

# Cops, Robbers, and Threatening Skeletons: Padded Decomposition for Minor-Free Graphs\*

Ittai Abraham<sup>†</sup>   Cyril Gavaille<sup>‡</sup>   Anupam Gupta<sup>§</sup>   Ofer Neiman<sup>¶</sup>   Kunal Talwar<sup>||</sup>

March 18, 2019

## Abstract

We prove that any graph excluding  $K_r$  as a minor can be partitioned into clusters of diameter at most  $\Delta$  while removing at most  $O(r/\Delta)$  fraction of the edges. This improves over the results of Fakcharoenphol and Talwar, who building on the work of Klein, Plotkin and Rao gave a partitioning that required to remove  $O(r^2/\Delta)$  fraction of the edges.

Our result is obtained by a new approach that relates the topological properties (excluding a minor) of a graph to its geometric properties (the induced shortest path metric). Specifically, we show that techniques used by Andreae in his investigation of the cops and robbers game on graphs excluding a fixed minor, can be used to construct padded decompositions of the metrics induced by such graphs. In particular, we get probabilistic partitions with padding parameter  $O(r)$  and strong-diameter partitions with padding parameter  $O(r^2)$  for  $K_r$ -minor-free graphs,  $O(k)$  for treewidth- $k$  graphs, and  $O(\log g)$  for graphs with (Euler) genus  $g$ .

## 1 Introduction

This paper considers the problem of constructing random partitioning schemes for minor-free graphs. Loosely speaking, the goal is to find a partition of the graph vertices so that each part (called a cluster) has small diameter, and the probability of any local neighborhood being cut (and not being contained within just one cluster) is small. There is a natural tradeoff between these two parameters (the diameter, and the probability of being cut). Such random partitions have found numerous applications in algorithm design, including: flow/cut gaps, metric embeddings, and recently as core primitives for several near linear time algorithms. Therefore improving the parameters of the partitions is a research program of considerable interest.

Tight parameters for such partitions are known in several settings. However, for the case of graphs that exclude some given graph  $H$  as a minor, the problem of finding the optimal tradeoff remains open. Progress was made in the seminal work of Klein, Plotkin and Rao [KPR93],

---

\*A preliminary version [AGG<sup>+</sup>14] of this paper appeared in STOC'14.

<sup>†</sup>VMWare. Email: [iabraham@vmware.com](mailto:iabraham@vmware.com).

<sup>‡</sup>LaBRI - University of Bordeaux, Bordeaux, France. Email: [gavoille@labri.fr](mailto:gavoille@labri.fr).

<sup>§</sup>Computer Science Department, Carnegie Mellon University, Pittsburgh, PA. Email: [anupam@cs.cmu.edu](mailto:anupam@cs.cmu.edu). Supported in part by NSF awards CCF-1016799 and CCF-1319811, and grant from the CMU-Microsoft Center for Computational Thinking.

<sup>¶</sup>Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva, Israel. Email: [neimano@cs.bgu.ac.il](mailto:neimano@cs.bgu.ac.il). Supported in part by ISF grant No. (523/12) and BSF grant No. 2015813.

<sup>||</sup>Google. Email: [kunal@kunaltalwar.org](mailto:kunal@kunaltalwar.org).

and improved by Fakcharoenphol and Talwar [FT03]. Despite attracting the attention of several researchers (see, e.g., [Lee13]), the KPR framework remained the only known approach to this problem for over 20 years.

In this paper we make progress on this question and improve known parameters. Equally importantly, we also introduce techniques and structural insights that we hope will be useful for further improvements on this and related problems. In particular, we observe that the result of Andreae [And86] can be reinterpreted as a *structure theorem* for graphs excluding a fixed minor. It constructively gives a *cop-decomposition* of a graph, which is a lot like a tree-decomposition except that instead of having  $r$  vertices per bag, it guarantees having  $r$  shortest-like paths in each bag. The cop-decomposition gives weaker structure than the beautiful work of Robertson and Seymour [RS03], but has the benefit of significantly better dependence on  $r$ . We extend this cop-decomposition framework to produce probabilistic partitions, and we believe that this high level approach may be useful in getting better algorithms for other problems involving excluded minor graphs.

We begin with some notation. For an undirected weighted graph  $G = (V, E)$  and a subset  $C \subseteq V$ , denote by  $G[C]$  the induced subgraph on  $C$ . Let  $d_G$  denote the shortest path metric on  $G$ , and for  $v \in V$  and  $t \geq 0$  define the ball  $B_G(v, t) = \{u \in V \mid d_G(v, u) \leq t\}$ . The (*weak*) *diameter* of a set  $S \subseteq V$  is  $\max_{x, y \in S} d_G(x, y)$ , whereas the *strong-diameter* of the set  $S$  is  $\max_{x, y \in S} d_{G[S]}(x, y)$  — note that the latter distance is being measured in the induced subgraph.

**Definition 1** ( $\Delta$ -bounded partitions). *A partition  $P = \{C_1, \dots, C_t\}$  of  $V$  is  $\Delta$ -bounded if for all  $i$ , the weak-diameter  $\text{diam}(C_i) \leq \Delta$ . The partition  $P$  is strong-diameter  $\Delta$ -bounded if the strong-diameter  $\text{diam}(G[C_i]) \leq \Delta$  for all  $i$ .*

Given a partition  $P = \{C_1, \dots, C_t\}$  of  $V$ , let  $P(z)$  denote the unique cluster containing  $z \in V$ .

**Definition 2.** *A distribution  $\mathcal{P}$  over  $\Delta$ -bounded partitions is  $(\beta, \delta)$ -padded if for any  $z \in V$  and any  $0 \leq \gamma \leq \delta$ ,*

$$\Pr[B_G(z, \gamma\Delta) \subseteq P(z)] \geq 2^{-\beta\gamma} .$$

*We call  $\mathcal{P}$   $\beta$ -padded if it is  $(\beta, \delta)$ -padded where  $\delta$  is a universal constant that does not depend on  $\beta$ , and efficient if it can be sampled in polynomial time.*

Our definition of padded partitions is similar to the one in [ABN11], which generalizes several definitions that appeared before, e.g. [KPR93, GKL03, AGMW10]. In particular, our definition refers to cutting balls, and not only edges, and also allows for  $\gamma > 1/\beta$ .

Our main result is the following.

