c) $\mathcal{C}_{i+1}$ via cluster aggregation.

Fig. 7: An illustration of our algorithm for coarsening $\mathcal{C}_i$ to $\mathcal{C}_{i+1}$. 7a gives $\mathcal{C}_i$ as transparent blue squares and one ball of radius $\gamma^{i+1}/4\alpha$ centered at v. 7b illustrates our dangling net N (opaque blue squares) and the fact that there are τ net vertices within distance $d(v, N) + \gamma^{i+1}/\alpha$ of v; in this case 5 net vertices. 7c gives the $\mathcal{C}_{i+1}$ resulting from cluster aggregation (in red) which guarantees that every vertex $u \in B_G(v, \frac{\gamma^{i+1}}{4\alpha})$ is sent to a portal at distance at most $d_G(v, N) + \beta \cdot \gamma^i + \frac{\gamma^{i+1}}{2\alpha} \leq d_G(v, N) + \frac{\gamma^{i+1}}{\alpha}$ from v, which are exactly the 5 net vertices.

ity after $O(\log n)$ iterations, every cluster is assigned to some portal. The additive distortion of this process can be bounded by the maximum number of heads any one cluster gets across all phases. The key to arguing $O(\log n)$ distortion is to observe that, while the distortion we incur may be as large as $\Theta(\log n)$ in one phase, the distortion any portal incurs *across all phases* is also $O(\log n)$. Thus, we bound across all phases at once. This can be contrasted with [BDR+12] who performed $O(\log n)$ phases of merging with $O(\log n)$ distortion per phase. See the full version of our work [BCF+23] for a description of how we exploit the structure of the family to argue that there are limited conflicts when merging clusters.

III. Notation, Conventions and Preliminaries

We review the (mostly standard) notation we use throughout this work.

General: We use $\sqcup$ for disjoint union; i.e. $U \sqcup V$ is the same set as $U \cup V$ but indicates that $U \cap V = \emptyset$. $\text{Geom}(p)$ is the geometric distribution where the probability for value i is $(1-p)^{i-1} \cdot p$, and the expectation is $\frac{1}{p}$. $\text{Bin}(n, p)$ stands for a binomial distribution with n samples, each with success probability p.

Graphs: Given edge-weighted graph $G = (V, E, w)$ and vertex subset $U \subseteq V$, we let $G[U] = (U, \{e : e \subseteq U\}, w)$ be the induced graph of U. Given two edge-weight functions w and w' on disjoint edge sets E and E', we let $w \sqcup w'$ be the edge-weight function that gives $w(e)$ to each $e \in E$ and $w'(e')$ to each $e' \in E'$. We let $d_G(u, v)$ be the weight of the shortest path between u and v according to w in G and for $S \subseteq V$ we let $d_G(v, S) = \min_{u \in S} d_G(v, u)$. The diameter of G is the maximum distance between a pair of vertices, i.e. $\max_{u,v \in U} d_G(u, v)$. The strong diameter of $S \subseteq V$ is the diameter of the induced graph $G[S]$, as opposed to the weak diameter $\max_{u,v \in S} d_G(u, v)$ (which is the maximum distance w.r.t. d_G). A partition $\mathcal{C}$ of V has strong (resp. weak) diameter Δ if $G[C_i]$ has strong (resp. weak) diameter for every $C_i \in \mathcal{C}$. The (closed) ball $B_G(v, r) := \{u : d_G(u, v) \leq r\}$ is all vertices within distance r from v in G. We drop the G subscript when the graph is clear from context. We let $n := |V|$ be the number of nodes in G throughout this paper. A metric space (X, d_X) induces a complete graph G with X as a vertex set, where the weight of the edge $\{u, v\}$ equals to the metric distance $d_X(u, v)$.

Dangling Net Constructions: We summarize results regarding dangling nets for general graphs; see the full version of our paper [BCF+23] for corresponding results for pathwidth- and doubling dimension-bounded graphs.

Theorem III.1 ([Fil20]). *Every weighted graph $G = (V, E, w)$ has a poly-time computable Δ-covering $(O(\log n), O(\log n))$-sparse dangling net for every $\Delta > 0$.*

Theorems III.1 is proven in [Fil20] in the context of "MPX partitions". There we sample shifts $\{\delta_t\}_{t \in N}$ and each vertex v joins the cluster of the center t maximizing $\delta_t - d_G(v, t)$. This is equivalent to our framework here, where in our dangling net we add t at distance $\Delta - \delta_t$ from its corresponding vertex in G. The statement corresponding to Theorem III.1 is Theorem 4 of III.1.

IV. Hierarchies via Cluster Aggregation and Dangling Nets

In this section we reduce the existence of strong sparse partition hierarchies to the existence of good dangling nets and cluster aggregation solutions. Our algorithm for doing so is Algorithm 1. It may be useful for the reader to recall the relevant definitions: strong sparse hierarchies (Definition I.3), cluster aggregation (Definition I.5) and dangling nets (Definition I.6).

Algorithm 1: `SSP Hierarchy`

input : Weighted graph $G = (V, E, w)$ (edge weights at least 1), algorithm for Δ-covering (α, τ)-sparse dangling net, algorithm for β-distortion cluster aggregation.

output: A $2\beta \cdot (2\alpha + 1)$-hierarchy of strong $(8\alpha + 4, \tau, \gamma)$-sparse partitions.

$i = 0$ and $\gamma = 2\beta(2\alpha + 1)$.
Set $\mathcal{C}_0 = \{\{v\} : v \in V\}$.
while $\mathcal{C}_i \neq \{\{V\}\}$ **do**
    Set $\Delta = 2\alpha\beta \cdot \gamma^i$.
    Compute a Δ-covering (α, τ)-sparse dangling net N.
    Compute a β-distortion cluster aggregation solution f on $G + N$ with portals N and clusters $\mathcal{C}_i \cup \{\{t\}\}_{t \in N}$ with corresponding coarsened partition $\mathcal{C}' := \{f^{-1}(t) : t \in N\}$.
    Let $\mathcal{C}_{i+1} = \{C \setminus N \mid C \in \mathcal{C}'\}$.
    $i \leftarrow i + 1$.
return $\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2 \ldots$

Formally, we show the following theorem whose proof is illustrated in Figure 7.

