Scattering and Sparse Partitions, and Their Applications

Arnold Filtser
Columbia University, United States, New York, NY, USA
http://www.cs.columbia.edu/~arnoldf/
arnold273@gmail.com

Abstract

A partition $\mathcal{P}$ of a weighted graph G is (σ,τ,Δ)-sparse if every cluster has diameter at most Δ, and every ball of radius Δ/σ intersects at most τ clusters. Similarly, $\mathcal{P}$ is (σ,τ,Δ)-scattering if instead for balls we require that every shortest path of length at most Δ/σ intersects at most τ clusters. Given a graph G that admits a (σ,τ,Δ)-sparse partition for all $\Delta>0$, Jia et al. [STOC05] constructed a solution for the Universal Steiner Tree problem (and also Universal TSP) with stretch $O(\tau\sigma^2\log_\tau n)$. Given a graph G that admits a (σ,τ,Δ)-scattering partition for all $\Delta>0$, we construct a solution for the Steiner Point Removal problem with stretch $O(\tau^3\sigma^3)$. We then construct sparse and scattering partitions for various different graph families, receiving many new results for the Universal Steiner Tree and Steiner Point Removal problems.

2012 ACM Subject Classification Theory of computation

Keywords and phrases Scattering partitions, sparse partitions, sparse covers, Steiner point removal, Universal Steiner tree, Universal TSP

Digital Object Identifier 10.4230/LIPIcs.ICALP.2020.47

Category Track A: Algorithms, Complexity and Games

Related Version A full version of the paper is available at https://arxiv.org/abs/2001.04447. The reader is encouraged to read the full version of the paper [31].

Funding The research was supported by the Simons Foundation.

Acknowledgements The author would like to thank Alexandr Andoni, Robert Krauthgamer, Jason Li, Ofer Neiman, Anastasios Sidiropoulos, and Ohad Trabelsi for helpful discussions.

1 Introduction

Graph and metric clustering are widely used for various algorithmic applications (e.g., divide and conquer). Such partitions come in a variety of forms, satisfying different requirements. This paper is dedicated to the study of bounded diameter partitions, where small neighborhoods are guaranteed to intersects only a bounded number of clusters.

The first problem we study is the *Steiner Point Removal* (SPR) problem. Here we are given an undirected weighted graph $G=(V,E,w)$ and a subset of terminals $K\subseteq V$ of size k (the non-terminal vertices are called Steiner vertices). The goal is to construct a new weighted graph $M=(K,E',w')$, with the terminals as its vertex set, such that: (1) M is a graph minor of G, and (2) the distance between every pair of terminals t,t' in M is distorted by at most a multiplicative factor of α, formally, $\forall t,t'\in K,\quad d_G(t,t')\le d_M(t,t')\le\alpha\cdot d_G(t,t')$. Property (1) expresses preservation of the topological structure of the original graph. For example if G was planar, so will M be. Whereas property (2) expresses preservation of the geometric structure of the original graph, that is, distances between terminals. The question is thus: given a graph family $\mathcal{F}$, what is the minimal α such that every graph in $\mathcal{F}$ with a terminal set of size k will admit a solution to the SPR problem with distortion α.

47th International Colloquium on Automata, Languages, and Programming (ICALP 2020).
Editors: Artur Czumaj, Anuj Dawar, and Emanuela Merelli; Article No. 47; pp. 47:1–47:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Consider a weighted graph $G = (V, E, w)$ with a shortest path metric d_G. The *weak* diameter of a cluster $C \subseteq V$ is the maximal distance between a pair of vertices in the cluster w.r.t. d_G (i.e., $\max_{u,v\in C} d_G(u, v)$). The *strong* diameter is the maximal distance w.r.t. the shortest path metric in the induced graph $G[C]$ (i.e., $\max_{u,v\in C} d_{G[C]}(u, v)$). A partition $\mathcal{P}$ of G has weak (resp. strong) diameter Δ if every cluster $C \in \mathcal{P}$ has weak (resp. strong) diameter at most Δ. Partition $\mathcal{P}$ is *connected*, if the graph induced by every cluster $C \in \mathcal{P}$ is connected. Given a shortest path $\mathcal{I} = \{v_0, v_1, \ldots, v_s\}$, denote by $Z_{\mathcal{I}}(\mathcal{P}) = \sum_{C\in\mathcal{P}} \mathbb{1}_{C\cap\mathcal{I}\neq\emptyset}$ the number of clusters in $\mathcal{P}$ intersecting $\mathcal{I}$. If $Z_{\mathcal{I}}(\mathcal{P}) \leq \tau$, we say that $\mathcal{I}$ is τ-*scattered* by $\mathcal{P}$.

▶ **Definition 1** (Scattering Partition). *Given a weighted graph $G = (V, E, w)$, we say that a partition $\mathcal{P}$ is (σ, τ, Δ)-scattering if the following conditions hold:*

- *$\mathcal{P}$ is connected and has weak diameter Δ.*
- *Every shortest path $\mathcal{I}$ of length at most Δ/σ is τ-scattered by $\mathcal{P}$, i.e., $Z_{\mathcal{I}}(\mathcal{P}) \leq \tau$.*

*We say that a graph G is (σ, τ)-*scatterable *if for every parameter Δ, G admits an (σ, τ, Δ)-scattering partition that can be computed efficiently.*

The main contribution of this paper is the finding that scattering partitions imply solutions for the SPR problem. The proof appears in Section 3.[1]

▶ **Theorem 2** (Scattering Partitions imply SPR). *Let $G = (V, E, w)$ be a weighted graph such that for every subset $A \subseteq V$, $G[A]$ is $(1, \tau)$-scatterable. Let $K \subseteq V$ be some subset of terminals. Then there is a solution to the* SPR *problem with distortion $O(\tau^3)$ that can be computed efficiently.*

Jia, Lin, Noubir, Rajaraman, and Sundaram [45] [2] defined the notion of *sparse partitions*, which is closely related to scattering partitions. Let $\mathcal{P}$ be a partition. Given a ball $B = B_G(x, r)$, denote by $Z_B(\mathcal{P}) = \sum_{C\in\mathcal{P}} \mathbb{1}_{C\cap B\neq\emptyset}$ the number of clusters in $\mathcal{P}$ intersecting B.

▶ **Definition 3** (Strong/Weak Sparse Partition). *Given a weighted graph $G = (V, E, w)$, we say that a partition $\mathcal{P}$ is (σ, τ, Δ)-weak (resp. strong) sparse partition if the following conditions hold:*

- *$\mathcal{P}$ has weak (resp. strong) diameterΔ.*
- *Every ball $B = B_G(v, r)$ of radius $r \leq \Delta/\sigma$ intersects at most τ clusters, i.e., $Z_B(\mathcal{P}) \leq \tau$.*

*We say that a graph G admits an (σ, τ)-*weak *(resp. strong) sparse partition scheme if for every parameter Δ, G admits an efficiently computable (σ, τ, Δ)-weak (resp. strong) sparse partition.*

Jia et al. [45] found a connection between sparse partitions to the *Universal Steiner Tree Problem* (UST).[3] Consider a complete weighted graph $G = (V, E, w)$ (or a metric space (X, d)) where there is a special server vertex $\mathrm{rt} \in V$, which is frequently required to multicast messages to different subsets of clients $S \subseteq V$. The cost of a multicast is the total weight of all edges used for the communication. Given a subset S, the optimal solution is to use the minimal Steiner tree spanning $S \cup \{\mathrm{rt}\}$. In order to implement an infrastructure for multicasting, or in order to make routing decisions much faster (and not compute it from scratch once S is given), a better solution will be to compute a *Universal Steiner Tree* (UST). A UST is a tree T over V, such that for every subset S, the message will be sent using the

[1] In Observation 2 we argue that (σ, τ, Δ)-scattering partition is also $(1, \tau\sigma, \Delta)$-scattering.

[2] Awerbuch and Peleg [8] were the first to study sparse covers (see Definition 5). Their notion of sparse partition is somewhat different from the one used here (introduced by [45]).

[3] A closely related problem is the *Universal Traveling Salesman Problem* (UTSP), see Section 1.4.

sub-tree $T(S)$ spanning $S \cup \{\mathrm{rt}\}$. The stretch of T is the maximum ratio among all subsets $S \subseteq X$ between the weight of $T(S)$ and the weight of the minimal Steiner tree spanning $S \cup \{\mathrm{rt}\}$, $\max_{S\subseteq X} \frac{w(T(S))}{\mathrm{Opt}(S\cup\{\mathrm{rt}\})}$.

Jia et al. [45] proved that given a sparse partition scheme, one can efficiently construct a UST with low stretch (the same statement holds w.r.t. UTSP as well).

▶ **Theorem 4** (Sparse Partitions imply UST, [45])**.** *Suppose that an n-vertex graph G admits an (σ,τ)-weak sparse partition scheme, then there is a polynomial time algorithm that given a root $\mathrm{rt} \in V$ computes a UST with stretch $O(\tau\sigma^2 \log_\tau n)$.*

Jia et al. [45] constructed $(O(\log n), O(\log n))$-weak sparse partition scheme for general graphs, receiving a solution with stretch $\mathrm{polylog}(n)$ for the UST problem. In some instances the communication is allowed to flow only in certain routes. It is therefore natural to consider the case where $G = (V, E, w)$ is not a complete graph, and the UST is required to be a subgraph of G. Busch, Jaikumar, Radhakrishnan, Rajaraman, and Srivathsan [12] proved a theorem in the spirit of Theorem 4, stating that given a (σ,τ,γ)-*hierarchical strong sparse partition*, one can efficiently construct a subgraph UST with stretch $O(\sigma^2\tau^2\gamma \log n)$. A (σ,τ,γ)-hierarchical strong sparse partition is a laminar collection of partitions $\{\mathcal{P}_i\}_{i\geq 0}$ such that $\mathcal{P}_i$ is (σ,τ,γ^i)-strong sparse partition which is a refinement of $\mathcal{P}_{i+1}$.[4] Busch et al. constructed a $\left(2^{O(\sqrt{\log n})}, 2^{O(\sqrt{\log n})}, 2^{O(\sqrt{\log n})}\right)$-hierarchical strong sparse partition, obtaining a $2^{O(\sqrt{\log n})}$ stretch algorithm for the subgraph UST problem. We tend to believe that poly-logarithmic stretch should be possible. It is therefore interesting to construct strong sparse partitions, as it eventually may lead to hierarchical ones.