**Theorem 3.** *Every  $K_r$ -minor-free graph  $G$  admits an efficient  $O(r)$ -padded partition scheme.*

It has long been known that for arbitrary graphs the best possible padding parameter is  $\Theta(\log |V|)$  [Bar96]. For special cases better bounds are known, e.g., for metrics of doubling constant  $\lambda$ , the padding parameter is  $\Theta(\log \lambda)$  [GKL03]. For graphs that can be drawn on a orientable surface of genus  $g$ , ideas developed in a recent sequence of papers [IS07, BLS10, Sid10] have culminated in the optimal padding parameter of  $\Theta(\log g)$  [LS10].

The first bounds for  $K_r$ -minor-free graphs were due to the influential work of Klein, Plotkin, and Rao [KPR93], who gave  $(O(r^3), 1/r)$ -padded partition scheme. Fakcharoenphol and Talwar [FT03] improved this to an  $(O(r^2), 1/r)$ -padded partition scheme. In this work, we improve the padding

parameter from  $O(r^2)$  to  $O(r)$ ; moreover, we provide padding guarantees to larger balls — the previous guarantees give padding only for balls of diameter  $< O(\Delta/r)$ , compared to  $O(\Delta)$  for our result. The partitioning scheme in [KPR93] was motivated by bounding the maximum-multicommodity-flow/sparsest-cut gap for  $K_r$ -minor-free graphs. Subsequently, it found applications to metric embeddings [Rao99, Rab03] with its natural connections to edge-cut problems [Mat02] and also to vertex-cut problems [FHL08], to bounding higher eigenvalues and higher-order Cheeger inequalities for graphs [BLR10, KLPT09, LGT12], to metric extension problems and approximation algorithms [CKR05, AFH<sup>+</sup>04, LN05], and others. The quantitative improvements given by our results thus give improvement in all these settings.

[Theorem 3](#) above gives us a weak-diameter guarantee. However, our techniques are versatile, and can be extended to give strong-diameter partitions — in particular, we obtain the following results.

**Theorem 4.** *Let  $G$  be an undirected weighted graph.*

1. *If  $G$  is a  $K_r$ -minor-free graph then it admits an efficient  $(O(r^2), O(1/r^2))$ -padded strong-diameter partition scheme.*
2. *If  $G$  is a tree-width  $r$  graph then it admits an efficient  $(O(r), O(1/r))$ -padded strong-diameter partition scheme.*
3. *If  $G$  is a Euler-genus  $g$  graph then it admits an efficient  $O(\log g)$ -padded strong-diameter partition scheme.*

The first result in [Theorem 4](#) is an exponential improvement over the strong-diameter partitions of Abraham et. al. [AGMW10]. The third result strengthens the result of Lee and Sidiropoulos [LS10] by providing the same asymptotic padding guarantees while ensuring that clusters have a strong-diameter. It holds for graphs embedded on orientable or non-orientable surfaces of Euler characteristic bounded by  $2 - g$  (see more details in [Section 7](#)). The second and the third results assume that the embedding of the graph (into an optimal width tree-decomposition or optimal Euler characteristic surface embedding) is given. Note that such embeddings when  $r$  or  $g$  is bounded can be determined in polynomial time (see for instance [Rec92, BK96, Moh99, KMR08]).

## 1.1 Discussion of Techniques

How does one prove a property for a graph that does not contain a  $K_r$  minor? One approach relies on the beautiful results of Robertson and Seymour that turn this negative property, namely not having a certain minor, into a positive constructive one. This gives a complete structural characterization of how such graphs are built from simple building blocks by applying simple rules to them. This structure theorem allows one to prove properties of excluded minor graphs by structural induction on the constructive procedure. On the negative side this approach typically inherits the rather bad dependence on  $r$  from the Robertson-Seymour structure theorem [RS03]. Nevertheless, this approach has been highly successful and used to prove several results for such graphs.

The other, somewhat more mysterious approach, is to work more directly and design an algorithm establishing the property, such that by failing it constructs a  $K_r$  minor. This approach is often problem-specific but usually leads to better dependence on  $r$ . Examples of this approach

include the work of Andreae [And86] for the Cops and Robbers game, results of Alon, Seymour and Thomas [AST90] on separators, and the aforementioned work of Klein, Plotkin and Rao [KPR93].

Let us now give a high-level description of some of the ideas and techniques used to prove Theorem 3 and Theorem 4.

**The Bounded Threatener Program and its probabilistic extension.** A well-studied approach to obtain  $\Delta$ -bounded  $\beta$ -padded probabilistic partitions is to find a set of “suitable” centers  $S$ , and iteratively build balls around the points in  $S$  with radii drawn from a *truncated exponential distribution* in the range  $[\Delta/4, \Delta/2]$  with rate  $\beta$ . The memoryless property of the exponential distribution ensures that balls of radius  $\approx \Delta/\beta$  around any vertex  $z$  avoid being cut with constant probability, conditioned on the exponential distribution not being truncated. To handle the truncation, we need to bound the number of centers at distance at most  $(1/2 + 1/\beta)\Delta$  from any vertex  $z$ . We will call such centers the *threateners* of  $z$ . If the number of threateners is bounded by  $2^{O(\beta)}$  then a trivial union bound implies that with constant probability: none of them will reach diameter  $(1/2 - 1/\beta)\Delta$  and hence none of them will intersect the ball  $B(z, \Delta/\beta)$ .

A contribution of this work is in extending the bounded threatener program and showing how a bound on the *expected* number of threateners suffices for obtaining probabilistic partitions.

**Cop-Decompositions.** Andreae [And86] considered the following game, a set of cops plays against a robber. At each round the robber can move across one edge and then each one of the cops can move across one edge. The cops win if they land on the same vertex as the robber. A key observation: if the robber is limited to a subgraph  $V' \subset V$  and  $P$  is a geodesic shortest path with respect to  $G[V']$  then eventually a single cop can “patrol”  $P$  and prevent the robber from even stepping on  $P$ . Using this observation, Andreae showed that if  $G$  is  $K_r$ -minor-free then  $O(r^2)$  cops have a winning strategy. The cop strategy is simple: each cop controls one shortest path and collectively they try to iteratively build a  $K_r$  minor. The shortest paths controlled by the cops induce a set of supernodes (disjoint connected subsets) and edges containing a minor that is a subgraph of  $K_r$ . At each round one can fix a center for a new supernode and use free cops to connect this center to all previous supernodes via shortest paths. The new center and each new shortest path is fully contained in the component containing the robber that is induced by removing the supernodes from  $G$  (hence these new paths are disjoint from all previous supernodes).

We view Andreae’s result as constructing a cop-decomposition of width  $O(r)$ , as we shall define now. First, recall that a tree-decomposition for  $G$  is a tree  $T$  whose nodes, called bags, are subsets of  $V$  with the following properties: (1)  $\cup_{B \in V(T)} B = V$ ; (2) for every edge  $(u, v)$  of  $G$ , there is a bag of  $T$  containing  $u$  and  $v$ ; and (3) for every  $u \in V$ , the set of bags containing  $u$  induces a subtree of  $T$ .