Theorem I.7. *Fix edge-weighted graph G and $\alpha, \beta, \tau \geq 0$. If for every $\Delta > 0$:*

- ***Dangling Net:*** *there is a dangling net N that is Δ-covering (α, τ)-sparse and;*
- ***Cluster Aggregation:*** *$G + N$ cluster aggregation on portals N is always β-distortion solvable;*

then, G has a $2\beta \cdot (2\alpha + 1)$-hierarchy of strong $(8\alpha + 4, \tau)$-sparse partitions. Furthermore, if each N and cluster aggregation solution is poly-time computable then the hierarchy is poly-time computable.

Proof. We begin by describing our algorithm for strong sparse partition hierarchies in words; see Algorithm 1 for pseudo-code. Our algorithm proceeds in rounds in a bottom up fashion, with round 0 being the trivial partition $\mathcal{C}_0$ to singletons, and round i constructing the coarsening of strong sparse partition $\mathcal{C}_i$ to obtain strong sparse partition $\mathcal{C}_{i+1}$.

In the remainder, we elaborate on the coarsening step of round i. Here, we receive as input a strong γ^i-diameter $(8\alpha + 4, \tau)$-sparse partition $\mathcal{C}_i$. Let $\Delta = 2\alpha\beta \cdot \gamma^i$. Using the assumption of our theorem, we create a Δ-covering (α, τ)-sparse dangling net N.

Next, we apply the cluster aggregation algorithm in the graph $G + N$ using N as the portals and $\mathcal{C}_i$ with a singleton cluster for each element of N as the input clusters. As a result we obtain assignment function f and corresponding coarsening $\mathcal{C}' := \{f^{-1}(t)\}_{t \in N}$. We obtain $\mathcal{C}_{i+1}$ by removing any vertex in N from any cluster in $\mathcal{C}'$.

We now establish that for every i, $\mathcal{C}_i$ forms a strong γ^i-diameter (α, τ)-sparse partition. The claim holds for $i = 0$ since $\mathcal{C}_0$ is a strong γ^0-diameter $(4(\alpha + 1), \gamma)$-sparse partition. Consider arbitrary $i > 0$. We assume by induction that $\mathcal{C}_i$ is a strong γ^i-diameter $(4(\alpha + 1), \tau)$-sparse partition. Recall that $\Delta = 2\alpha\beta \cdot \gamma^i$.

We begin by bounding the diameter of every cluster in $\mathcal{C}_i$. For any vertex $v \in V$, we know that $d_{G+N}(v, N) \leq \Delta$. Next we obtain a solution to the cluster aggregation problem f such that

$$\begin{aligned} d_{G+N[f^{-1}(v)]}(v, f(v)) &\leq d_{G+N}(v, N) + \beta \cdot \gamma^i \\ &\leq \Delta + \beta \cdot \gamma^i. \end{aligned}$$

It follows that $f^{-1}(v)$ has strong diameter at most

$$\begin{aligned} 2 \cdot \left(\Delta + \beta \cdot \gamma^i\right) &= 2 \cdot (2\alpha\beta + \beta) \cdot \gamma^i \\ &= 2\beta \cdot (2\alpha + 1) \cdot \gamma^i \\ &= \gamma^{i+1}. \end{aligned}$$

Finally, in the actual partition that we use, $\mathcal{C}_{i+1}$, we only remove vertices of degree 1 and this can only decrease the diameter. We conclude that $\mathcal{C}_{i+1}$ has strong diameter at most γ^{i+1} as required. It is also clear that $\mathcal{C}_{i+1}$ coarsens $\mathcal{C}_i$.

Next, we prove the ball preservation property. Fix a vertex $v \in V$. Consider a ball $B_G(v, R)$ around v of radius $R = \frac{\Delta}{4\alpha}$. For every $u \in B_G(v, R)$, the cluster aggregation solution assigns u to a portal $t_u \in N$. By the guarantees of cluster aggregation we have

$$\begin{aligned} d_{G+N}(v, t_u) &\leq d_{G+N}(v, u) + d_{G+N}(u, t_u) \\ &\leq d_G(v, u) + d_{G+N}(u, N) + \beta \cdot \gamma_i \\ &\leq d_{G+N}(v, N) + 2d_G(v, u) + \beta \cdot \gamma_i \\ &\leq d_{G+N}(v, N) + \frac{\Delta}{2\alpha} + \frac{\Delta}{2\alpha} \\ &= d_{G+N}(v, N) + \frac{\Delta}{\alpha} \ . \end{aligned}$$

As N is a Δ-covering (α, τ)-sparse dangling net, it holds that

$$\left|\left\{t \in N \mid d_{G+N}(v, t) \leq d_{G+N}(v, N) + \frac{\Delta}{\alpha}\right\}\right| \leq \tau.$$

It follows that the vertices in $B_G(v, R)$ are assigned to at most τ different portals, as required.