A notion which is closely related to sparse partitions is *sparse covers*.

▶ **Definition 5** (Strong/Weak Sparse cover)**.** *Given a weighted graph $G = (V, E, w)$, a (σ,τ,Δ)-weak (resp. strong) sparse cover is a set of clusters $\mathcal{C} \subset 2^V$, where all the clusters have weak (resp. strong) diameter at most Δ, and the following conditions hold:*

- *Cover: $\forall u \in V$, exists $C \in \mathcal{C}$ such that $B_G(u, \frac{\Delta}{\sigma}) \subseteq C$.*
- *Sparsity: every vertex $u \in V$ belongs to at most $|\{C \in \mathcal{C} \mid u \in C\}| \leq \tau$ clusters.*

We say that a graph G admits an (σ,τ)-weak *(resp. strong) sparse cover scheme if for every parameter Δ, G admits an (σ,τ,Δ)-weak (resp. strong) sparse cover that can be computed efficiently.*

It was (implicitly) proven in [45] that given (σ,τ,Δ)-weak sparse cover $\mathcal{C}$, one can construct an (σ,τ,Δ)-weak sparse partition. In fact, most previous constructions of weak sparse partitions were based on sparse covers.

1.1 Previous results

SPR. Given an n-point tree, Gupta [38] provided an upper bound of 8 for the SPR problem (on trees). This result were recently reproved by the author, Krauthgamer, and Trabelsi [33] using the `Relaxed-Voronoi` framework. Chan, Xia, Konjevod, and Richa [14] provided a lower bound of 8 for trees. This is the best known lower bound for the general SPR problem. Basu and Gupta [11] provided an $O(1)$ upper bound for the family of outerplanar graphs.[5] For general n-vertex graphs with k terminals the author [27, 29] recently proved an $O(\log k)$ upper bound for the SPR problem using the `Relaxed-Voronoi` framework, improving upon

[4] We assume here w.l.o.g. that the minimal distance in G is 1.
[5] Actually the manuscript [11] was never published, and thus did not go through a peer review process.

previous works by Kamma, Krauthgamer, and Nguyen [46] ($O(\log^5 k)$), and Cheung [16] ($O(\log^2 k)$) (which were based on the `Ball-Growing` algorithm). Interestingly, there are no results on any other restricted graph family, although several attempts have been made (see [25, 50, 17]).

UST. Given an n-point metric space and root rt, Gupta, Hajiaghayi and Räcke [39] constructed a UST with stretch $O(\log^2 n)$, improving upon a previous $O(\log^4 n/\log\log n)$ result by [45]. [45] is based on sparse partitions, while [39] is based on *tree covers*. Jia et al. [45] proved a lower bound of $\Omega(\log n)$ to the UST problem, based on a lower bound to the online Steiner tree problem by Alon and Azar [6]. Using the same argument, they [45] proved an $\Omega(\frac{\log n}{\log\log n})$ lower bound for the case where the space is the $n\times n$ grid (using [43]). Given a space with doubling dimension ddim, [6] Jia et al. [45] provided a solution with stretch $2^{O(\mathrm{ddim})}\cdot\log n$, using sparse partitions. Given an n vertex planar graph, Busch, LaFortune, and Tirthapura [13] proved an $O(\log n)$ upper bound (improving over Hajiaghayi, Kleinberg, and Leighton [42]). More generally, for graphs G excluding a fixed minor, both Hajiaghayi et al. [42] (implicitly) and Busch et al. [13] (explicitly) provided a solution with stretch $O(\log^2 n)$. Both constructions used sparse covers. Finally, Busch et al. [12] constructed a subgraph UST with stretch polylog(n) for graphs excluding a fixed minor (using hierarchical strong sparse partitions).

Scattering Partitions. As we are the first to define scattering partitions there is not much previous work. Nonetheless, Kamma et al. [46] implicitly proved that general n-vertex graphs are $(O(\log n), O(\log n))$-scatterable.[7]

Sparse Covers and Partitions. Awerbuch and Peleg [8] introduced the notion of sparse covers and constructed $(O(\log n), O(\log n))$-strong sparse cover scheme for n-vertex weighted graphs.[8] Jia et al. [45] induced an $(O(\log n), O(\log n))$-weak sparse partition scheme. Hajiaghayi et al. [42] constructed an $(O(1), O(\log n))$-weak sparse cover scheme for n-vertex planar graph, concluding an $(O(1), O(\log n))$-weak sparse partition scheme. Their construction is based on the [48] clustering algorithm. Abraham, Gavoille, Malkhi, and Wieder [5] constructed $(O(r^2), 2^{O(r)}\cdot r!)$-strong sparse cover scheme for K_r-free graphs. Busch et al. [13] constructed a $(48, 18)$-strong sparse cover scheme for planar graphs [9] and $(8, O(\log n))$-strong sparse cover scheme for graphs excluding a fixed minor, concluding a $(48, 18)$ and $(8, O(\log n))$-weak sparse partition schemes for these families (respectively). For graphs with doubling dimension ddim, Jia et al. [45] constructed an $(1, 8^{\mathrm{ddim}})$-weak sparse scheme. Abraham et al. [3] constructed a $(2, 4^{\mathrm{ddim}})$-strong sparse cover scheme. In a companion paper, the author [28] constructed an $(O(\mathrm{ddim}), O(\mathrm{ddim}\cdot\log\mathrm{ddim}))$-strong sparse cover scheme.[10] Busch et al. [12] constructed $\big(O(\log^4 n), O(\log^3 n), O(\log^4 n)\big)$-hierarchical strong sparse partition for graphs excluding a fixed minor.

[6] A metric space (X, d) has doubling dimension ddim if every ball of radius $2r$ can be covered by 2^{ddim} balls of radius r. The doubling dimension of a graph is the doubling dimension of its induced shortest path metric.

[7] This follows from Theorem 1.6 in [46] by choosing parameters $t=\beta=O(\log n)$ and using union bounds over all n^2 shortest paths. Note that they assume that for every pair of vertices there is a unique shortest path.

[8] More generally, for $k\in\mathbb{N}$, [8] constructed a $(2k-1, 2k\cdot n^{\frac{1}{k}})$-strong sparse cover scheme.

[9] Busch et al. argued that they constructed $(24, 18)$-strong sparse covering scheme. However they measured radius rather than diameter.

[10] More generally, for a parameter $t=\Omega(1)$, [28] constructed $\big(O(t), O(2^{\mathrm{ddim}/t}\cdot\mathrm{ddim}\cdot\log t)\big)$-sparse cover scheme.

1.2 Our Contribution

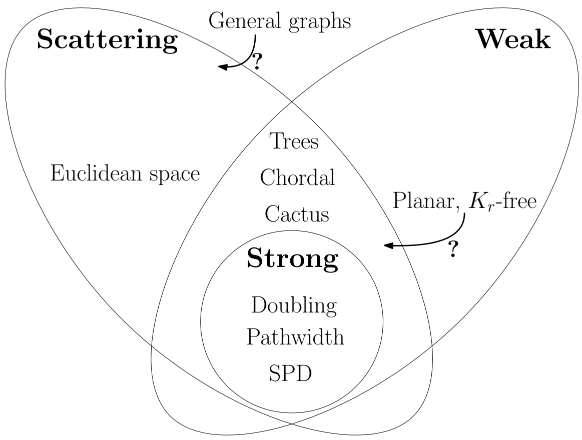

Figure 1 Classification of various graph families according to the possibility of construction different partitions. Graphs with bounded doubling dimension or SPD [14] (pathwidth) admit strong sparse partitions with parameters depending only on the dimension/SPDdepth. Trees, Chordal and Cactus graphs admit both $(O(1),O(1))$-weak sparse and scattering partitions, while similar strong partitions are impossible. $\mathbb{R}^d$ with norm 2 admit $(1,2d)$ scattering partition while weak sparse partition with constant padding will have an exponential number of intersections. Planar graphs admit $(O(1),O(1))$-weak sparse partitions, while it is an open question whether similar scattering partitions exist. Finally, while sparse partitions for general graphs are well understood, we lack a lower bound for scattering partitions.

Formal statements, and proofs of all our partitions are differed to the full version [31]. The main contribution of this paper is the definition of scattering partition and the finding that good scattering partitions imply low distortion solutions for the SPR problem (Theorem 2). We construct various scattering and sparse partition schemes for many different graph families, and systematically classify them according to the partition types they admit. In addition, we provide several lower bounds. The specific partitions and lower bounds are described below. Our findings are summarized in Table 1, while the resulting classification is illustrated in Figure 1.

Recall that [45] (implicitly) showed that sparse covers imply weak sparse partitions. We show that the opposite direction is also true. That is, given a (σ,τ,Δ)-weak sparse partition, one can construct an $(\sigma+2,\tau,(1+\frac{2}{\sigma})\Delta)$-weak sparse cover. Interestingly, in addition we show that strong sparse partitions imply strong sparse covers, while the opposite is not true. Specifically there are graph families that admit $(O(1),O(1))$-strong sparse cover schemes, while there are no constants σ,τ, such that they admit (σ,τ)-strong sparse partitions. Description of our findings on the connection between sparse partitions and sparse covers, and a classification of various graph families are differed to the full version [31].

The scattering partitions we construct imply new solutions for the SPR problem previously unknown. Specifically, for every graph with pathwidth ρ we provide a solution to the SPR problem with distortion poly(ρ), independent of the number of terminals. After trees [38] and outerplanar graphs [11] [5], this is the first graph family to have solution for the SPR problem independent from the number of terminals (although attempts were made). Furthermore, we obtain solution with constant distortion for Chordal and Cactus graphs.[11]

[11] Note that the family of cactus graph is contained in the family of outerplanar graph. Basu and Gupta [11] solved the SPR problem directly on outerplanar graphs with constant distortion. However, this manuscript was never published. See also 5.