**Definition 5.** A cop-decomposition of width  $k$  for graph  $G$  is a rooted tree  $T$  that is a tree-decomposition for  $G$  satisfying the following property. For every bag  $B$  of  $T$ , the set of vertices of  $B \setminus B'$ , where  $B'$  is the parent<sup>1</sup> bag of  $B$ , is composed of at most  $k$  shortest paths of  $G \setminus B'$ .

Note that the core difference between the width of a cop-decomposition and the width of a tree-decomposition is that we count the number of shortest paths instead of the number of vertices in each bag. The *cop-width* of  $G$ , denoted by  $cw(G)$ , is the least number  $k$  such that  $G$  admits a cop-decomposition of width  $k$ . Observe that trees have cop-width 1. If  $G$  excludes a  $K_r$  as a minor,

<sup>1</sup>If  $B$  is the root bag of  $T$ , then we set  $B' = \emptyset$ .

then Andreae shows that  $cw(G) \leq r - 1$ . In fact, Andreae’s cop algorithm constructively creates a cop-decomposition for  $G$  of width  $r - 1$ , moreover, each bag  $B$  is actually a rooted shortest-path tree with at most  $r - 1$  leaves and whose root is in  $B \setminus B'$ .

**From Cop-Decompositions to Padded Partitions via Skeletons.** The cop-decomposition induces a partition of the vertices of the graph into bags that consist of at most  $r - 1$  shortest paths. Note that the number of vertices in each bag in a cop-decomposition may be large, and depend on  $n$ . Why are these bags useful? Since each bag  $B$  contains at most  $r - 1$  shortest paths in the induced subgraph  $B \setminus B'$  (where  $B'$  is the parent of  $B$ ), one can choose a “net” of centers along each path so that each node in the graph is threatened by  $O(r)$  centers from any one bag. Hence it now suffices to bound the number of bags that get close enough to a vertex  $z$  so that some centers from this bag may threaten  $z$ . (We call such a bag a “*threatening skeleton*” for  $z$ .) As mentioned above, we do not bound the worst-case number of such threatening skeletons; we prove it suffices to bound their expected number.

**Bounding the Expected Number of Threateners.** How to bound the expected number of threatening skeletons for some node  $z \in V$ ? We need a notion of progress. The cop-decomposition ensures that in any given moment there are less than  $r$  bags (a.k.a. threatening skeletons) that  $z$  can see on the boundaries of its component, where each bag consists of a tree with at most  $r - 1$  paths. We observe the following property of the distances from  $z$  to these trees: if constructing a new tree  $T_{new}$  in the induced subgraph containing  $z$  causes some current tree  $T_{curr}$  to become farther from  $z$  (or even to be disconnected from  $z$ ) because it cuts off some short path from  $z$  to  $T_{curr}$ , the distance from  $z$  to  $T_{new}$  is strictly less than the distance from  $z$  to  $T_{curr}$ . Indeed, if this distance were to miraculously decrease (deterministically) by  $\Delta/k$  then one can prove a bound of  $O\binom{r+k}{k}$  on the number of threateners. But why should such a large decrease happen? It doesn’t, but *we force this to happen in expectation*. We change the above construction and build a “buffer” of some random radius around each skeleton we build. Note that the supernodes did not have to be trees in the above arguments, and hence “fattening” them by growing buffers around the trees would not change any of the preceding arguments. Now by choosing the buffer radius from a truncated exponential with rate  $O(r)$ , we may naïvely hope to decrease the distance by  $\Delta/r$  with constant probability (assuming no truncation). The proof is much more subtle, and requires to overcome the truncation of the buffer. We use a potential function with delicately chosen parameters, such that for each new tree, this potential increases in expectation by  $\approx r/2^r$ . The potential starts at 0 and once it reaches  $r$ , it means that  $z$  is at distance 0 from some buffered tree and will not be threatened again. Finally, the optional stopping theorem helps us bound the expected number of threateners by  $\approx 2^r$ .

**Bounding Expected Increase in Potential.** In order to bound the number of threateners for  $z$ , the potential function we use is a sum of exponentials  $\sum_{\text{buffers } B} e^{-\alpha d(z, B)}$  for some parameter  $\alpha$ ; the sum is over those buffered trees that the node  $z$  can see. The main challenge is that in the worst case, one new buffered tree can cause all the other current buffered trees to be disconnected from the component containing  $z$ , hence losing  $r$  summands of the potential. To overcome this we need to guarantee that the expected gain from the new tree is  $O(r)$  times more than the expected loss of any single current tree, which is one of the technical cores of the analysis. We note that

obtaining any deterministic bound on the number of threateners using a cop-decomposition, rather than only bounding the expectation, remains an open question.

## 1.2 Other Related Work

The ideas of either finding a “good” decomposition or else building a  $K_r$ -minor used by [KPR93, And86] also appear in “shallow-minor theorems” of Alon, Seymour, and Thomas [AST90], Plotkin, Rao, and Smith [PRS94], and others. The parameters and run-times of these constructions have been considerably improved, see the paper of Wulff-Nilsen [WN11] and the references therein.

Busch, LaFortune, and Tirthapura [BLT07] first suggested the idea of decomposing a graph into paths and building balls around these paths; they considered this in the context of strong-diameter covers. They give the best constants for covers of planar graphs; for  $K_r$ -minor-free graphs, they give  $O(1)$ -padding and  $O(\log |V| \cdot f(r))$ -overlap, where  $f(r)$  depends on the Robertson-Seymour structure theorem.

In contrast to the weak-diameter partitions of Klein et. al. and Fakcharoenphol and Talwar [KPR93, FT03], the previously best *strong-diameter* partitions are due to Abraham et. al. [AGMW10], who guarantee strong-diameter  $\Delta$  and probability of an edge  $\{u, v\}$  being separated is  $O(6^r r^2 \cdot \frac{d(u,v)}{\Delta})$ . Abraham et. al. [AGMW10] also present *sparse covers* with strong-diameter  $\Delta$ , padding of  $O(r^2)$  and overlap of  $2^{O(r)} r!$ .

The papers [IS07, BLS10, Sid10] give algorithms to probabilistically embed genus- $g$  graphs into planar graphs with  $2^{O(g)}$ ,  $O(g^2)$  and  $O(\log g)$  distortion respectively. The ideas developed in this line of work lead to an asymptotically optimal padding parameter of  $O(\log g)$  for genus- $g$  graphs [LS10].