Finally, to conclude that $\mathcal{C}_{i+1}$ is $(8\alpha + 4, \tau)$-sparse, we observe that

$$\begin{aligned}\frac{\gamma^{i+1}}{R} &= \frac{\gamma^{i+1}}{\frac{\Delta}{4\alpha}} \\ &= \frac{4\alpha \cdot \gamma^{i+1}}{2\alpha\beta \cdot \gamma^{i}} \\ &= \frac{2\gamma}{\beta} \\ &= \frac{2 \cdot 2\beta \cdot (2\alpha + 1)}{\beta} \\ &= 8\alpha + 4 \ ,\end{aligned}$$

concluding our analysis and proof. □

V. Improved Cluster Aggregation

Having reduced strong sparse partition hierarchies to dangling nets and cluster aggregation in the previous section, we now give our new algorithms for $O(\log \kappa)$-distortion cluster aggregation solutions in general graphs when we are given $\kappa \leq n$ input clusters; see the full version of our work [BCF+23] for results for trees, doubling dimension-bounded and pathwidth-bounded graphs. The reader may want to review the definition of cluster aggregation (Definition I.5).

The following summarizes the main theorem of this section.

Theorem V.1. *Every instance of cluster aggregation with input partition $\mathcal{C} = \{C_1, \ldots, C_\kappa\}$ has an $O(\log \kappa)$-distortion solution that can be computed in polynomial time.*

To show the above we will bound the "detour" of a given cluster aggregation solution f; informally, how much extra distance a vertex travels in the solution.

Definition V.2 (Cluster Aggregation Detour)**.** *Given cluster aggregation solution f in graph G on portals P, we let the detour of vertex v be*

$$\mathrm{dtr}_f(v) := d_{G[f^{-1}(f(v))]}(v, f(v)) - d_G(v, P).$$

Observe that cluster aggregation solution f has distortion β if $\mathrm{dtr}_f(v) \leq \beta \cdot \Delta$ for every vertex v.

Our main approach is to grow the cluster of each portal in a round-robin and geometric fashion but treat each vertex's path to its nearest cluster with a portal as having length 0; this idea is generally in the spirit of the star decompositions of [DGR06] (see also the related Relaxed Voronoi algorithm [Fil19b]).

To formalize this, for each cluster $C_i \in \mathcal{C}$, arbitrarily choose a representative vertex $v_i \in C_i$, and let π_i denote a shortest path in G from v_i to its closest portal in P. At all times in the algorithm, we refer to the *maximal internally disjoint* (MID) prefix of π_i as π'_i. It is the maximal prefix of π_i such that its final node is the only node of the prefix belonging to a cluster already assigned to some portal. We denote the final node of the prefix by $\mathrm{final}(\pi'_i)$. Initially no clusters are assigned, and thus $\pi'_i = \pi_i$, and $\mathrm{final}(\pi'_i)$ is the closest portal to v_i.

Also observe that we may assume without loss of generality that no cluster contains more than one portal: any assignment that uses more than one portal contained in a given cluster C_i has infinite detour (that is, the portal not assigned cluster C_i is not reachable from any of its assigned clusters), and the use of one portal over another can increase the detour by at most Δ.

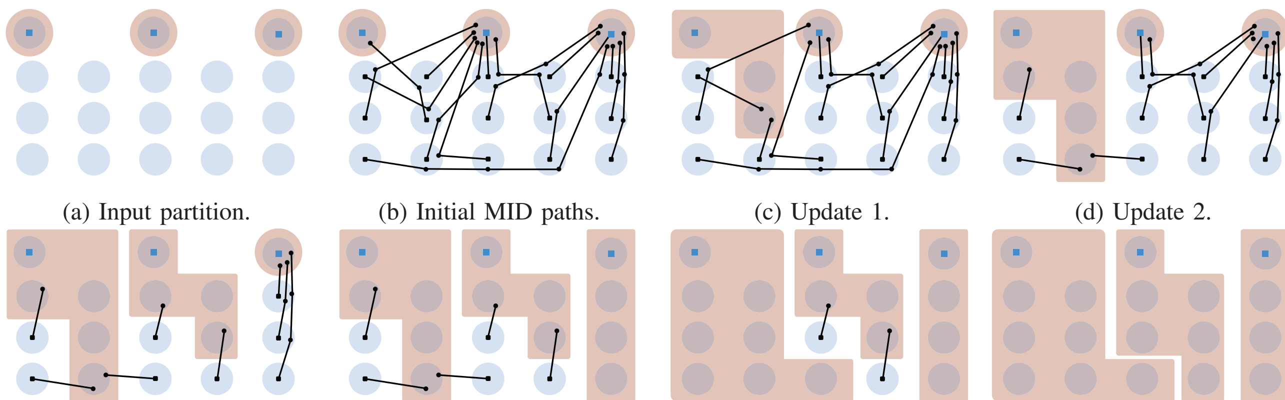
Algorithm Overview: Order the portals arbitrarily $p_1, \ldots, p_L$, and proceed in rounds. In every round j, each portal p_ℓ in sequence expands $f^{-1}(p_\ell)$, the set of clusters assigned to it, by claiming *all* clusters $C_i \in \mathcal{C}$ such that $f(\mathrm{final}(\pi'_i)) = p_\ell$, and all of the clusters along the paths π'_i. In other words, p_ℓ claims all the clusters C_k such that for some cluster C_i with $f(\mathrm{final}(\pi'_i)) = p_\ell$, C_k intersects π'_i.

Portal p_ℓ repeats this expansion a geometric variable $g_\ell^{(j)}$-many times, then we move on to the next portal. Clearly $f^{-1}(p_\ell)$ remains connected. We will show that only $O(\log(|\mathcal{C}|))$ rounds are needed to assign every cluster to a portal, and that the total detour of a node assigned to any portal p_ℓ is at most $2\Delta \cdot \sum_j g_\ell^{(j)}$, which will suffice to prove the theorem. The algorithm is presented formally in Algorithm 2 and illustrated in Figure 8.

Proof of Theorem V.1. We begin by showing that with high probability, after $10 \log |\mathcal{C}|$ rounds f is defined on the entire set C.