Table 1 Summery of the various new/old, weak/strong scattering/sparse partitions.
Table footnotes: $^{\clubsuit}$ More generally, there is a partition $\mathcal{P}$ s.t. every ball of radius $\frac{\Delta}{8\alpha}$ intersects at most $\tilde{O}(n^{1/\alpha})$ clusters, for all $\alpha > 1$ simultaneously. $^{\blacksquare}$ More generally, it must hold that $\tau \geq n^{\Omega(1/\sigma)}$. $^{\spadesuit}$ More generally, there is a partition $\mathcal{P}$ s.t. every ball of radius $\Omega(\frac{\Delta}{\alpha})$ intersects at most $O(2^{\mathrm{ddim}/\alpha})$ clusters, for all $\alpha > 1$ simultaneously. $^{\blacklozenge}$ More generally, it must hold that $\tau > (1 + \frac{1}{2\sigma})^{d}$. $^{\bigstar}$ Note that this lower bound holds chordal/cactus/planar/K_r-free graphs. More generally, it must hold that $\tau \geq \Omega(n^{2/\sigma+1)})$.

Family	**Partition type**	**Padding (σ)**	**#inter. (τ)**	**Ref.**
General n-vertex Graphs	Weak	$O(\log n)$	$O(\log n)$	[45]
	Scattering	$O(\log n)$	$O(\log n)$	[46]
	Strong	$O(\log n)$	$O(\log n)$	This paper $\clubsuit$
	Weak L.B.	$\Omega(\log n/\log\log n)$	$O(\log n)$	This paper $\blacksquare$
ddim **doubling dimension**	Weak	1	8^{ddim}	[45]
	Strong	$O(\mathrm{ddim})$	$\tilde{O}(\mathrm{ddim})$	This paper $\spadesuit$
Euclidean space $(\mathbb{R}^d, \|\cdot\|_2)$	Scattering	1	$2d$	This paper
	Weak L.B.	$O(1)$ $\Omega(d/\log d)$	$2^{\Omega(d)}$ $\mathrm{poly}(d)$	This paper $\blacklozenge$
Trees	Scattering	2	3	This paper
	Weak	4	3	This paper
	Strong L.B.	$\log n/\log\log n$ $\sqrt{\log n}$	$\log n$ $2^{\sqrt{\log n}}$	This paper $\bigstar$
Pathwidth ρ (SPDdepth [14])	Strong	$O(\rho)$	$O(\rho^2)$	This paper
	Weak	8	5ρ	This paper
Chordal	Scattering	2	3	This paper
	Weak	24	3	This paper
K_r free	Weak	$O(r^2)$	2^r	This paper
Cactus	Scattering	4	5	This paper

The weak sparse partitions we construct imply improved solutions for the UST (and UTSP) problem. Specifically, we conclude that for graphs with doubling dimension ddim a UST (and UTSP) with stretch $\mathrm{poly}(\mathrm{ddim}) \cdot \log n$ can be efficiently computed, providing an exponential improvement in the dependence on ddim compared with the previous state of the art [45] of $2^{O(\mathrm{ddim})} \cdot \log n$. For K_r-minor free graphs we conclude that an UST (or UTSP) with stretch $2^{O(r)} \cdot \log n$ can be efficiently computed, providing a quadratic improvement in the dependence on n compared with the previous state of the art [39] of $O(\log^2 n)$. [12] Finally, for pathwidth ρ graphs (or more generally, graph with SPDdepth ρ) we can compute a UST (or UTSP) with stretch $O(\rho \cdot \log n)$, improving over previous solutions that were exponential in ρ (based on the fact that pathwidth ρ graphs are $K_{\rho+2}$-minor free).

Before we proceed to describe our partitions we make two observations.

[12] This result is a mere corollary obtained by assembling previously existing parts together. Mysteriously, although UTSP on minor free graphs was studied before [42, 13], this corollary was never drawn.

▶ **Observation 1.** *Every (σ,τ,Δ)-strong sparse partition is also scattering partition and weak sparse partition with the same parameters.*

▶ **Observation 2.** *Every (σ,τ,Δ)-scattering partition is also $(1,\sigma\tau,\Delta)$-scattering partition.*

Observation 1 follows as every path of weight $\sigma\Delta$ is contained in a ball of radius $\sigma\Delta$. Observation 2 follows as every shortest path of length $\leq\Delta$ can be assembled as a concatenation of at most σ shortest paths of length $\leq\frac{\Delta}{\sigma}$.

Next we survey the partitions for various graph families. Formal statements and proofs are differed to the full version [31].

General Graphs. Given an n-vertex general graph and parameter $\Delta>0$ we construct a single partition $\mathcal{P}$ which is simultaneously $\left(8k,O(n^{1/k}\cdot\log n),\Delta\right)$-strong sparse partition for all parameters $k\geq 1$. Thus we generalize the result of [45] and obtained a strong diameter guarantee. This partition implies that general graphs are $(O(\log n),O(\log n))$-scatterable (reproving [46] via an easier proof), inducing a solution for the SPR problem with stretch $\mathrm{polylog}(|K|)$. While quantitatively better solutions are known, this one is arguably the simplest, and induced by a general framework. Further, we provide a lower bound, showing that if all n-vertex graphs admit (σ,τ)-weak sparse partition scheme, then $\tau\geq n^{\Omega(\frac{1}{\sigma})}$. In particular there is no sparse partition scheme with parameters smaller than $(\Omega(\log n/\log\log n),\Omega(\log n))$. This implies that both our results and [45] are tight up to second order terms. Although we do not provide any lower bound for scattering partitions, we present some evidence that general graphs are not $(O(1),O(1))$-scatterable. Specifically, we define a stronger notion of partitions called *super-scattering* and show that general graphs are not $(1,\Omega(\log n))$-super scatterable.

Trees. Trees are the most basic of the restricted graph families. Weak sparse partitions for trees follows from the existence of sparse covers. Nevertheless, in order to improve parameters and understanding we construct $(4,3)$-weak sparse partition scheme for trees. Further, we prove that trees are $(2,3)$-scatterable. Finally, we show that there are no good strong sparse partition for trees. Specifically, we prove that if all n-vertex trees admit (σ,τ)-strong sparse partition scheme, then $\tau\geq\frac{1}{3}\cdot n^{\frac{2}{\sigma+1}}$. This implies that for strong sparse partitions, trees are essentially as bad as general graphs.

Doubling Dimension. We prove that for every graph with doubling dimension ddim and parameter $\Delta>0$, there is a partition $\mathcal{P}$ which is simultaneously $\left(58\alpha,2^{\mathrm{ddim}/\alpha}\cdot\tilde{O}(\mathrm{ddim}),\Delta\right)$-strong sparse partition for all parameters $\alpha\geq 1$. Note that this implies an $\left(O(\mathrm{ddim}),\tilde{O}(\mathrm{ddim})\right)$-strong sparse partition scheme.

Euclidean Space. We prove that the d-dimensional Euclidean space $(\mathbb{R}^d,\|\cdot\|_2)$ is $(1,2d)$-scatterable [13], while for every (σ,τ)-weak sparse partition scheme it holds that $\tau>(1+\frac{1}{2\sigma})^d$. In particular, if σ is at most a constant, then τ must be exponential. This provides an interesting example of a family where scattering partitions have considerably better parameters than sparse partitions.

[13] In Euclidean space, we say that a partition is (σ,τ,Δ)-scattering if every interval of length Δ/σ intersects at most τ clusters.

SPDdepth.[14] We prove that every graph with SPDdepth ρ (in particular graph with pathwidth ρ) admit $\big(O(\rho), O(\rho^2)\big)$-strong sparse partition scheme. Further, we prove that such graphs admit $(8, 5\rho)$-weak sparse partition scheme.

Chordal Graphs. We prove that every Chordal graph is $(2, 3)$-scatterable.

Cactus Graphs. We prove that every Cactus graph is $(4, 5)$-scatterable.

1.3 Technical Ideas

Scattering Partition Imply SPR. Similarly to previous works on the SPR problem, we construct a minor via a terminal partition. That is, a partition of V into k connected clusters, where each cluster contains a single terminal. The minor is then induced by contracting all the internal edges. Intuitively, to obtain small distortion, one needs to ensure that every Steiner vertex is clustered into a terminal not much further than its closest terminal, and that every shortest path between a pair of terminals intersects only a small number of clusters. However, the local partitioning of each area in the graph requires a different scale, according to the distance to the closest terminal. Our approach is similar in spirit to the algorithm of Englert et al. [25], who constructed a minor with small expected distortion[15] using stochastic decomposition for all possible distance scales. We however, work in the more restrictive regime of worst case distortion guarantee. Glossing over many details, we create different scattering partitions to different areas, where vertices at distance $\approx \Delta$ to the terminal set are partitioned using a $(1, \tau, \Delta)$-scattering partition. Afterwards, we assemble the different clusters from the partitions in all possible scales into a single terminal partition. We use the scattering property twice. First to argue that each vertex v is clustered to a terminal at distance at most $O(\tau) \cdot D(v)$ (here $D(v)$ is the distance to the closest terminal). Second, to argue that every shortest path where all the vertices are at similar distance to the terminal set, intersect the clusters of at most $O(\tau^2)$ terminals.

1.4 Related Work

In the functional analysis community, the notion of *Nagata dimension* was studied. The Nagata dimension of a metric space (X, d), $\dim_N X$, is the infimum over all integers n such that there exists a constant c s.t. X admits a $(c, n+1)$-weak sparse partition scheme. In contrast, in this paper our goal is to minimize this constant c. See [53] and the references therein.