For general graphs, the decomposition schemes in, e.g., [Awe85, LS93, Bar96, CKR05, FRT04] give asymptotically optimal  $O(\log |V|)$  padding. The best result known for tree-width- $r$  graphs was the same as for  $K_r$ -minor-free graphs, i.e.,  $O(r^2)$ -padding partitions.

## 1.3 Organization of the Paper

After a few preliminary definitions, we provide in Section 3 a bound on the expected number of threateners for a wide range of partition algorithms, and show how to use this to bound the padding probability. Our main result Theorem 3 is proved in Section 4. The three assertions of Theorem 4 are then proven in Sections 5, 6 and 7.

## 2 Definitions and Notation

**Graphs.** We assume familiarity with graph-theoretic notions; see, e.g., [Die00] for background. Here are some definitions we will use. Given a graph  $G = (V, E)$ , a ball around  $A \subseteq V$  of radius  $t \geq 0$  is  $B_G(A, t) = \{u \in V \mid d_G(A, u) \leq t\}$ . Also let  $N(A) = \{u \in V \mid \exists v \in A, \{u, v\} \in E\}$ . For subsets  $A, B \subseteq V$  define a relation  $\sim$  where  $A \sim B$  iff  $A \cap N(B) \neq \emptyset$ , that is, iff there is an edge between a vertex of  $A$  to a vertex of  $B$ .

A *minor* of  $G$  is a subgraph of a graph obtained from  $G$  by a sequence of edge contractions. Equivalently,  $G'$  is a minor of  $G$  if there exists a map  $f : V(G) \rightarrow V(G')$  such that (a) for each  $u' \in V(G')$  the “supernode”  $f^{-1}(u')$  is connected in  $G$ , and (b) for every edge  $\{u', v'\} \in E(G')$ , there is at least one edge between  $f^{-1}(u')$  and  $f^{-1}(v')$  in  $E(G)$ . A graph  $G$  is  *$H$ -minor-free* (or *excludes an  $H$ -minor*) if  $G$  does not contain a minor isomorphic to  $H$ . As is well-known, planar graphs are

exactly the graphs excluding  $K_{3,3}$  and  $K_5$  as minors. In fact, Robertson and Seymour proved that every graph family closed under taking minors is characterized by a finite set of excluded minors.

Many one-way implications are also known: if we can show that a class  $\mathcal{G}$  of graphs is closed under taking minors, and  $H \notin \mathcal{G}$ , then  $\mathcal{G}$  contains only  $H$ -minor-free-graphs. Hence, graphs with tree-width at most  $r$  are  $K_{r+2}$ -minor-free (since tree-width of a clique is one smaller than its size, and the tree-width of a graph does not increase under taking minors); graphs with genus  $g$  exclude  $K_r$  as a minor for some  $r = \Theta(\sqrt{g})$ , since the genus of  $K_r$  is  $\Theta(r^2)$ .

**Truncated Exponential Distributions.** We will extensively use the following probability distribution over positive reals. The  $[\theta_1, \theta_2]$ -truncated exponential distribution with parameter  $b$  is denoted by  $\text{Texp}_{[\theta_1, \theta_2]}(b)$ , and the density function is:

$$f_{\text{texp}; b; \theta_1, \theta_2}(y) := \frac{b e^{-b \cdot y}}{e^{-b \cdot \theta_1} - e^{-b \cdot \theta_2}} \quad \text{for } y \in [\theta_1, \theta_2]. \quad (2.1)$$

For the  $[0, 1]$ -truncated exponential distribution we drop the subscripts and denote it by  $\text{Texp}(b)$ ; the density function is

$$f_{\text{texp}; b}(y) := \frac{b e^{-b \cdot y}}{1 - e^{-b}} \quad \text{for } y \in [0, 1]. \quad (2.2)$$

Note that if  $Y \propto \text{Texp}(b)$  then  $u \cdot Y \propto \text{Texp}_{[0, u]}(b/u)$ .

### 3 Analysis

Our algorithms induce an iterative process that creates “skeletons” (e.g., trees, paths, or vertices) and remove their neighborhoods (a buffer), defined according to some truncated exponential distribution, from the graph. Once we have these skeletons, our algorithms define a second iterative process that creates clusters from the skeletons.

Let us abstract out the properties needed from our first and second processes.

**Definition 6** (Skeleton-Process). *Given a graph  $G$ , parameters  $0 \leq l < u \leq 1$  and  $b > 0$ , any process which generates a sequence of graphs  $G = G_0, G_1, \dots$ , skeletons  $A_0, A_1, \dots$  and vertex sets  $K_0, K_1, \dots$ , that satisfies the following property is a skeleton-process:*

- For any  $i \geq 0$ ,  $A_i \subseteq V(G_i)$  and  $K_i = B_{G_i}(A_i, R_i \Delta)$ , where  $R_i \propto \text{Texp}_{[l, u]}(b/(u-l))$ .

*The process is threatening if the graph sequence satisfies  $G_{i+1} = G_i \setminus K_i$ , and the process is cutting if the graph sequence satisfies  $G_{i+1} \supseteq G_0 \setminus (\cup_{j \leq i} K_j)$ .*

The first process is a threatening process which creates buffers around the trees of the cop-decomposition. The second process is a cutting process that creates the actual clusters centered at net-points of the trees. For the strong-diameter results, we will have a single process that satisfies both definitions.

#### 3.1 Analysis of the Threatening Process: Bounding the Expected Threats

A crucial property of all of our algorithms is that any vertex  $z$  can “see” at most  $s$  buffers (the  $K_i$  sets) at any time, for some parameter  $s$  (in the weak-diameter partition we will have  $s = r$ ). By this

we mean that for any connected component  $C$  in one of the remaining graphs (after some buffers were removed), there are at most  $s$  buffers that are connected to  $C$  by an edge of  $G$ . This property will enable us to prove that  $z$  is expected to be “threatened” by a small number of skeletons, that is, we expect a few skeletons that are sufficiently close to cut a certain ball around  $z$ .

Consider a threatening skeleton-process with parameters  $l = 0$ ,  $u \in (0, 1]$  and  $b = 2s$ . We prove a bound on the expected number of threateners for a ball around any vertex  $z$  of  $G$  with padding parameter  $\gamma > 0$ . For some  $u \leq u' \leq 1$ , let  $\mathcal{J}_z = \{A_i \mid d_{G_i}(z, A_i) \leq (u' + \gamma)\Delta\}$  be the set of vertex sets whose subset  $K_i$  may intersect  $B_z = B_G(z, \gamma\Delta)$ . Observe that once  $z \in K_j$  for some index  $j$  then it is removed from the graph, and  $\mathcal{J}_z$  cannot increase anymore. For a connected component  $C_i \in G_i$  let  $\mathcal{K}_{|C_i} = \{K_j \mid j < i \wedge C_i \sim K_j\}$ . (Recall that  $A \sim B$  if there exists an edge from a node in  $A$  to some node in  $B$ .)