Lemma V.3. *The algorithm assigns every cluster, with high probability.*

Proof. We claim that in each round j, an unassigned cluster C_i has probability at least $\frac{1}{2}$ of being assigned to a portal. Consider the MID prefix π'_i of π_i at the start of round j, and let $p_{\ell^*} = f(\mathrm{final}(\pi'_i))$. If no vertex along π'_i was assigned to another cluster before iteration ℓ^*, then in iteration ℓ^* we will set $f(C_i) = p_{\ell^*}$, and be done with C_i. Otherwise, let ℓ' be the first iteration where some node u lying on π'_i is assigned to some other cluster. It may be that C_i was assigned, and we are done. Otherwise, let h be the expansion iteration for



(a) Input partition. (b) Initial MID paths. (c) Update 1. (d) Update 2.

(e) Update 3. (f) Update 4. (g) Update 5. (h) Update 6 (output).

Fig. 8: An illustration of our cluster aggregation algorithm. 8a gives the initial partition $\mathcal{C}$ in blue and initialized output $\mathcal{C}'$ in red. 8b gives the initial MID paths. We assume that the geometric random variables (left to right) is $2, 1, 1$ in the first round and $1, 1, 0$ in the second round. 8c, 8d, 8e, 8f, 8g and 8h give the updated $\mathcal{C}'$ and MID paths after each heads.

Algorithm 2: `General Graph CA`

input : Weighted graph $G = (V, E, w)$, portal set $P \subseteq V$, partition $\mathcal{C} = \{C_i\}_i$ into clusters of strong diameter at most Δ.

output: Assignment $f : \mathcal{C} \to P$ of additive distortion $\beta = O(\log|\mathcal{C}|)$.

Name the portals $P = \{p_1, \ldots, p_L\}$
For each p_ℓ in cluster C_i, set $f(C_i) = p_\ell$
for *rounds* $j = 1, 2, \ldots, 10\log|\mathcal{C}|$ **do**
  for *portals* $p_\ell = p_1, \ldots, p_L$ **do**
    Draw $g_\ell^{(j)} \sim \text{Geom}(\frac{1}{2})$
    for $h = 1, \ldots, g_\ell^{(j)}$ *(expansion iterations)* **do**
      Set $\mathcal{U}_1$ to all $C_i \in \mathcal{C}$ such that C_i is unassigned and $f(\text{final}(\pi_i')) = p_\ell$
      Set $\mathcal{U}_2$ to all $C_j \in \mathcal{C}$ such that $\exists C_i \in \mathcal{U}_1$ such that $C_j \cap \pi_i' \neq \emptyset$
      `// Note` $\mathcal{U}_1 \subseteq \mathcal{U}_2$
      For every cluster $C_i \in \mathcal{U}_2$ set $f(C_i) = p_\ell$
return f

$p_{\ell'}$ at which this first occurs. At this point the MID prefix π_i' is updated accordingly and $f(\text{final}(\pi_i')) = p_{\ell'}$. If $p_{\ell'}$ performs one additional expansion iteration then the cluster C_i will be assigned to $p_{\ell'}$. By the memorylessness of geometric distributions, this occurs with probability $\frac{1}{2}$. It follows that indeed C_i is clustered in round j with probability at least $\frac{1}{2}$.

While the rounds are not necessarily independent, it is clear from the above argument that this bound holds for any unassigned cluster in round j conditioned on any events that depend only on previous rounds. Therefore, denoting by $B_{C_i,j}$ the event that cluster C_i is *not* assigned in round j, we have that $\Pr[C_i \text{ not assigned}]$ is

$$\begin{aligned} &= \Pr\left[\bigcap_j B_{C_i,j}\right] \\ &= \prod_j \Pr\left[B_{C_i,j} \mid B_{C_i,1}, \ldots, B_{C_i,j-1}\right] \\ &\leq \left(\frac{1}{2}\right)^{10\log|\mathcal{C}|} \\ &= \frac{1}{|\mathcal{C}|^{10}}. \end{aligned}$$

Taking a union bound over $\mathcal{C}$ gives the desired result. □

The following summarizes the detour guarantees of our algorithm.

Lemma V.4. *The algorithm produces an assignment with detour* $\text{dtr}_f \leq O(\log|\mathcal{C}|) \cdot \Delta$*, with high probability.*

Proof. We claim that in any round j, a cluster C_i assigned to some p_ℓ in its h'th iteration of expansion satisfies $\forall v \in C_i$ that

$$d_{G[f^{-1}(p_\ell)]}(v, p_\ell) \le d_G(v, P) + 2\left(\sum_{j'=1}^{j-1} g_\ell^{(j')} + h\right) \cdot \Delta \tag{1}$$

We prove this by induction. When $j = 0$, only the portals are sent to themselves, and hence the distortion is 0. For any $j \ge 1$, consider some cluster C_i claimed by portal p_ℓ during round j after h expansion iterations. This means there was some cluster $C_{i'}$ such that it's MID prefix $\pi'_{i'}$ intersects C_i, and $f(\text{final}(\pi'_{i'})) = p_\ell$ (note that it is possible that $i = i'$). Let $w \in C_i \cap \pi'_{i'}$. Note that a shortest path from w to P follows π'_i. As $\text{final}(\pi'_{i'})$ is already assigned to p_ℓ at this time, by the induction hypothesis it holds that $d_{G[f^{-1}(p_\ell)]}(\text{final}(\pi'_{i'}), p_\ell) \le d_G(\text{final}(\pi'_{i'}), P) + 2\left(\sum_{j'=1}^{j-1} g_\ell^{(j')} + (h-1)\right) \cdot \Delta$. We conclude that for every vertex $v \in C_i$ it holds that (see Figure 9 for an illustration)