A closely related problem to UST is the *Universal Traveling Salesman Problem* (UTSP). Consider a postman providing post service for a set X of clients with n different locations (with distance measure d_X). Each morning the postman receives a subset $S \subset X$ of the required deliveries for the day. In order to minimize the total tour length, one solution may be to compute each morning an (approximation of an) Optimal TSP tour for the set S. An alternative solution will be to compute a *Universal TSP* (UTSP) tour. This is a universal tour R containing all the points X. Given a subset S, $R(S)$ is the tour visiting all the points in S w.r.t. the order induced by R. Given a tour T denote its length by $|T|$. The *stretch* of R is the maximum ratio among all subsets $S \subseteq X$ between the length of $R(S)$ and the length of the optimal TSP tour on S, $\max_{S\subseteq X} \frac{|R(S)|}{|\mathrm{Opt}(S)|}$.

[14] Every (weighted) path graph has an SPDdepth 1. A graph G has an SPDdepth ρ if there exist a *shortest path* P, such that every connected component in $G \setminus P$ has an SPDdepth $\rho - 1$. This family includes graphs with pathwidth at most ρ, and more. See [2].

[15] A distribution $\mathcal{D}$ over solutions to the SPR problem has expected distortion α if $\forall t, t' \in K$, $\mathbb{E}_{M\sim\mathcal{D}}[d_M(t, t')] \le \alpha \cdot d_G(t, t')$.

All the sparse partition based upper bounds for the UST problem translated directly to the UTSP problem with the same parameters. The first to study the problem were Platzman and Bartholdi [57], who given n points in the Euclidean plane constructed a solution with stretch $O(\log n)$, using *space filling curves*. Recently, Christodoulou, and Sgouritsa [18] proved a lower and upper bound of $\Theta(\log n/\log\log n)$ for the $n \times n$ grid, improving a previous $\Omega(\sqrt[6]{\log n/\log\log n})$ lower bound of Hajiaghayi, Kleinberg, and Leighton [42] (and the $O(\log n)$ upper bound of [57]). For general n vertex graphs Gupta et al. [39] proved an $O(\log^2 n)$ upper bound, while Gorodezky, Kleinberg, Shmoys, and Spencer [36] proved an $\Omega(\log n)$ lower bound. From the computational point of view, Schalekamp and Shmoys [59] showed that if the input graph is a tree, an UTSP with optimal stretch can be computed efficiently.

The *A Priori TSP* problem is similar to the UTSP problem. In addition there is a distribution $\mathcal{D}$ over subsets $S \subseteq V$ and the stretch of tour a R is the expected ratio between the induced solution to optimal $\mathbb{E}_{S\sim\mathcal{D}}\frac{|R(S)|}{|\mathrm{Opt}(S)|}$ (instead of a worst case like in UTSP). Similarly, *A Priori Steiner Tree* was studied (usually omitting rt from the problem). See [44, 59, 36] for further details. Another similar problem is the *Online (or dynamic) Steiner Tree* problem. Here the set S of vertices that should be connected is evolving over time, see [43, 6, 37] and references therein.

Unlike the definition used in this paper (taken from [45]), sparse partitions were also defined in the literature as partitions where only a small fraction of the edges are inter-cluster (see for example [5]). A closely related notion to sparse partitions are padded and separating decompositions. A graph G is β-decomposable if for every $\Delta > 0$, there is a distribution $\mathcal{D}$ over Δ bounded partitions such that for every $u, v \in V$, the probability that u and v belong to different clusters is at most $\beta \cdot \frac{d_G(u,v)}{\Delta}$. Note that by linearity of expectation, a path $\mathcal{I}$ of length Δ/σ intersects at most $1 + \beta/\sigma$ clusters in expectation. For comparison, in scattering partition we replace the distribution by a single partition and receive a bound on the number of intersections in the worst case. See [48, 9, 26, 40, 1, 5, 4, 34, 28] for further details.

Englert et al. [25] showed that every graph which is β-decomposable, admits a distribution $\mathcal{D}$ over solution to the SPR problem with expected distortion $O(\beta\log\beta)$. [15] In particular this implies constant expected distortion for graphs excluding a fixed minor, or bounded doubling dimension.

For a set K of terminals of size k, Krauthgamer, Nguyen and Zondiner [50] showed that if we allow the minor M to contain at most $\binom{k}{2}^2$ Steiner vertices (in addition to the terminals), then distortion 1 can be achieved. They further showed that for graphs with constant treewidth, $O(k^2)$ Steiner points will suffice for distortion 1. Cheung, Gramoz and Henzinger [17] showed that allowing $O(k^{2+\frac{2}{t}})$ Steiner vertices, one can achieve distortion $2t - 1$. For planar graphs, Cheung et al. al. achieved $1 + \epsilon$ distortion with $\tilde{O}((\frac{k}{\epsilon})^2)$ Steiner points.

There is a long line of work focusing on preserving the cut/flow structure among the terminals by a graph minor. See [56, 54, 15, 55, 25, 19, 51, 7, 35, 52].

There were works studying metric embeddings and metric data structures concerned with preserving distances among terminals, or from terminals to other vertices, out of the context of minors. See [20, 58, 41, 47, 21, 22, 10, 23, 49, 32, 24].

2 Preliminaries

All the logarithms in the paper are in base 2. We use $\tilde{O}$ notation to suppress constants and logarithmic factors, that is $\tilde{O}(f(j)) = f(j) \cdot \mathrm{polylog}(f(j))$.

Graphs. We consider connected undirected graphs $G = (V, E)$ with edge weights $w : E \to \mathbb{R}_{\geq 0}$. Let d_G denote the shortest path metric in G. $B_G(v, r) = \{u \in V \mid d_G(v, u) \leq r\}$ is the ball of radius r around v. For a vertex $v \in V$ and a subset $A \subseteq V$, let $d_G(x, A) := \min_{a \in A} d_G(x, a)$, where $d_G(x, \emptyset) = \infty$. For a subset of vertices $A \subseteq V$, let $G[A]$ denote the induced graph on A, and let $G \setminus A := G[V \setminus A]$.

Special graph families. A graph H is a *minor* of a graph G if we can obtain H from G by edge deletions/contractions, and vertex deletions. A graph family $\mathcal{G}$ is *H-minor-free* if no graph $G \in \mathcal{G}$ has H as a minor. Some examples of minor free graph families are planar graphs (K_5 and $K_{3,3}$ free), outerplanar graphs (K_4 and $K_{3,2}$ free), series-parallel graphs (K_4 free), Cactus graphs (also known as tree of cycles) (⧅ free), and trees (K_3 free).

Given a graph $G = (V, E)$, a *tree decomposition* of G is a tree T with nodes $B_1, \dots, B_s$ (called *bags*) where each B_i is a subset of V such that the following properties hold:

- For every edge $\{u, v\} \in E$, there is a bag B_i containing both u and v.
- For every vertex $v \in V$, the set of bags containing v form a connected subtree of T.

The *width* of a tree decomposition is $\max_i\{|B_i| - 1\}$. The *treewidth* of G is the minimal width of a tree decomposition of G. A *path decomposition* of G is a special kind of tree decomposition where the underlying tree is a path. The *pathwidth* of G is the minimal width of a path decomposition of G.

Chordal graphs are unweighted graphs where each cycle of length greater then 4 contains a chordal. In other words, if the induced graph on a set of vertices V' is the cycle graph, than necessarily $|V'| \leq 3$. Chordal graphs contain interval graphs, subtree intersection graphs and other interesting sub families. A characterization of Chordal graphs is that they have a tree decomposition such that each bag is a clique. That is, there is a tree decomposition T of G where there is no upper bound on the size of a bag, but for every bag $B \in T$ the induced graph $G[B]$ is a clique.

A *Cactus graph* (a.k.a. tree of cycles) is a graph where each edge belongs to at most one simple cycle. Alternatively it can be defined as the graph family that excludes K_4 minus an edge (⧅) as a minor.

Abraham et al. [2] defined *shortest path decompositions* (SPDs) of "low depth". Every (weighted) path graph has an SPDdepth 1. A graph G has an SPDdepth k if there exist a *shortest path* P, such that every connected component in $G \setminus P$ has an SPDdepth $k - 1$. In other words, given a graph, in SPD we hierarchically delete shortest paths from each connected component, until no vertices remain. See [2] for formal definition (or full version [31]). Every graph with pathwidth ρ has SPDdepth at most $\rho + 1$, treewidth ρ implies SPDdepth at most $O(\rho \log n)$, and every graph excluding a fixed minor has SPDdepth $O(\log n)$. See [2, 30] for further details and applications.

3 From Scattering Partitions to SPR: Proof of Theorem 2

We will assume w.l.o.g. that the minimal pairwise distance in the graph is exactly 1, otherwise we can scale all the weights accordingly. The set of terminals denoted $K = \{t_1, \dots, t_k\}$. For every vertex $v \in V$, denote by $D(v) = d_G(v, K)$ the distance to its closest terminal. Note that $\min_{v \in V \setminus K} D(v) \geq 1$.