**Lemma 7.** *Suppose that in a threatening skeleton-process we have the property that for every  $i \in \mathbb{N}$  and every connected component  $C_i \in G_i$ , we are guaranteed that  $|\mathcal{K}_{|C_i}| \leq s$ , then*

$$\mathbb{E}[|\mathcal{J}_z|] \leq 3e^{(2s+1) \cdot (u'+\gamma)/u} .$$

*Proof.* Fix any  $i \in \mathbb{N}$ . W.l.o.g., we may assume that the process always picks the set  $A_i$  in the connected component  $C_i$  of  $G_i$  that contains  $z$  (the other components do not affect  $\mathcal{J}_z$ ). Let  $\mathbf{x} = \mathbf{x}(i)$  be a vector of the “normalized distances” from  $z$  to  $\mathcal{K}_{|C_i}$ . More formally, if  $\mathcal{K}_{|C_i} = \{K_{i_1}, \dots, K_{i_l}\}$  (with  $l \leq s$  by the assumption of the lemma), then for  $j \in [l]$  define

$$x_j := \frac{d_{G_i \cup K_{i_j}}(z, A_{i_j}) - R_{i_j}\Delta}{u\Delta} .$$

Intuitively,  $x_j$  should have been the distance from  $z$  to  $K_{i_j}$ , normalized by  $u\Delta$ . Note that by the definition of  $K_{i_j}$  we have that  $d_{G_i \cup K_{i_j}}(z, K_{i_j}) \geq d_{G_i \cup K_{i_j}}(z, A_{i_j}) - R_{i_j}\Delta$ .

Define the potential function for the vector  $\mathbf{x} := (x_1, \dots, x_l)$  as

$$\Phi(\mathbf{x}) = \sum_{j=1}^l e^{-(2s+1) \cdot x_j} . \tag{3.3}$$

We would like to analyze the change to  $\mathbf{x}$  over time. Assume w.l.o.g that  $x_1 \leq \dots \leq x_l$ . Let  $h := \frac{d_{G_i}(z, A_i)}{u\Delta} \geq 0$  be the normalized distance of  $z$  from the set  $A_i$ , and let  $y = h - R_i/u$ . Observe that if  $x_j \leq y$  then the shortest path from  $K_{i_j}$  to  $z$  is completely disjoint from  $K_i$ ; seeking contradiction, assume  $a \in K_i$  lies on the shortest path in  $G_i$  from  $z$  to  $K_{i_j}$ . Since every vertex of distance  $R_{i_j}\Delta$  from  $A_{i_j}$  is in  $K_{i_j}$  and thus was removed from the graph, it must be that  $d_{G_i \cup K_{i_j}}(a, A_{i_j}) > R_{i_j}\Delta$ . We conclude that

$$d_{G_i}(z, a) < d_{G_i \cup K_{i_j}}(z, A_{i_j}) - R_{i_j}\Delta = x_j \cdot (u\Delta) \leq y \cdot (u\Delta) = d_{G_i}(z, A_i) - R_i\Delta \leq d_{G_i}(z, K_i) \leq d_{G_i}(z, a) ,$$

contradiction.

We get that if  $j^*$  is the maximal index such that  $x_{j^*} \leq y$ , then the first  $j^*$  entries of  $\mathbf{x}$  will not change. The new set  $K_i$  will always be in  $\mathcal{K}_{|C_{i+1}}$  (recall that  $C_{i+1}$  is the component containing  $z$  in  $G_{i+1} = G_i \setminus K_i$ ), so we have that the  $j^* + 1$  entry in  $\mathbf{x}(i+1)$  will be  $x_{j^*+1} = y$ . For  $j^* < j \leq l$ , it could be the case that  $K_i$  intersects the shortest path from  $K_{i_j}$  to  $z$ , in which case the distance may increase or  $K_{i_j}$  can even be disconnected from  $z$ . Note that if  $l = s$ , then it must be that at least one  $K_{i_j}$  is disconnected from  $z$ , because we assume that  $|\mathcal{K}_{|C_{i+1}}| \leq s$ .



Next we attempt to bound the expected change to the potential function  $\Phi$  in any single step. To this end, it suffices to consider the worst scenario, in which all the  $K_{i_j}$  for  $j^* < j \leq l$  become disconnected from  $z$  by  $K_i$  (in such a case the potential decreases the most). Define the “filtered subsequence”  $\mathbf{x} \downarrow y$  to be the sequence obtained by dropping all the coordinates of  $\mathbf{x}$  which are strictly larger than  $y$ , and adding in  $y$ . (E.g.,  $(-0.4, -0.3, 0.7, 5, 6.9) \downarrow 1.42 = (-0.4, -0.3, 0.7, 1.42)$ .)

**Lemma 8.** *Let  $\Phi$  be a function defined as in (3.3). Fix any non-decreasing sequence  $\mathbf{x}$  of length at most  $s$ , and a random variable  $Y \propto \text{Texp}(2s)$  as in (2.2). Then for any  $h \geq 0$ :*

$$\mathbb{E}_Y[\Phi(\mathbf{x} \downarrow (h - Y)) - \Phi(\mathbf{x})] \geq \frac{s(e - 2) \cdot e^{-(2s+1)h}}{1 - e^{-2s}}.$$

*Proof of Lemma 8.* Let  $b = 2s$  and  $a = -(b + 1)$ . The increase of the potential due to the new coordinate  $y = h - Y$  is  $e^{a \cdot (h - Y)}$ , so the expected gain is

$$\mathbb{E}[e^{a(h - Y)}] = e^{ah} \cdot \int_0^1 e^{-aw} f_{\text{texp};b}(w) dw = e^{ah} \cdot \int_0^1 \frac{b}{1 - e^{-b}} e^{-(a+b)w} dw = \frac{b(e - 1) \cdot e^{ah}}{1 - e^{-b}}. \quad (3.4)$$