$$\begin{aligned}
&d_{G[f^{-1}(p_\ell)]}(v, p_\ell) \\
&\quad \le d_{G[f^{-1}(p_\ell)]}(v, w) + d_{G[f^{-1}(p_\ell)]}(w, \text{final}(\pi'_{i'})) \\
&\qquad\qquad + d_{G[f^{-1}(p_\ell)]}(\text{final}(\pi'_{i'}), p_\ell) \\
&\quad \le d_{G[f^{-1}(p_\ell)]}(v, w) + d_G(w, \text{final}(\pi'_{i'})) \\
&\qquad\qquad + d_G(\text{final}(\pi'_{i'}), P) \\
&\qquad\qquad + 2\left(\sum_{j'=1}^{j-1} g_\ell^{(j')} + (h-1)\right) \cdot \Delta \\
&\quad = d_{G[f^{-1}(p_\ell)]}(v, w) + d_G(w, P) \\
&\qquad\qquad + 2\left(\sum_{j'=1}^{j-1} g_\ell^{(j')} + (h-1)\right) \cdot \Delta \\
&\quad \le \quad d_G(v, P) \quad + 2 d_{G[f^{-1}(p_\ell)]}(v, w) \\
&\qquad\qquad + 2\left(\sum_{j'=1}^{j-1} g_\ell^{(j')} + (h-1)\right) \cdot \Delta \\
&\quad \le \quad d_G(v, P) \quad + 2\left(\sum_{j'=1}^{j-1} g_\ell^{(j')} + h\right) \cdot \Delta \ .
\end{aligned}$$

With the claim proved, we get that at the end of the algorithm every p_ℓ and $v \in f^{-1}(p_\ell)$, satisfies $d_{G[f^{-1}(p_\ell)]}(v, p_\ell) \le d_G(v, P) + 2 \cdot (\sum_{j'=1}^{10 \log|\mathcal{C}|} g_\ell^{(j')}) \cdot \Delta$. Let $X \sim \text{Bin}(40 \log|\mathcal{C}|, \frac{1}{2})$, and observe that the probability of needing more than $40 \log|\mathcal{C}|$ coin tosses to get $10 \log|\mathcal{C}|$ tails is equal to the probability that *exactly* $40 \log|\mathcal{C}|$ coin tosses results in *fewer than* $10 \log|\mathcal{C}|$ tails.

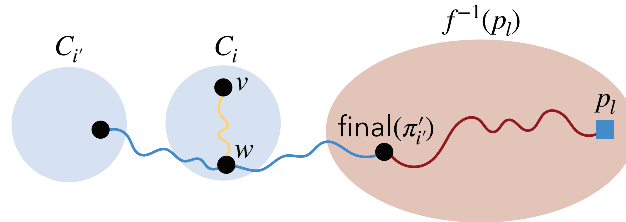


Fig. 9: Paths involved in bounding $d_{G[f^{-1}(p_\ell)]}(v, p_\ell)$. Blue path $\pi'_{i'}$ is the MID prefix that caused C_i to be assigned to p_ℓ; yellow path is a shortest path in C_i, so has length at most Δ; red path is a shortest path within coarsened cluster $f^{-1}(p_\ell)$, so its length is bounded by the induction hypothesis.

That is, we have the sum of IID geometric random variables $\Pr\left[\sum_{j'=1}^{10 \log|\mathcal{C}|} g_\ell^{(j')} > 40 \log|\mathcal{C}|\right]$ is

$$\begin{aligned}
&= \Pr\left[X < 10 \log|\mathcal{C}|\right] \\
&= \Pr\left[X < \frac{1}{2}\mathbb{E}[X]\right] \\
&< e^{-\mathbb{E}[X]/8} \le \frac{1}{|\mathcal{C}|^2}
\end{aligned}$$

by a standard Chernoff bound. Then, a union bound over $\mathcal{C}$ shows that the algorithm produces an assignment with detour $\le 80 \log|\mathcal{C}| \cdot \Delta$. Finally, letting B_U denote the event that there is an unassigned cluster, and B_R the event that some assigned node has detour $> 80 \log|\mathcal{C}| \cdot \Delta$, we have by lemma V.3 and a union bound:

$$\begin{aligned}
\Pr[\text{dtr}_f > 80 \log|\mathcal{C}| \cdot \Delta] &\le \Pr[B_U] + \Pr[B_R] \\
&\le \frac{1}{|\mathcal{C}|^9} + \frac{1}{|\mathcal{C}|} \\
&\le \frac{2}{|\mathcal{C}|} \ . \qquad \square
\end{aligned}$$

As $\kappa = |\mathcal{C}|$, the theorem follows. $\square$

VI. Combining Reduction, Cluster Aggregation and Dangling Nets

In this section, we combine our reduction of strong sparse partitions to dangling nets and cluster aggregation (Theorem I.7) with known dangling net constructions and our cluster aggregation solutions from Section V. The result is our strong sparse partition hierarchies (Definition I.3) which when fed into Theorem I.4 gives our UST constructions. We give our results for general graphs; see the full version of our work [BCF+23] for results on pathwidth- and doubling-dimension bounded graphs.

Theorem VI.1. *Every edge-weighted graph $G = (V, E, w)$ admits a γ-hierarchy of (α, τ)-sparse strong partitions for $\alpha = \tau = O(\log n)$ and $\gamma = O(\log^2 n)$.*

Proof. By Theorem III.1 every general graph has a poly-time computable Δ-covering $(O(\log n), O(\log n))$-sparse dangling net N for every $\Delta > 0$. Furthermore, for any such N we have that $G + N$ has a poly-time computable $O(\log n)$-distortion cluster aggregation by Theorem V.1. Applying our reduction theorem (Theorem I.7) gives the result. □

We note that our cluster aggregation for trees (see the full version [BCF+23]) allows us to improve γ to $O(\log n)$ in the above theorem (for the case where G is a tree).

Combining Theorem I.4 with Theorem VI.1 gives our UST theorem for general graphs.