Similarly to previous papers on the SPR problem, we will create a minor using *terminal partitions*. Specifically, we partition the vertices into k connected clusters, with a single terminal in each cluster. Such a partition induces a minor by contracting all the internal edges in each cluster. More formally, a partition $\{V_1, \dots, V_k\}$ of V is called a terminal partition (w.r.t to K) if for every $1 \leq i \leq k$, $t_i \in V_i$, and the induced graph $G[V_i]$ is connected. For a

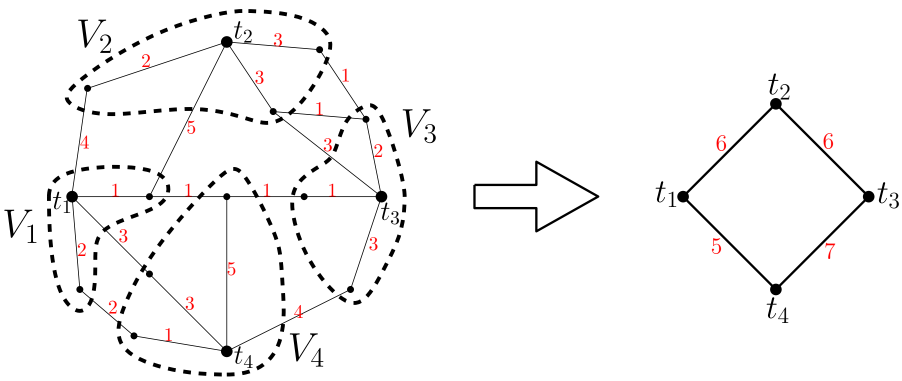

Figure 2 The left side of the figure contains a weighted graph $G = (V, E)$, with weights specified in red, and four terminals $\{t_1, t_2, t_3, t_4\}$. The dashed black curves represent a terminal partition of the vertex set V into the subsets V_1, V_2, V_3, V_4. The right side of the figure represents the minor M induced by the terminal partition. The distortion is realized between t_1 and t_3, and is $\frac{d_M(t_1,t_3)}{d_G(t_1,t_3)} = \frac{12}{4} = 3$.

vertex $v \in V_i$, we say that v is assigned to t_i. See Figure 2 for an illustration. The *induced minor* by the terminal partition $\{V_1, \dots, V_k\}$, is a minor M, where each set V_i is contracted into a single vertex called (abusing notation) t_i. Note that there is an edge in M from t_i to t_j if and only if there are vertices $v_i \in V_i$ and $v_j \in V_j$ such that $\{v_i, v_j\} \in E$. We determine the weight of the edge $\{t_i, t_j\} \in E(M)$ to be $d_G(t_i, t_j)$. Note that by the triangle inequality, for every pair of (not necessarily neighboring) terminals t_i, t_j, it holds that $d_M(t_i, t_j) \geq d_G(t_i, t_j)$. The *distortion* of the induced minor is $\max_{i,j} \frac{d_M(t_i,t_j)}{d_G(t_i,t_j)}$.

3.1 Algorithm

For $i \geq 1$, set $\mathcal{R}_i = \{v \in V \mid 2^{i-1} \leq D(v) < 2^i\}$ to be the set of vertices at distance between 2^{i-1} and 2^i from K. Set $\mathcal{R}_0 = K$. We create the terminal partition in an iterative manner, where initially each set $V_i = \{t_i\}$ is a singleton, and gradually more vertices are joining. We will denote the stage of the terminal partition after i steps, using a function $f_i : V \to K \cup \{\perp\}$. For a yet unassigned vertex v we write $f_i(v) = \perp$, otherwise the vertex v will be assigned to $f_i(v)$. Initially for every terminal t_j, $f_0(t_j) = t_j$ while for every Steiner vertex $v \in V \setminus K$, $f_0(v) = \perp$. In iteration i we will define f_i by "extending" f_{i-1}. That is, unassigned vertices may be assigned (i.e., for v such that $f_{i-1}(v) = \perp$ it might be $f_i(v) = t_j$), while the function will remain the same on the set of assigned vertices ($f_{i-1}(v) \neq \perp \Rightarrow f_i(v) = f_{i-1}(v)$). We will guarantee that all the vertices in $\mathcal{R}_i$ will be assigned in f_i. In particular, after $\log(\max_v D(v))$ steps, all the vertices will be assigned. Denote by V_i the set of vertices assigned by f_i. Initially $V_0 = K = \mathcal{R}_0$. By induction we will assume that $\cup_{j \leq i-1} \mathcal{R}_j \subseteq V_{i-1}$. Let $G_i = G[V \setminus V_{i-1}]$ be the graph induced by the set of yet unassigned vertices. Fix $\Delta_i = 2^{i-1}$. Let $\mathcal{P}_i$ be an $(1, \tau, \Delta_i)$-scattering partition of G_i. Let $\mathcal{C}_i \subseteq \mathcal{P}_i$ be the set of clusters C which contain at least one vertex $v \in \mathcal{R}_i$. All the vertices in $\cup \mathcal{C}_i$ will be assigned by f_i.

We say that a cluster $C \in \mathcal{C}_i$ is at level 1, noting $\delta_i(C) = 1$, if there is an edge $\{v, u_C\}$ (in G) from a vertex $v \in C$ to a vertex $u_C \in V_{i-1}$ of weight at most 2^i. In general, $\delta_i(C) = l$, if l is the minimal index such that there is an edge $\{v, u_C\}$ from a vertex $v \in C$ to a vertex $u_C \in C'$ of weight at most 2^i, such that $\delta_i(C') = l - 1$. In both cases u_C is called the *linking* vertex of C. Next, we define f_i based on f_{i-1}. For every vertex $v \in V_{i-1}$ set $f_i(v) = f_{i-1}(v)$.

For every vertex not in $\cup\mathcal{C}_i$ (or V_{i-1}) set $f_i(v)=\perp$. For a cluster $C\in\mathcal{C}_i$ s.t. $\delta_i(C)=1$, let $u_C\in V_{i-1}$ be its linking vertex. For every $v\in C$ set $f_i(v)=f_i(u_C)$. Generally, for level l suppose that f_i is already defined on all the clusters of level $l-1$. Let $C\in\mathcal{C}_i$ s.t. $\delta_i(C)=l$. Let u_C be the linking vertex of C. For every $v\in C$, set $f_i(v)=f_i(u_C)$. Note that for every cluster, all the vertices are mapped to the same terminal. This finishes the definition of f_i.

The algorithm continues until there is f_i where all the Steiner vertices are assigned. Set $f=f_i$. The algorithm returns the terminal-centered minor M of G induced by $\{f^{-1}(t_1),\dots,f^{-1}(t_k)\}$.

3.2 Basic Properties

It is straightforward from the construction that $f^{-1}(t_1),\dots,f^{-1}(t_k)$ define a terminal partition. We will prove that every vertex v will be assigned during either iteration $\lceil\log D(v)\rceil$ or $\lceil\log D(v)\rceil-1$ (Claim 9), to a terminal at distance at most $O(\tau)\cdot D(v)$ from v (Corollary 8). We begin by arguing that in each iteration, the maximum possible level of a cluster is τ.

▷ Claim 6. For every cluster $C\in\mathcal{C}_i$, $\delta_i(C)\le\tau$.

Proof. Consider a cluster $C\in\mathcal{C}_i$, and let $v\in C$ be a vertex s.t. $D(v)\le 2^i$. Let $P=\{v=v_0,\dots,v_s\}$ be a prefix of the $D(v)$ length path from v to its closest terminal such that v_s has a neighbor in V_{i-1}. Note that P has (weighted) length at most $2^{i-1}=\Delta_i$ (as all vertices v' for which $D(v')\le 2^{i-1}$ are necessarily clustered). $\mathcal{P}_i$ is a $(1,\tau,\Delta_i)$-scattering partition. Hence the vertices of P are partitioned to $\tau'\le\tau$ clusters $C_1,\dots,C_{\tau'}$ where $v_s\in C_1$, $v_0\in C_{\tau'}$ and there is an edge from C_j to C_{j+1} of weight at most $2^{i-1}<2^i$, while the edge from v_s towards V_{i-1} is of weight at most 2^i. It holds that $\delta_i(C_1)=1$, and by induction $\delta(C_j)\le j$. In particular $\delta(C)\le\tau'\le\tau$. ◁

▷ Claim 7. For every vertex v which is assigned during the i'th iteration (i.e., $v\in C\in\mathcal{C}_i$) it holds that $d_G(v,f(v))\le 3\tau\cdot 2^i$.

Proof. The proof is by induction on i. For $i=0$ the assertion holds trivially as every terminal is assigned to itself. We will assume the assertion for $i-1$ and prove it for i. Let $C\in\mathcal{C}_i$ be some cluster, and let $v\in C$. Suppose first that $\delta_i(C)=1$. Let $u_C\in V_{i-1}$ be the linking vertex of C. By the induction hypothesis $d_G(u_C,f(u_C))\le 3\tau\cdot 2^{i-1}$. As the diameter of C is bounded by 2^{i-1}, and the weight of the edge towards u_C is at most 2^i we conclude $d_G(v,f(v))\le d_G(v,u_C)+d_G(u_C,f(u_C))\le(2^{i-1}+2^i)+3\tau\cdot 2^{i-1}=3\cdot 2^{i-1}+3\tau\cdot 2^{i-1}$. Generally, for $\delta_i(C)=l$, we argue by induction that for every $v\in C$ it holds that $d_G(v,f(v))\le l\cdot 3\cdot 2^{i-1}+3\tau\cdot 2^{i-1}$. Indeed, let u_C by the linking vertex of C. By the induction hypothesis it holds that $d_G(u_C,f(u_C))\le(l-1)\cdot 3\cdot 2^{i-1}+3\tau\cdot 2^{i-1}$. Using similar arguments, it holds that $d_G(v,f(v))\le d_G(v,u_C)+d_G(u_C,f(u_C))\le(2^{i-1}+2^i)+(l-1)\cdot 3\cdot 2^{i-1}+3\tau\cdot 2^{i-1}=l\cdot 3\cdot 2^{i-1}+3\tau\cdot 2^{i-1}$. Using Claim 6, $d_G(v,f(v))\le 3\tau\cdot 2^{i-1}+3\tau\cdot 2^{i-1}=3\tau\cdot 2^i$ as required. ◁

► **Corollary 8.** *For every vertex v it holds that $d_G(v,f(v))<6\tau\cdot D(v)$.*

Proof. Let $i\ge 0$ such that $2^{i-1}<D(v)\le 2^i$. The vertex v is assigned at iteration i or earlier. By Claim 7 we conclude $d_G(v,f(v))\le 3\tau\cdot 2^i<6\tau\cdot D(v)$. ◀

▷ Claim 9. Consider a vertex v such that $2^{i-1}<D(v)\le 2^i$. Then v is assigned either at iteration $i-1$ or i.