Next we analyze the loss in  $\Phi$  for the coordinates  $x_j$  that are dropped. Recall that a coordinate  $x_j$  is dropped exactly when  $x_j > h - Y$ . Since  $Y \in [0, 1]$  the only interesting case is when  $x_j = h - \gamma$  for some  $\gamma \in [0, 1]$ , which is dropped when  $Y > \gamma$ , so the maximum loss is

$$\begin{aligned} \max_{\gamma \in [0, 1]} e^{a(h - \gamma)} \Pr[Y > \gamma] &= e^{ah} \cdot \max_{\gamma \in [0, 1]} e^{-a\gamma} \int_{\gamma}^1 f_{\text{texp};b}(w) dw \\ &= e^{ah} \cdot \max_{\gamma \in [0, 1]} e^{-a\gamma} \frac{e^{-b\gamma} - e^{-b}}{1 - e^{-b}} \\ &= \frac{e^{ah}}{1 - e^{-b}} \cdot \max_{\gamma \in [0, 1]} e^{\gamma} - e^{-a\gamma - b} \\ &\leq \frac{e^{ah}}{1 - e^{-b}} \cdot e. \end{aligned}$$

Since  $\mathbf{x}$  has only  $s$  coordinates, the total expected loss incurred on  $\Phi$  is thus at most

$$\frac{s \cdot e \cdot e^{ah}}{1 - e^{-b}}. \quad (3.5)$$

Finally using (3.4) and (3.5) we see that

$$\begin{aligned} \mathbb{E}[\Phi(\mathbf{x} \downarrow (h - Y)) - \Phi(\mathbf{x})] &\geq \frac{b(e - 1) \cdot e^{ah}}{1 - e^{-b}} - \frac{s \cdot e^{ah+1}}{1 - e^{-b}} \\ &= \frac{(e(b - s) - b) \cdot e^{-(b+1)h}}{1 - e^{-b}}. \end{aligned}$$

□

Applying Lemma 8 with the vector  $\mathbf{x}$  (whose length is indeed at most  $s$ ) and  $Y = R_i/u$ , noting that  $Y \propto \text{Texp}(2s)$  by a property of the exponential distribution mentioned in Section 2, we get that

$$\mathbb{E}_Y[\Phi(\mathbf{x} \downarrow y) - \Phi(\mathbf{x})] \geq \frac{(e-2)s \cdot e^{-(2s+1)h}}{1 - e^{-2s}} \geq 2s \cdot e^{-(2s+1)h}/3. \quad (3.6)$$

It will be more convenient to analyze a slightly different potential function

$$\Phi'(\mathbf{x}) = \begin{cases} 2s & \exists j, x_j \leq 0 \\ \Phi(\mathbf{x}) & \text{otherwise} \end{cases}$$

We claim that (3.6) still holds for  $\Phi'$  (as long as  $z \in G_i$ ). This is because it is the same as  $\Phi$  as long as  $\mathbf{x} > 0$ , and if  $\mathbf{x}(i+1)$  is the first to have a non-positive coordinate, then  $\Phi'(\mathbf{x}(i)) = \Phi(\mathbf{x}(i)) < s$  and  $\Phi'(\mathbf{x}(i+1)) = 2s$ , but since  $h \geq 0$  the claimed expected increase in (3.6) is never more than  $s$ .

Denote by  $\Phi_i = \Phi'(\mathbf{x}(i))$ . Recall that for every  $A_i \in \mathcal{J}_z$  we have that  $d_{G_i}(z, A_i) \leq (u' + \gamma)\Delta$  and thus  $h = d_{G_i}(z, A_i)/(u\Delta) \leq (u' + \gamma)/u$ . Observe that the expectation of (3.6) is taken only over the current choice of  $Y$ , and since  $Y$  is chosen independently we can condition on any other event that depends on previous steps, and obtain the same bound. In particular, for  $A_i \in \mathcal{J}_z$ ,

$$\mathbb{E}[\Phi_{i+1} \mid \Phi_i] \geq \Phi_i + 2s \cdot e^{-(2s+1)h}/3 \geq \Phi_i + 2s \cdot e^{-(2s+1) \cdot (u'+\gamma)/u}/3, \quad (3.7)$$

and define  $\zeta = 2s \cdot e^{-(2s+1) \cdot (u'+\gamma)/u}/3$ . Also note that the bound of (3.6) is always positive, so even if  $A_i \notin \mathcal{J}_z$  we still have

$$\mathbb{E}[\Phi_{i+1} \mid \Phi_i] \geq \Phi_i. \quad (3.8)$$

For  $t \in \mathbb{N}$  let  $j_t = |\{i : A_i \in \mathcal{J}_z \text{ and } i \leq t\}|$  be the number of time steps until  $t$  in which  $z$  is threatened. We claim that the process  $X_0, X_1, \dots$  where  $X_t = \Phi_t - \zeta \cdot j_t$ , is a submartingale. To prove this consider two cases: If  $A_{t+1} \in \mathcal{J}_z$  then  $j_{t+1} = j_t + 1$ , and by (3.7) we get  $\mathbb{E}[\Phi_{t+1} \mid \Phi_t, j_t] \geq \Phi_t + \zeta$ , and so

$$\mathbb{E}[X_{t+1} \mid X_1, \dots, X_t] = \mathbb{E}[\Phi_{t+1} - \zeta \cdot j_{t+1} \mid \Phi_t, j_t] \geq \Phi_t + \zeta - \zeta \cdot j_{t+1} = \Phi_t + \zeta - \zeta \cdot (j_t + 1) = X_t.$$

If it is the case that  $A_{t+1} \notin \mathcal{J}_z$ , then  $j_{t+1} = j_t$  and

$$\mathbb{E}[\Phi_{t+1} - \zeta \cdot j_{t+1} \mid \Phi_t, j_t] \geq \Phi_t - \zeta \cdot j_t = X_t.$$

The stopping time of a (sub)martingale  $X_0, X_1, \dots$  is a random variable  $\tau$  that has support in  $\mathbb{N}$ , and such that the event  $\tau = t$  depends only on  $X_0, \dots, X_t$ . Define  $\tau$  as the first time in which  $\Phi_\tau = 2s$ . Observe that if  $t$  is the time where  $z \in K_t$ , then it must be that  $d_{G_t}(z, A_t) \leq R_t\Delta$ , and so we get a non-positive coordinate in  $\mathbf{x}(t)$  which implies that  $\tau = t$ . Since the stopping time is bounded by  $|V|$  (there can be at most  $|V|$  rounds, because at least one vertex is removed every round), we can apply Doob's optional stopping time Theorem [GS01, Section 12.5] and obtain that

$$\mathbb{E}[\Phi_\tau] - \zeta \cdot \mathbb{E}[j_\tau] = \mathbb{E}[X_\tau] \geq \mathbb{E}[X_0] = 0.$$

Finally, as  $\Phi_\tau = 2s$ , we obtain that

$$\mathbb{E}[|\mathcal{J}_z|] = \mathbb{E}[j_\tau] \leq 2s/\zeta = 3e^{(2s+1) \cdot (u'+\gamma)/u}.$$

This completes the proof of Lemma 7. □

### 3.2 Analysis of the Cutting Process: Bounding the Probability of Cutting a Ball

In this section we give a bound on the probability that a ball is cut by a cutting skeleton-process, which depends on the *expected* number of threateners.