Theorem VI.2. *Every edge-weighted graph $G = (V, E, w)$ admits an $O(\log^7 n)$-approximate universal Steiner tree. Furthermore, this tree can be computed in polynomial time.*

VII. Conclusion and Future Directions

In this work we gave the first poly-logarithmic universal Steiner trees in general graphs and strong sparse partition hierarchies. Our approach reduces poly-logarithmic strong sparse partition hierarchies to the cluster aggregation problem and dangling nets and then leverages a known connection between strong sparse partition hierarchies and universal Steiner trees. We gave $O(\log n)$-distortion solutions for cluster aggregation and improved solutions in trees and bounded pathwidth and doubling dimension graphs.

We conclude with some open questions and potential future directions:

1) **Improved UST and Strong Sparse Partition Bounds:** The most obvious remaining open direction is to close the gap between our $O(\log^7 n)$-approximate USTs and the known $\Omega(\log n)$ lower bound of [JLN+05]. Note that even assuming that G is the complete graph with metric weights, the best upper bound is $O(\log^2 n)$ [GHR06]. Along these lines, it would be interesting to improve the reduction of USTs to hierarchical strong sparse partitions or to improve the γ-parameter in the strong hierarchical sparse partitions for general graphs (the α and τ parameters are tight up to a $\log \log n$ factor [Fil20]).
2) **USTs and Strong Sparse Partition Hierarchies for New Graph Families:** Similarly, improving the bounds for new restricted graph families to get better USTs and strong sparse partitions is exciting. Currently, we know of no super-constant lower bound for UST for either constant treewidth or constant pathwidth graphs.
3) **Improved Cluster Aggregation in Restricted Graph Families:** One particularly interesting piece of this puzzle for restricted graph families is the status of cluster aggregation in special graph families. In particular, we conjecture that planar graphs and, more generally, minor-free graphs always admit $O(1)$-distortion cluster aggregation solutions. Likewise, we conjecture that $\tilde{O}(d)$-distortion cluster aggregation should be possible in graphs of doubling dimension d (our current upper bound is $\tilde{O}(d^2)$).
4) **Scattering Partitions:** A graph decomposition closely related to sparse partitions are the scattering partitions of [Fil20]. A $(1, \tau, \Delta)$-scattering partition is a partition into connected clusters with (weak) diameter at most Δ, such that every shortest path of length at most Δ intersects at most τ different clusters. Note that every strong sparse partition is also scattering, while weak sparse partitions and scattering partitions are incomparable. In a similar spirit to Theorem I.4, in [Fil20] it was shown that if every induced subgraph of G admits an $(1, \tau, \Delta)$-scattering partition for all Δ, then G admits an $O(\tau^3)$-stretch solution for the "Steiner point removal problem" (SPR). See [Fil20] for background and definitions. Recently Chang et al. [CCL+23] showed that planar graphs admit $(1, O(1))$-scattering partition schemes (implying an $O(1)$-stretch solution for SPR problem on such graphs). However, scattering partitions for general graphs are not yet understood. Filtser [Fil20] showed that n-vertex graphs admit $(1, O(\log^2 n))$-scattering partition schemes, and conjectured that they admit $(1, O(\log n))$-scattering partition schemes.

Acknowledgments

The authors would like to thank Bernhard Haeupler for helpful discussions leading to our doubling dimension result. The authors would also like to thank R. Ravi for many useful discussions related to Theorem V.1.

References

[AA92] Noga Alon and Yossi Azar. On-line steiner trees in the euclidean plane. In *Symposium on Computational Geometry (SoCG)*, pages 337–343, 1992. 6

[ABCP96] Baruch Awerbuch, Bonnie Berger, Lenore Cowen, and David Peleg. Fast distributed network decompositions and covers. *Journal of Parallel and Distributed Computing*, 39(2):105–114, 1996. 3

[ABN08] Ittai Abraham, Yair Bartal, and Ofer Neiman. Nearly tight low stretch spanning trees. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 781–790, 2008. 3

[ACE+20] Ittai Abraham, Shiri Chechik, Michael Elkin, Arnold Filtser, and Ofer Neiman. Ramsey spanning trees and their applications. *ACM Transactions on Algorithms (TALG)*, 16(2):1–21, 2020. 6

[ACGP16] Ittai Abraham, Shiri Chechik, Cyril Gavoille, and David Peleg. Forbidden-set distance labels for graphs of bounded doubling dimension. *ACM Transactions on Algorithms (TALG)*, 12(2):1–17, 2016. 3

[AGGM06] Ittai Abraham, Cyril Gavoille, Andrew V Goldberg, and Dahlia Malkhi. Routing in networks with low doubling dimension. In *26th IEEE International Conference on Distributed Computing Systems (ICDCS'06)*, pages 75–75. IEEE, 2006. 3

[AKR91] Ajit Agrawal, Philip Klein, and Ramamoorthi Ravi. When trees collide: An approximation algorithm for the generalized steiner problem on networks. In *Annual ACM Symposium on Theory of Computing (STOC)*, pages 134–144, 1991. 6

[AN12] Ittai Abraham and Ofer Neiman. Using petal-decompositions to build a low stretch spanning tree. In *Annual ACM Symposium on Theory of Computing (STOC)*, pages 395–406, 2012. 6

[Ang07] Spyros Angelopoulos. Improved bounds for the online steiner tree problem in graphs of bounded edge-asymmetry. In *Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, volume 7, pages 248–257. Citeseer, 2007. 6

[AP90] Baruch Awerbuch and David Peleg. Sparse partitions (extended abstract). In *Symposium on Foundations of Computer Science (FOCS)*, pages 503–513, 1990. 3, 4

[Bar96] Yair Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *Symposium on Foundations of Computer Science (FOCS)*, pages 184–193, 1996. 4

[Bar98] Yair Bartal. On approximating arbitrary metrices by tree metrics. In *Annual ACM Symposium on Theory of Computing (STOC)*, pages 161–168, 1998. 6