Proof. Clearly if v remains un-assigned until iteration i, it will be assigned during the i'th iteration. Suppose that v was assigned during iteration j. Then v belongs to a cluster $C \in \mathcal{C}_j$. In particular there is a vertex $u \in C$ such that $D(u) \leq 2^j$. As C has diameter at most 2^{j-1}, it holds that

$$2^{i-1} < D(v) \leq D(u) + d_G(v,u) \leq 2^j + 2^{j-1} = 3 \cdot 2^{j-1} \ .$$

i,j are integers, hence $j \geq i-1$. $\triangleleft$

3.3 Distortion Analysis

In this section we analyze the distortion of the minor induced by the terminal partition created by our algorithm. We have several variables that are defined with respect to the algorithm. Note that all these definitions are for analysis purposes only, and have no impact on the execution of the algorithm. Consider a pair of terminals t and t'. Let $P_{t,t'} = \{t = v_0, \dots, v_\gamma = t'\}$ be the shortest path from t to t' in G. We can assume that there are no terminals in $P_{t,t'}$ other than t, t'. This is because if we will prove the distortion guarantee for every pair of terminals t, t' such that $P_{t,t'} \cap K = \{t, t'\}$, then by the triangle inequality the distortion guarantee will hold for all terminal pairs.

Detours. The terminals t, t' are fixed. During the execution of the algorithm, for every terminal t_j we will maintain a *detour* $\mathcal{D}_{t_j}$ (or shortly $\mathcal{D}_j$). A detour is a consecutive subinterval $\{a_j, \dots, b_j\}$ of $P_{t,t'}$, where $a_j \in \mathcal{D}_j$ is the leftmost (i.e., with minimal index) vertex in the detour and b_j is the rightmost. Initially $\mathcal{D}_t = \{t\}$ and $\mathcal{D}_{t'} = \{t'\}$, while for every $t_j \notin \{t, t'\}$, $\mathcal{D}_j = \emptyset$. Every pair of detours $\mathcal{D}_j, \mathcal{D}_{j'}$ will be disjoint throughout the execution of the algorithm.

A vertex $v \in P_{t,t'}$ is *active* if and only if it does not belong to any detour. It will hold that every active vertex is necessarily unassigned (while there might be unassigned vertices which are inactive). Initially, t, t' are *inactive*, while all the other vertices of $P_{t,t'}$ are active. Consider the i'th iteration of the algorithm. We go over the terminals according to an arbitrary order $\{t_1, \dots, t_k\}$. Consider the terminal t_j with detour $\mathcal{D}_j = \{a_j, \dots, b_j\}$ (which might be empty). If no active vertices are assigned to t_j we do nothing. Otherwise, let $a'_j \in P_{t,t'}$ (resp. b'_j) be the leftmost (resp. rightmost) active vertex that was assigned to t_j during the i'th iteration. Set a_j to be vertex with minimal index between the former a_j and a'_j (a'_j if there was no a_j). Similarly b_j is the vertex with maximal index between the former b_j and b'_j. $\mathcal{D}_j$ is updated to be $\{a_j, \dots, b_j\}$. All the vertices in $\{a_j, \dots, b_j\} = \mathcal{D}_j$ become inactive. Note that a vertex might become inactive while remaining yet unassigned.

Consider an additional detour $\mathcal{D}_{j'}$. Before the updating of $\mathcal{D}_j$ at iteration i, $\mathcal{D}_j, \mathcal{D}_{j'}$ are disjoint. If a'_j, b'_j were active they cannot belong to $\mathcal{D}_{j'}$. Thus after the update, a_j, b_j did not belong to $\mathcal{D}_{j'}$ as well. However, it is possible that after the update $\mathcal{D}_j$ and $\mathcal{D}_{j'}$ are no longer disjoint. The only such possibility is when $\mathcal{D}_{j'} \subset \mathcal{D}_j$. In such a case, we set $\mathcal{D}_{j'} \leftarrow \emptyset$, maintaining the disjointness property (while not changing the (in)active status of any vertex).

After we nullify all the detours that were contained in $\mathcal{D}_j$, we will proceed to treat the next terminals in turn. Once we finish going over all the terminals, we proceed to the $i+1$ iteration. Eventually, all the vertices cease to be active, and in particular belong to some detour. In other words, all the vertices of $P_{t,t'}$ are partitioned to consecutive disjoint detours $\mathcal{D}_{\ell_1}, \dots, \mathcal{D}_{\ell_s}$.

Intervals. For an *interval* $Q=\{v_a,\dots,v_b\}\subseteq P_{t,t'}$, the *internal length* is $L(Q)=d_G(v_a,v_b)$, while the *external length* is $L^+(Q)=d_G(v_{a-1},v_{b+1})$.[16] We denote by $D(Q)=D(v_a)$ the distance from the leftmost vertex $v_a\in Q$ to its closest terminal. Set $c_{\text{int}}=\frac{1}{7}$ ("int" for interval). We partition the vertices in $P_{t,t'}$ into consecutive intervals $\mathcal{Q}$, such that for every $Q\in\mathcal{Q}$,

$$L(Q)\le c_{\text{int}}\cdot D(Q)\le L^+(Q)\ . \tag{3.1}$$

Such a partition could be obtained as follows: Sweep along the path $P_{t,t'}$ in a greedy manner, after partitioning the prefix $v_0,\dots,v_{h-1}$, to construct the next interval Q, simply pick the minimal index s such that $L^+(\{v_h,\dots,v_{h+s}\})\ge c_{\text{int}}\cdot D(v_h)$. By the minimality of s, $L(\{v_h,\dots,v_{h+s}\})\le L^+(\{v_h,\dots,v_{h+s-1}\})\le c_{\text{int}}\cdot D(v_h)$ (in the case $s=0$, trivially $L(\{v_h\})=0\le c_{\text{int}}\cdot D(v_h)$). Note that such s could always be found, as $L^+(\{v_h,\dots,v_\gamma=t'\})=d_G(v_{h-1},t')\ge d_G(v_h,t')\ge D(v_h)=D(Q)$.

Consider some interval $Q=\{v_a,\dots,v_b\}\in\mathcal{Q}$. For every vertex $v\in Q$, by triangle inequality it holds that $D(Q)-L(Q)\le D(v)\le D(Q)+L(Q)$. Therefore,

$$(1-c_{\text{int}})D(Q)\le D(v)\le(1+c_{\text{int}})D(Q)\ . \tag{3.2}$$

Note that the set $\mathcal{Q}$ of intervals is determined before the execution of the algorithm, and is never changed. In particular, it is independent from the set of detours (which evolves during the execution of the algorithm).
For an interval Q, we denote by i_Q the first iteration when some vertex v belonging to the interval Q is assigned.

▷ Claim 10. All Q vertices are assigned in either iteration i_Q or i_Q+1.

Proof. Let $u\in Q$ be some vertex which is assigned during iteration i_Q. Then u belongs to a cluster $C\in\mathcal{C}_{i_Q}$, containing a vertex $u'\in C$ such that $D(u')\le 2^{i_Q}$. As C has diameter at most 2^{i_Q-1}, it holds that $2^{i_Q}\ge D(u')\ge D(u)-d_G(u,u')\ge D(u)-2^{i_Q-1}$. Hence $D(u)\le\frac{3}{2}\cdot 2^{i_Q}$. It follows that

$$D(Q)\overset{(3.2)}{\le}\frac{1}{1-c_{\text{int}}}\cdot D(u)\le\frac{3}{2}\cdot\frac{1}{1-c_{\text{int}}}\cdot 2^{i_Q}\ . \tag{3.3}$$

For every vertex $v\in Q$ it holds that,

$$D(v)\le D(Q)+L(Q)\overset{(3.2)}{\le}(1+c_{\text{int}})\cdot D(Q\overset{(3.3)}{\le}\frac{3}{2}\cdot\frac{(1+c_{\text{int}})}{(1-c_{\text{int}})}\cdot 2^{i_Q}=2^{i_Q+1}\ .$$

Therefore, in the i_Q+1 iteration, all the (yet unassigned) vertices of Q will necessarily be assigned. ◁

▶ **Lemma 11.** *Consider an interval $Q\in\mathcal{Q}$. Then the vertices of Q are partitioned into at most $O(\tau^2)$ different detours.*

Proof. By definition, by the end of the i_Q-1'th iteration all the vertices of Q are unassigned. We first consider the case where by the end i_Q-1'th iteration some vertex $v\in Q$ is inactive. It holds that v belongs to some detour $\mathcal{D}_j$. As all the vertices of Q are unassigned, necessarily $Q\subset\mathcal{D}_j$. In particular, all the vertices of Q belong to a single detour. This property will not change till the end of the algorithm, thus the lemma follows.

[16] For ease of notation we will denote $v_{-1}=t$ and $v_{\gamma+1}=t'$.

Next, we consider the case where by the end of the $i_Q - 1$'th iteration all the vertices of Q are active. The algorithm at iteration i_Q creates an $\left(1,\tau,\Delta_{i_Q}\right)$-scattering partition $\mathcal{P}_{i_Q}$. The length of Q is bounded by

$$L(Q) \overset{(3.1)}{\leq} c_{\text{int}} \cdot D(Q) \overset{(3.3)}{\leq} c_{\text{int}} \cdot \frac{3}{2} \cdot \frac{1}{1-c_{\text{int}}} \cdot 2^{i_Q} = \frac{1}{4} \cdot 2^{i_Q} < \Delta_{i_Q} \tag{3.4}$$

Hence Q is partitioned by $\mathcal{P}_{i_Q}$ to $\tau' \leq \tau$ clusters $C_1,\dots,C_{\tau'} \in \mathcal{P}_{i_Q}$. It follows that by the end of the i_Q'th iteration, the inactive vertices in Q are partitioned to at most τ detours. If all the vertices in Q become inactive, then we are done, as the number of detours covering Q can only decrease further in the algorithm (as a result of detour nullification). Hence we will assume that some of Q vertices remain active.