Consider a cutting skeleton-process as in [Definition 6](#) with parameters  $0 \leq l < u \leq 1$ ,  $b > 0$ . Fix  $z \in V(G)$ , a parameter  $\gamma > 0$  and set  $B_z = B_G(z, \gamma\Delta)$ . Let  $\mathcal{T}_z = \{A_i \mid d_{G_i}(z, A_i) \leq (u + \gamma)\Delta\}$  be the set of vertex sets whose subset  $K_i$  may intersect  $B_z$ . Let  $N := |\mathcal{T}_z|$  be a random variable with  $\tau = \mathbb{E}[N]$ . We say that  $B_z$  is *cut* by the skeleton-process if it intersects more than a single  $K_i$ .

**Lemma 9.** *For  $\delta = e^{-2b\gamma/(u-l)}$ , the probability that  $B_z = B_G(z, \gamma\Delta)$  is cut by a cutting skeleton-process with the property that  $\tau = \mathbb{E}[|\mathcal{T}_z|]$ , is at most*

$$(1 - \delta) \left( 1 + \frac{\tau}{e^b - 1} \right).$$

Let us introduce some more notation and properties before proving this Lemma. Define the following events:

$$\begin{aligned} \mathcal{C}_i &= \{B_z \cap K_i \notin \{\emptyset, B_z\}\} && \text{“}B_z \text{ cut in round } i \text{ (by } A_i\text{)”}, \\ \mathcal{F}_i &= \{B_z \cap K_i = \emptyset\} && \text{“}i \text{ was a no-op round”}, \\ \mathcal{E}_i &= \left\{ \mathcal{C}_i \wedge \bigwedge_{j < i} \mathcal{F}_j \right\} && \text{“}B_z \text{ first cut in round } i\text{”}. \end{aligned}$$

Denote by  $\mathcal{F}_{(<i)}$  the event  $\bigwedge_{j < i} \mathcal{F}_j$ , so that  $\mathcal{E}_i = (\mathcal{C}_i \wedge \mathcal{F}_{(<i)})$ . Denote by  $\overline{\mathcal{F}}_i$  the complement of  $\mathcal{F}_i$ . Observe that  $\mathcal{C}_i$  (respectively  $\overline{\mathcal{F}}_i$ ) implies that  $A_i \in \mathcal{T}_z$ , so

$$\Pr[\mathcal{C}_i] = \Pr[\mathcal{C}_i \wedge A_i \in \mathcal{T}_z] = \Pr[A_i \in \mathcal{T}_z] \cdot \Pr[\mathcal{C}_i \mid A_i \in \mathcal{T}_z], \quad (3.9)$$

$$\Pr[\overline{\mathcal{F}}_i] = \Pr[A_i \in \mathcal{T}_z] \cdot \Pr[\overline{\mathcal{F}}_i \mid A_i \in \mathcal{T}_z]. \quad (3.10)$$

and the same holds also when conditioning on any other event. We have the following claim:

**Claim 10.** *For each  $i \in \mathbb{N}$ ,*

$$\Pr[\mathcal{C}_i \mid \mathcal{F}_{(<i)}, A_i \in \mathcal{T}_z] \leq (1 - \delta) \cdot \left( \Pr[\overline{\mathcal{F}}_i \mid \mathcal{F}_{(<i)}, A_i \in \mathcal{T}_z] + \frac{1}{e^b - 1} \right).$$

*Proof.* Fix any graph  $G_i$  and any set  $A_i \subseteq V(G_i)$  that agree with the conditioning on  $\mathcal{F}_0, \dots, \mathcal{F}_{i-1}$  and so that  $A_i \in \mathcal{T}_z$ . Denote by  $\rho = d_{G_i}(A_i, B_z)$ ,  $\bar{b} = b/(u-l)$ , and let  $m = \max\{l, \rho\}$ . Recall that  $R_i$  is chosen independently, so

$$\begin{aligned} \Pr[\overline{\mathcal{F}}_i \mid \mathcal{F}_0, \dots, \mathcal{F}_{i-1}, A_i \in \mathcal{T}_z, A_i] &= \int_m^u \frac{\bar{b}e^{-\bar{b}y}}{e^{-\bar{b}l} - e^{-\bar{b}u}} dy \\ &= \frac{e^{-\bar{b}m} - e^{-\bar{b}u}}{e^{-\bar{b}l} - e^{-\bar{b}u}}. \end{aligned}$$

Since  $\mathcal{F}_0, \dots, \mathcal{F}_{i-1}$  occurred and  $G_i \supseteq G_0 \setminus (\cup_{j < i} K_j)$ , we have that  $B_z \subseteq G_i$ . Now if  $R_i \geq \rho + 2\gamma$  then by the triangle inequality  $B_z \subseteq K_i$ , and the ball is “saved”. This bounds the cut probability

thus:

$$\begin{aligned}
\Pr[\mathcal{C}_i \mid \mathcal{F}_{(<i)}, A_i \in \mathcal{T}_z, A_i] &\leq \int_m^{\rho+2\gamma} \frac{\bar{b}e^{-\bar{b}y}}{e^{-\bar{b}l} - e^{-\bar{b}u}} dy \\
&\leq \frac{e^{-\bar{b}m} - e^{-\bar{b}(m+2\gamma)}}{e^{-\bar{b}l} - e^{-\bar{b}u}} \\
&= \frac{e^{-\bar{b}m}(1-\delta)}{e^{-\bar{b}l} - e^{-\bar{b}u}} \\
&= (1-\delta) \cdot \Pr[\bar{\mathcal{F}}_i \mid \mathcal{F}_{(<i)}, A_i \in \mathcal{T}_z, A_i] + (1-\delta) \frac{e^{-\bar{b}u}}{e^{-\bar{b}l} - e^{-\bar{b}u}} \\
&= (1-\delta) \cdot \left( \Pr[\bar{\mathcal{F}}_i \mid \mathcal{F}_{(<i)}, A_i \in \mathcal{T}_z, A_i] + \frac{1}{e^b - 1} \right).
\end{aligned}$$

Finally, because the bound holds for any  $A_i$ , it holds without conditioning on it.  $\square$