[Bar04] Yair Bartal. Graph decomposition lemmas and their role in metric embedding methods. In *European Symposium on Algorithms (ESA)*, pages 89–97, 2004. 3

[BCF+23] Costas Busch, Da Qi Chen, Arnold Filtser, Daniel Hathcock, D Ellis Hershkowitz, and Rajmohan Rajaraman. One tree to rule them all: Poly-logarithmic universal steiner tree. *arXiv preprint arXiv:2308.01199*, 2023. 2, 3, 5, 6, 8, 10, 12, 13

[BCK11] Anand Bhalgat, Deeparnab Chakrabarty, and Sanjeev Khanna. Optimal lower bounds for universal and differentially private steiner trees and tsps. In *APPROX-RANDOM*, pages 75–86. Springer, 2011. 6

[BDR+12] Costas Busch, Chinmoy Dutta, Jaikumar Radhakrishnan, Rajmohan Rajaraman, and Srivathsan Srinivasagopalan. Split and join: Strong partitions and universal steiner trees for graphs. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 81–90, 2012. 2, 4, 5, 6, 7, 8

[BG89] Dimitris Bertsimas and Michelangelo Grigni. Worst-case examples for the spacefilling curve heuristic for the euclidean traveling salesman problem. *Operations Research Letters*, 8(5):241–244, 1989. 6

[BGK+11] Guy E Blelloch, Anupam Gupta, Ioannis Koutis, Gary L Miller, Richard Peng, and Kanat Tangwongsan. Near linear-work parallel SDD solvers, low-diameter decomposition, and low-stretch subgraphs. In *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, pages 13–22, 2011. 3

[BGRS10] Jaroslaw Byrka, Fabrizio Grandoni, Thomas Rothvoß, and Laura Sanità. An improved lp-based approximation for steiner tree. In *Annual ACM Symposium on Theory of Computing (STOC)*, pages 583–592, 2010. 6

[BGRS13] Jaroslaw Byrka, Fabrizio Grandoni, Thomas Rothvoß, and Laura Sanità. Steiner tree approximation via iterative randomized rounding. *Journal of the ACM (JACM)*, 60(1):1–33, 2013. 6

[BGS16] Guy E Blelloch, Yan Gu, and Yihan Sun. Efficient construction of probabilistic tree embeddings. *arXiv preprint arXiv:1605.04651*, 2016. 6

[BLT14] Costas Busch, Ryan LaFortune, and Srikanta Tirthapura. Sparse covers for planar graphs and graphs that exclude a fixed minor. *Algorithmica*, 69(3):658–684, 2014. 2

[CCL+23] Hsien-Chih Chang, Jonathan Conroy, Hung Le, Lazar Milenkovic, Shay Solomon, and Cuong Than. Resolving the steiner point removal problem in planar graphs via shortcut partitions. *CoRR*, abs/2306.06235, 2023. 13

[CG21] Yi-Jun Chang and Mohsen Ghaffari. Strong-diameter network decomposition. In *ACM Symposium on Principles of Distributed Computing (PODC)*, pages 273–281, 2021. 3

[CJK+22] Artur Czumaj, Shaofeng H-C Jiang, Robert Krauthgamer, Pavel Veselỳ, and Mingwei Yang. Streaming facility location in high dimension via geometric hashing. In *Symposium on Foundations of Computer Science (FOCS)*, pages 450–461. IEEE, 2022. 4, 6

[DGR06] Kedar Dhamdhere, Anupam Gupta, and Harald Räcke. Improved embeddings of graph metrics into random trees. In *Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, volume 22, pages 61–69, 2006. 6, 10

[EHRG22] Michael Elkin, Bernhard Haeupler, Václav Rozhoň, and Christoph Grunau. Deterministic low-diameter decompositions for weighted graphs and distributed and parallel applications. *arXiv preprint arXiv:2204.08254*, 2022. 3

[FG19] Sebastian Forster and Gramoz Goranci. Dynamic low-stretch trees via dynamic low-diameter decompositions. In *ACM Symposium on Theory of Computing (STOC)*, pages 377–388, 2019. 3

[Fil19a] Arnold Filtser. On strong diameter padded decompositions. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2019, September 20-22, 2019, Massachusetts Institute of Technology, Cambridge, MA, USA.*, pages 6:1–6:21, 2019. 3

[Fil19b] Arnold Filtser. Steiner point removal with distortion $O(\log k)$ using the relaxed-voronoi algorithm. *SIAM J. Comput.*, 48(2):249–278, 2019. 10

[Fil20] Arnold Filtser. Scattering and sparse partitions, and their applications. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *International Colloquium on Automata, Languages and Programming (ICALP)*, volume 168, pages 47:1–47:20, 2020. 2, 3, 4, 5, 6, 8, 13

[Fil22] Arnold Filtser. Hop-constrained metric embeddings and their applications. In *Symposium on Foundations of Computer Science (FOCS)*, pages 492–503. IEEE, 2022. 6

[FKT19] Arnold Filtser, Robert Krauthgamer, and Ohad Trabelsi. Relaxed voronoi: A simple framework for terminal-clustering problems. In *Symposium on Simplicity in Algorithms (SOSA)*, page 1, 2019. 3

[FL21] Arnold Filtser and Hung Le. Clan embeddings into trees, and low treewidth graphs. In *Annual ACM Symposium on Theory of Computing (STOC)*, pages 342–355, 2021. 6

[FL22] Arnold Filtser and Hung Le. Locality-sensitive orderings and applications to reliable spanners. In Stefano Leonardi and Anupam Gupta, editors, *Annual ACM Symposium on Theory of Computing (STOC)*, pages 1066–1079. ACM, 2022. 3