A *slice* is a maximal sub-interval $S \subseteq Q$ of active vertices. The active vertices in Q are partitioned to at most $\tau+1$ slices $S_1, S_2, \dots, S_{\tau''}$.[17] By the end of the i_Q+1 iteration, according to Claim 10 all Q vertices will be assigned, and in particular belong to some detour. The algorithm creates a $\left(\Delta_{i_Q+1},\tau,1\right)$-scattering partition $\mathcal{P}_{i_Q+1}$ of the unassigned vertices. By equation (3.4) the length of every slice S is bounded by $L(S) \leq L(Q) \leq \frac{1}{4} \cdot 2^{i_Q} \leq \Delta_{i_Q+1}$. Therefore the vertices S intersect at most τ clusters of $\mathcal{P}_{i_Q+1}$, and thus will be partitioned to at most τ detours. Some detours might get nullified, however in the worst case, by the end of the i_Q+1 iteration, the vertices in $\cup_i S_i$ are partitioned to at most $\tau \cdot (\tau+1)$ detours. In particular all the vertices in Q are partitioned to at most $O(\tau^2)$ detours. As the number of detours covering Q can only decrease further in the algorithm, the lemma follows. ◀

By the end of algorithm, we will *charge* the intervals for the detours. Consider the detour $\mathcal{D}_j = \{a_j,\dots,b_j\}$ of t_j. Let $Q_j \in \mathcal{Q}$ be the interval containing a_j. We will charge Q_j for the detour $\mathcal{D}_j$. Denote by $X(Q)$ the number of detours for which the interval Q is charged for. By Lemma 11, $X(Q) = O(\tau^2)$ for every interval $Q \in \mathcal{Q}$.

Recall that by the end of the algorithm, all the vertices of $P_{t,t'}$ are partitioned to consecutive disjoint detours $\mathcal{D}_{\ell_1},\dots,\mathcal{D}_{\ell_s}$, where $\mathcal{D}_{\ell_j} = \{a_{\ell_j},\dots,b_{\ell_j}\}$ and a_{ℓ_j}, b_{ℓ_j} belong to the cluster of t_{ℓ_j}. In particular $t_{\ell_1} = t$ and $t_{\ell_s} = t'$, as each terminal belongs to the cluster of itself. Moreover, for every $j < s$, there is an edge $\{b_{\ell_j}, a_{\ell_{j+1}}\}$ in G between the cluster of t_{ℓ_j} to that of $t_{\ell_{j+1}}$. Therefore, in the minor induced by the partition there is an edge between t_{ℓ_j} to $t_{\ell_{j+1}}$. We conclude

$$\begin{aligned} d_M(t,t') &\leq \sum_{j=1}^{s-1} d_G(t_{\ell_j}, t_{\ell_{j+1}}) \leq \sum_{j=1}^{s-1} \left[d_G(t_{\ell_j}, a_{\ell_j}) + d_G(a_{\ell_j}, a_{\ell_{j+1}}) + d_G(a_{\ell_{j+1}}, t_{\ell_{j+1}})\right] \\ &\leq \sum_{j=1}^{s-1} d_G(a_{\ell_j}, a_{\ell_{j+1}}) + 2\sum_{j=1}^{s} d_G(t_{\ell_j}, a_{\ell_j}) \ . \end{aligned}$$

Note that $\sum_{j=1}^{s-1} d_G(a_{\ell_j}, a_{\ell_{j+1}}) \leq d_G(t,t')$ as $P_{t,t'}$ is a shortest path. Denote by Q_{ℓ_j} the interval containing a_{ℓ_j}. By Corollary 8,

$$d_G(t_{\ell_j}, a_{\ell_j}) = d_G(a_{\ell_j}, f(a_{\ell_j})) \leq O(\tau) \cdot D(a_{\ell_j}) \overset{(3.2)}{=} O(\tau) \cdot D(Q_{\ell_j}) \overset{(3.1)}{=} O(\tau) \cdot L^+(Q_{\ell_j}) \ .$$

By changing the order of summation we get

$$\sum_{j=1}^{s} d_G(t_{\ell_j}, a_{\ell_j}) = O(\tau) \cdot \sum_{Q \in \mathcal{Q}} X(Q) \cdot L^+(Q) = O(\tau^3) \cdot \sum_{Q \in \mathcal{Q}} L^+(Q) \ .$$

Finally, note that $\sum_{Q \in \mathcal{Q}} L^+(Q) \leq 2 \cdot d_G(t,t')$ as every edge in $P_{t,t'}$ is counted at most twice. We conclude $d_M(t,t') \leq O(\tau^3) \cdot d_G(t,t')$. Theorem 2 now follows.

[17] Actually, as at least one Q vertex remained active, at the beginning of the i_Q+1 iteration the inactive vertices of Q partitioned to at most $\tau - 1$ detours. Therefore the maximal number of slices is τ.

4 Discussion and Open Problems

In this paper we defined scattering partitions, and showed how to apply them in order to construct solutions to the SPR problem. We proved an equivalence between sparse partitions and sparse covers. Finally, we constructed many sparse and scattering partitions for different graph families (and lower bounds), implying new results for the SPR, UST, and UTSP problems. An additional contribution of this paper is a considerable list of (all but question (5)) new intriguing open questions and conjectures.

1. **Planar graphs:** The SPR problem is most fascinating and relevant for graph families which are closed under taking a minor. Note that already for planar graphs (or even treewidth 2 graphs), the best upper bound for the SPR problem is $O(\log k)$ (same as general graphs), while the only lower bound is 8. The most important open question coming out of this paper is the following conjecture:

   ▶ **Conjecture 1.** *Every graph family excluding a fixed minor is $(O(1), O(1))$-scatterable.*

   Note that proving this conjecture for a family $\mathcal{F}$, will imply a solution to the SPR problem with constant distortion. Proving the conjecture for planar graphs will be fascinating. However, it is already open for outerplanar graphs, and graphs with treewidth 2.

2. **Scattering Partitions for General Graphs:** While we provide almost tight upper and lower bounds for sparse partitions, for scattering partitions, the story is different.

   ▶ **Conjecture 2.** *Consider an n vertex weighted graph G such that between every pair of vertices there is a unique shortet path. Then G is $(1, O(\log n))$-scatterable. Furthermore, this is tight.*

   In the full version [31], we provide some evidence that Conjecture 2 cannot be pushed further. However, any nontrivial lower bound will be interesting. Furthermore, every lower bound larger than 8 for the general SPR problem will be intriguing.

3. **Doubling graphs:** While we constructed strong sparse partition for doubling graphs (which imply scattering), it has no implication for the SPR problem. This is due to the fact that Theorem 2 required scattering partition for every induced subgraph. As induced subgraphs of a doubling graph might have unbounded doubling dimension, the proof fails to follow through. We leave the required readjustments to future work.

4. **Sparse Covers:** We classify various graph families according to the type of partitions/-covers they admit. We currently lack any example of a graph family that admits weak sparse covers but does not admit strong sparse covers. It will be interesting to find such an example, or even more so to prove that every graph that admits weak sparse cover, also has strong sparse cover with (somewhat) similar parameters.

References

1 Ittai Abraham, Yair Bartal, and Ofer Neiman. Advances in metric embedding theory. *Advances in Mathematics*, 228(6):3026–3126, 2011. doi:10.1016/j.aim.2011.08.003.

2 Ittai Abraham, Arnold Filtser, Anupam Gupta, and Ofer Neiman. Metric embedding via shortest path decompositions. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 952–963, 2018. Full version: arxiv:1708.04073. doi:10.1145/3188745.3188808.

3 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 2006), 4-7 July 2006, Lisboa, Portugal*, page 75, 2006. doi:10.1109/ICDCS.2006.72.

4 Ittai Abraham, Cyril Gavoille, Anupam Gupta, Ofer Neiman, and Kunal Talwar. Cops, robbers, and threatening skeletons: Padded decomposition for minor-free graphs. *SIAM J. Comput.*, 48(3):1120–1145, 2019. `doi:10.1137/17M1112406`.

5 Ittai Abraham, Cyril Gavoille, Dahlia Malkhi, and Udi Wieder. Strong-diameter decompositions of minor free graphs. *Theory Comput. Syst.*, 47(4):837–855, 2010. `doi:10.1007/s00224-010-9283-6`.

6 Noga Alon and Yossi Azar. On-line steiner trees in the euclidean plane. In *Proceedings of the Eighth Annual Symposium on Computational Geometry, Berlin, Germany, June 10-12, 1992*, pages 337–343, 1992. `doi:10.1145/142675.142744`.

7 Alexandr Andoni, Anupam Gupta, and Robert Krauthgamer. Towards $(1+\epsilon)$-approximate flow sparsifiers. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 279–293, 2014. `doi:10.1137/1.9781611973402.20`.

8 Baruch Awerbuch and David Peleg. Sparse partitions (extended abstract). In *31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Volume II*, pages 503–513, 1990. `doi:10.1109/FSCS.1990.89571`.

9 Yair Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996*, pages 184–193, 1996. `doi:10.1109/SFCS.1996.548477`.

10 Yair Bartal, Arnold Filtser, and Ofer Neiman. On notions of distortion and an almost minimum spanning tree with constant average distortion. *J. Comput. Syst. Sci.*, 105:116–129, 2019. `doi:10.1016/j.jcss.2019.04.006`.

11 A. Basu and A. Gupta. Steiner point removal in graph metrics. Unpublished Manuscript, available from `http://www.ams.jhu.edu/~abasu9/papers/SPR.pdf`, 2008.

12 Costas Busch, Chinmoy Dutta, Jaikumar Radhakrishnan, Rajmohan Rajaraman, and Srinivasagopalan Srivathsan. Split and join: Strong partitions and universal steiner trees for graphs. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 81–90, 2012. `doi:10.1109/FOCS.2012.45`.

13 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. `doi:10.1007/s00453-013-9757-4`.

14 T.-H. Chan, Donglin Xia, Goran Konjevod, and Andrea Richa. A tight lower bound for the steiner point removal problem on trees. In *Proceedings of the 9th International Conference on Approximation Algorithms for Combinatorial Optimization Problems, and 10th International Conference on Randomization and Computation*, APPROX'06/RANDOM'06, pages 70–81, Berlin, Heidelberg, 2006. Springer-Verlag. `doi:10.1007/11830924_9`.

15 Moses Charikar, Tom Leighton, Shi Li, and Ankur Moitra. Vertex sparsifiers and abstract rounding algorithms. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 265–274, 2010. `doi:10.1109/FOCS.2010.32`.

16 Yun Kuen Cheung. Steiner point removal - distant terminals don't (really) bother. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1353–1360, 2018. `doi:10.1137/1.9781611975031.89`.

17 Yun Kuen Cheung, Gramoz Goranci, and Monika Henzinger. Graph minors for preserving terminal distances approximately - lower and upper bounds. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 131:1–131:14, 2016. `doi:10.4230/LIPIcs.ICALP.2016.131`.

18 George Christodoulou and Alkmini Sgouritsa. An improved upper bound for the universal TSP on the grid. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 1006–1021, 2017. `doi:10.1137/1.9781611974782.64`.

19 Julia Chuzhoy. On vertex sparsifiers with steiner nodes. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 673–688, 2012. doi:10.1145/2213977.2214039.

20 Don Coppersmith and Michael Elkin. Sparse sourcewise and pairwise distance preservers. *SIAM J. Discrete Math.*, 20(2):463–501, 2006. doi:10.1137/050630696.

21 Michael Elkin, Arnold Filtser, and Ofer Neiman. Terminal embeddings. *Theor. Comput. Sci.*, 697:1–36, 2017. doi:10.1016/j.tcs.2017.06.021.

22 Michael Elkin, Arnold Filtser, and Ofer Neiman. Prioritized metric structures and embedding. *SIAM J. Comput.*, 47(3):829–858, 2018. doi:10.1137/17M1118749.

23 Michael Elkin and Ofer Neiman. Near isometric terminal embeddings for doubling metrics. In *34th International Symposium on Computational Geometry, SoCG 2018, June 11-14, 2018, Budapest, Hungary*, pages 36:1–36:15, 2018. doi:10.4230/LIPIcs.SoCG.2018.36.

24 Michael Elkin and Ofer Neiman. Lossless prioritized embeddings. *CoRR*, abs/1907.06983, 2019. To appear in SODA2020. arXiv:1907.06983.

25 Matthias Englert, Anupam Gupta, Robert Krauthgamer, Harald Räcke, Inbal Talgam-Cohen, and Kunal Talwar. Vertex sparsifiers: New results from old techniques. *SIAM J. Comput.*, 43(4):1239–1262, 2014. doi:10.1137/130908440.

26 Jittat Fakcharoenphol and Kunal Talwar. An improved decomposition theorem for graphs excluding a fixed minor. In *Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques, 6th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2003 and 7th International Workshop on Randomization and Approximation Techniques in Computer Science, RANDOM 2003, Princeton, NJ, USA, August 24-26, 2003, Proceedings*, pages 36–46, 2003. doi:10.1007/978-3-540-45198-3_4.

27 Arnold Filtser. Steiner point removal with distortion $O(\log k)$. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1361–1373, 2018. doi:10.1137/1.9781611975031.90.

28 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. doi:10.4230/LIPIcs.APPROX-RANDOM.2019.6.

29 Arnold Filtser. Steiner point removal with distortion $O(\log k)$ using the relaxed-voronoi algorithm. *SIAM J. Comput.*, 48(2):249–278, 2019. doi:10.1137/18M1184400.

30 Arnold Filtser. A face cover perspective to ℓ_1 embeddings of planar graphs. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1945–1954, 2020. doi:10.1137/1.9781611975994.120.

31 Arnold Filtser. Scattering and sparse partitions, and their applications. *CoRR*, abs/2001.04447, 2020. arXiv:2001.04447.

32 Arnold Filtser, Lee-Ad Gottlieb, and Robert Krauthgamer. Labelings vs. embeddings: On distributed representations of distances. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1063–1075, 2020. doi:10.1137/1.9781611975994.65.

33 Arnold Filtser, Robert Krauthgamer, and Ohad Trabelsi. Relaxed voronoi: A simple framework for terminal-clustering problems. In *2nd Symposium on Simplicity in Algorithms, SOSA@SODA 2019, January 8-9, 2019 - San Diego, CA, USA*, pages 10:1–10:14, 2019. doi:10.4230/OASIcs.SOSA.2019.10.

34 Arnold Filtser and Ofer Neiman. Light spanners for high dimensional norms via stochastic decompositions. In *26th Annual European Symposium on Algorithms, ESA 2018, August 20-22, 2018, Helsinki, Finland*, pages 29:1–29:15, 2018. doi:10.4230/LIPIcs.ESA.2018.29.

35 Gramoz Goranci, Monika Henzinger, and Pan Peng. Improved guarantees for vertex sparsification in planar graphs. In *25th Annual European Symposium on Algorithms, ESA 2017, September 4-6, 2017, Vienna, Austria*, pages 44:1–44:14, 2017. doi:10.4230/LIPIcs.ESA.2017.44.

36 Igor Gorodezky, Robert D. Kleinberg, David B. Shmoys, and Gwen Spencer. Improved lower bounds for the universal and *a priori* TSP. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 13th International Workshop, APPROX 2010, and 14th International Workshop, RANDOM 2010, Barcelona, Spain, September 1-3, 2010. Proceedings*, pages 178–191, 2010. doi:10.1007/978-3-642-15369-3_14.

37 Albert Gu, Anupam Gupta, and Amit Kumar. The power of deferral: Maintaining a constant-competitive steiner tree online. *SIAM J. Comput.*, 45(1):1–28, 2016. doi:10.1137/140955276.

38 Anupam Gupta. Steiner points in tree metrics don't (really) help. In *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '01, pages 220–227, Philadelphia, PA, USA, 2001. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=365411.365448.

39 Anupam Gupta, Mohammad Taghi Hajiaghayi, and Harald Räcke. Oblivious network design. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2006, Miami, Florida, USA, January 22-26, 2006*, pages 970–979, 2006. URL: http://dl.acm.org/citation.cfm?id=1109557.1109665.

40 Anupam Gupta, Robert Krauthgamer, and James R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings*, pages 534–543, 2003. doi:10.1109/SFCS.2003.1238226.

41 Anupam Gupta, Viswanath Nagarajan, and R. Ravi. An improved approximation algorithm for requirement cut. *Oper. Res. Lett.*, 38(4):322–325, 2010. doi:10.1016/j.orl.2010.02.009.

42 Mohammad Taghi Hajiaghayi, Robert D. Kleinberg, and Frank Thomson Leighton. Improved lower and upper bounds for universal TSP in planar metrics. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2006, Miami, Florida, USA, January 22-26, 2006*, pages 649–658, 2006. URL: http://dl.acm.org/citation.cfm?id=1109557.1109628.

43 Makoto Imase and Bernard M. Waxman. Dynamic steiner tree problem. *SIAM J. Discrete Math.*, 4(3):369–384, 1991. doi:10.1137/0404033.

44 Patrick Jaillet. A priori solution of a traveling salesman problem in which a random subset of the customers are visited. *Operations Research*, 36(6):929–936, 1988. doi:10.1287/opre.36.6.929.

45 Lujun Jia, Guolong Lin, Guevara Noubir, Rajmohan Rajaraman, and Ravi Sundaram. Universal approximations for tsp, steiner tree, and set cover. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 386–395, 2005. doi:10.1145/1060590.1060649.

46 Lior Kamma, Robert Krauthgamer, and Huy L. Nguyen. Cutting corners cheaply, or how to remove steiner points. *SIAM J. Comput.*, 44(4):975–995, 2015. doi:10.1137/140951382.

47 Telikepalli Kavitha and Nithin M. Varma. Small stretch pairwise spanners. In *Proceedings of the 40th International Conference on Automata, Languages, and Programming - Volume Part I*, ICALP'13, pages 601–612, Berlin, Heidelberg, 2013. Springer-Verlag. doi:10.1007/978-3-642-39206-1_51.

48 Philip N. Klein, Serge A. Plotkin, and Satish Rao. Excluded minors, network decomposition, and multicommodity flow. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, May 16-18, 1993, San Diego, CA, USA*, pages 682–690, 1993. doi:10.1145/167088.167261.

49 Robert Krauthgamer, James R. Lee, and Havana Rika. Flow-cut gaps and face covers in planar graphs. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 525–534, 2019. doi:10.1137/1.9781611975482.33.

50 Robert Krauthgamer, Huy L. Nguyen, and Tamar Zondiner. Preserving terminal distances using minors. *SIAM J. Discrete Math.*, 28(1):127–141, 2014. doi:10.1137/120888843.

51 Robert Krauthgamer and Inbal Rika. Mimicking networks and succinct representations of terminal cuts. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 1789–1799, 2013. doi:10.1137/1.9781611973105.128.

52 Robert Krauthgamer and Inbal Rika. Refined vertex sparsifiers of planar graphs. *CoRR*, abs/1702.05951, 2017. arXiv:1702.05951.

53 Urs Lang and Thilo Schlichenmaier. Nagata dimension, quasisymmetric embeddings, and lipschitz extensions. *International Mathematics Research Notices*, 2005(58):3625–3655, 2005. doi:10.1155/IMRN.2005.3625.

54 Frank Thomson Leighton and Ankur Moitra. Extensions and limits to vertex sparsification. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 47–56, 2010. doi:10.1145/1806689.1806698.

55 Konstantin Makarychev and Yury Makarychev. Metric extension operators, vertex sparsifiers and lipschitz extendability. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 255–264, 2010. doi:10.1109/FOCS.2010.31.

56 Ankur Moitra. Approximation algorithms for multicommodity-type problems with guarantees independent of the graph size. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 3–12, 2009. doi:10.1109/FOCS.2009.28.

57 Loren K. Platzman and John J. Bartholdi, III. Spacefilling curves and the planar travelling salesman problem. *J. ACM*, 36(4):719–737, October 1989. doi:10.1145/76359.76361.

58 Liam Roditty, Mikkel Thorup, and Uri Zwick. Deterministic constructions of approximate distance oracles and spanners. In *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, July 11-15, 2005, Proceedings*, pages 261–272, 2005. doi:10.1007/11523468_22.

59 Frans Schalekamp and David B. Shmoys. Algorithms for the universal and a priori TSP. *Oper. Res. Lett.*, 36(1):1–3, 2008. doi:10.1016/j.orl.2007.04.009.