[FLL06] Pierre Fraigniaud, Emmanuelle Lebhar, and Zvi Lotker. A doubling dimension threshold θ (loglog n) for augmented graph navigability. In *Annual European Symposium on Algorithms (ESA)*, pages 376–386. Springer, 2006. 3

[FRT03] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Annual ACM Symposium on Theory of Computing (STOC)*, pages 448–455, 2003. 6

[FS10] Jacob Fox and Benny Sudakov. Decompositions into subgraphs of small diameter. *Combinatorics, Probability and Computing*, 19(5-6):753–774, 2010. 3

[FS16] Arnold Filtser and Shay Solomon. The greedy spanner is existentially optimal. In

ACM Symposium on Principles of Distributed Computing (PODC), pages 9–17, 2016. 3

[GGK13] Albert Gu, Anupam Gupta, and Amit Kumar. The power of deferral: maintaining a constant-competitive steiner tree online. In *Annual ACM Symposium on Theory of Computing (STOC)*, pages 525–534, 2013. 6

[GGL+08] Fabrizio Grandoni, Anupam Gupta, Stefano Leonardi, Pauli Miettinen, Piotr Sankowski, and Mohit Singh. Set covering with our eyes closed. In *Symposium on Foundations of Computer Science (FOCS)*, pages 347–356. IEEE, 2008. 7

[GHR06] Anupam Gupta, Mohammad T Hajiaghayi, and Harald Räcke. Oblivious network design. In *Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 970–979, 2006. 2, 6, 13

[GJ79] Michael R Garey and David S Johnson. *Computers and intractability*, volume 174. freeman San Francisco, 1979. 6

[GK14] Anupam Gupta and Amit Kumar. Online steiner tree with deletions. In *Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 455–467. SIAM, 2014. 6

[GKR00] Naveen Garg, Goran Konjevod, and Ramamoorthi Ravi. A polylogarithmic approximation algorithm for the group steiner tree problem. *Journal of Algorithms*, 37(1):66–84, 2000. 6

[GKSS10] Igor Gorodezky, Robert D Kleinberg, David B Shmoys, and Gwen Spencer. Improved lower bounds for the universal and a priori tsp. In *APPROX-RANDOM*, pages 178–191. Springer, 2010. 6

[GMP23] Arun Ganesh, Bruce M Maggs, and Debmalya Panigrahi. Universal algorithms for clustering problems. *ACM Transactions on Algorithms*, 19(2):1–46, 2023. 7

[HHZ21] Bernhard Haeupler, D Ellis Hershkowitz, and Goran Zuzic. Tree embeddings for hop-constrained network design. In *Annual ACM Symposium on Theory of Computing (STOC)*, pages 356–369, 2021. 6

[HKL06] Mohammad Taghi Hajiaghayi, Robert D. Kleinberg, and Frank Thomson Leighton. Improved lower and upper bounds for universal TSP in planar metrics. In *Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 649–658, 2006. https://dl.acm.org/citation.cfm?id=1109557.1109628. 6

[HR92] Frank K Hwang and Dana S Richards. Steiner tree problems. *Networks*, 22(1):55–89, 1992. 1

[IW91] Makoto Imase and Bernard M Waxman. Dynamic steiner tree problem. *SIAM Journal on Discrete Mathematics*, 4(3):369–384, 1991. 6

[JLN+05] Lujun Jia, Guolong Lin, Guevara Noubir, Rajmohan Rajaraman, and Ravi Sundaram. Universal approximations for TSP, Steiner tree, and set cover. In *Annual ACM Symposium on Theory of Computing (STOC)*, pages 386–395, 2005. 2, 4, 6, 13

[KK17] Lior Kamma and Robert Krauthgamer. Metric decompositions of path-separable graphs. *Algorithmica*, 79(3):645–653, 2017. 3

[KPR93] Philip N. Klein, Serge A. Plotkin, and Satish Rao. Excluded minors, network decomposition, and multicommodity flow. In *Annual ACM Symposium on Theory of Computing (STOC)*, pages 682–690, 1993. 3

[KRX08] Goran Konjevod, Andréa W Richa, and Donglin Xia. Dynamic routing and location services in metrics of low doubling dimension. In *International Symposium on Distributed Computing (DISC)*, pages 417–417, 2008. 3

[LS93] Nathan Linial and Michael Saks. Low diameter graph decompositions. *Combinatorica*, 13(4):441–454, 1993. 3

[MPX13] Gary L. Miller, Richard Peng, and Shen Chen Xu. Parallel graph decomposition using random shifts. In *ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, pages 196–203, 2013. 2, 4

[NPS11] Joseph Naor, Debmalya Panigrahi, and Mohit Singh. Online node-weighted steiner tree and related problems. In *Symposium on Foundations of Computer Science (FOCS)*, pages 210–219. IEEE, 2011. 6

[PBI89] Loren K Platzman and John J Bartholdi III. Spacefilling curves and the planar travelling salesman problem. *Journal of the ACM (JACM)*, 36(4):719–737, 1989. 6

[RG20] Václav Rozhoň and Mohsen Ghaffari. Polylogarithmic-time deterministic network decomposition and distributed derandomization. In *ACM Symposium on Theory of Computing (STOC)*, pages 350–363, 2020. 3

[RS86] Neil Robertson and Paul D. Seymour. Graph minors. ii. algorithmic aspects of tree-width. *Journal of algorithms*, 7(3):309–322, 1986. 3

[RZ05] Gabriel Robins and Alexander Zelikovsky. Tighter bounds for graph steiner tree approximation. *SIAM Journal on Discrete Mathematics*, 19(1):122–134, 2005. 6

[SS08] Frans Schalekamp and David B Shmoys. Algorithms for the universal and a priori tsp. *Operations Research Letters*, 36(1):1–3, 2008. 6

[XM22] Chenyang Xu and Benjamin Moseley. Learning-augmented algorithms for online steiner tree. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 36, pages 8744–8752, 2022. 6