*Proof of Lemma 9.* Observe that for each  $i \in [N]$ , the events  $\{\bar{\mathcal{F}}_i \wedge \mathcal{F}_{(<i)}\}$  are pairwise disjoint (this is the event that  $B_z$  is either cut or contained in  $K_i$  for the first time), thus by the law of total probability,

$$\sum_{i \in \mathbb{N}} \Pr[\bar{\mathcal{F}}_i \wedge \mathcal{F}_{(<i)}] \leq 1. \quad (3.11)$$

Also, by linearity of expectation

$$\tau = \sum_{i \in \mathbb{N}} \Pr[A_i \in \mathcal{T}_z]. \quad (3.12)$$

To bound the probability of the ball being cut, we start off with the trivial union bound:

$$\begin{aligned}
\Pr\left[\bigcup_{i \in \mathbb{N}} \mathcal{E}_i\right] &\leq \sum_i \Pr[\mathcal{E}_i] = \sum_i \Pr[\mathcal{C}_i \wedge \mathcal{F}_{(<i)}] \\
&= \sum_i \Pr[\mathcal{C}_i \mid \mathcal{F}_{(<i)}] \cdot \Pr[\mathcal{F}_{(<i)}] \\
&\stackrel{(3.9)}{=} \sum_i \Pr[\mathcal{C}_i \mid \mathcal{F}_{(<i)}, A_i \in \mathcal{T}_z] \cdot \Pr[A_i \in \mathcal{T}_z \mid \mathcal{F}_{(<i)}] \cdot \Pr[\mathcal{F}_{(<i)}] \\
&\stackrel{\text{Claim 10}}{\leq} \sum_i (1-\delta) \left( \Pr[\bar{\mathcal{F}}_i \mid \mathcal{F}_{(<i)}, A_i \in \mathcal{T}_z] + \frac{1}{e^b - 1} \right) \cdot \Pr[A_i \in \mathcal{T}_z \mid \mathcal{F}_{(<i)}] \cdot \Pr[\mathcal{F}_{(<i)}] \\
&\stackrel{(3.10)}{=} (1-\delta) \cdot \sum_i \Pr[\bar{\mathcal{F}}_i \wedge \mathcal{F}_{(<i)}] + \sum_i \Pr[A_i \in \mathcal{T}_z \wedge \mathcal{F}_{(<i)}] \cdot \frac{1-\delta}{e^b - 1} \\
&\stackrel{(3.11)}{\leq} (1-\delta) + \frac{1-\delta}{e^b - 1} \cdot \sum_i \Pr[A_i \in \mathcal{T}_z] \\
&\stackrel{(3.12)}{=} (1-\delta) \left( 1 + \frac{\tau}{e^b - 1} \right).
\end{aligned}$$

This completes the proof.  $\square$

## 4 A Weak-Diameter Partition

In this section, we show how to construct a weak-diameter partition for  $K_{r+1}$ -minor-free graphs which is  $O(r)$ -padded (with constant  $\delta = 1/160$ ). The ideas here will later extend to the case of strong-diameter partitions with a weaker  $(O(r^2), O(1/r^2))$ -padding.

### 4.1 The Algorithm

At a high level, the algorithm works as follows: in each step, pick a connected component of the remaining graph, and find (in a specific way) a shortest-path tree  $T$  in this component. Delete a random neighborhood of  $T$  from the graph, and recurse on each connected component of the graph, if any. We then construct a net of points on each tree, and from these net points grow “balls” of random radius to form the small-diameter regions of the partition. A key property to ensure the padding guarantee is that each vertex is expected to be close to few of these paths. We show that this property holds, otherwise we can construct a  $K_{r+1}$ -minor in  $G$ .

More specifically, the algorithm maintains a set of trees  $T_i$  and supernodes  $S_i$  that will be used in the construction, each tree and supernode have a “center” vertex associated with them. Let us describe a generic  $i$ -th iteration of the algorithm. Let  $\mathcal{S}$  be the set containing all the supernodes created so far, initially this will be empty. Let  $C$  be a connected component in the graph  $G_i = G \setminus (\cup \mathcal{S})$ , where  $\cup \mathcal{S}$  is the set of all vertices lying in the supernodes in  $\mathcal{S}$ , initially  $G \setminus (\cup \mathcal{S})$  will be the entire graph. Let  $\mathcal{S}_{|C} = \{S \in \mathcal{S} : S \sim C\}$  be the set of supernodes that have a neighbor in component  $C$ . Say  $\mathcal{S}_{|C} = \{S'_1, S'_2, \dots, S'_k\}$ , and consider the vertices  $F_j = N(S'_j) \cap C$  for each supernode, which are vertices in  $C$  neighbors of these “adjacent” supernodes. (These  $F_j$ ’s may intersect.) We pick an arbitrary vertex  $u_i$  from  $C$  and build a tree  $T_i$  rooted at  $u_i$ , which is comprised of shortest paths from  $u_i$  to each of the sets  $F_j$  (that is, for each  $j$  take a shortest path from  $u_i$  to the nearest vertex in  $F_j$ ). Define the next supernode

$$S_i := B_{G_i}(T_i, R_i \Delta),$$

where  $R_i \propto \text{Texp}_{[0,1/8]}(16r)$ . (Recall the definition of the truncated exponential distribution from (2.1).)

In order to create the random partition, choose a  $\Delta/8$ -net  $N_i$  over  $T_i$ , and enumerate  $N_i = \{v_1, \dots, v_{|N_i|}\}$ . For each  $1 \leq j \leq |N_i|$ , create a cluster  $B_{G_i}(v_j, \alpha_j \Delta) \cap U_j$  (where  $U_j$  is the set of points which have no cluster yet), where each  $\alpha_j \propto \text{Texp}_{[1/4,1/2]}(160r)$ . This completes the description of the algorithm; it is also given as Algorithm 1 and 2.

### 4.2 The Analysis

The following invariant holds for each time step  $i$ :

**Invariant 1.** *For every  $i \geq 0$ , every connected component  $C$  of  $G_i$  satisfies that if  $S, S' \in \mathcal{S}_{|C}$  then  $S \sim S'$ .*

*Proof.* The proof is by induction; the base case is trivial as there are no supernodes in  $\mathcal{S}_{|C}$ . Now by induction, assume that the invariant holds in  $G_i$ . Let  $T_i$  and  $S_i$  be the tree and supernode

---

**Algorithm 1** Weak-Random-Partition( $G, \Delta, r$ )

---

- 1: Let  $G_0 \leftarrow G$ ,  $i \leftarrow 0$ .
- 2: Let  $\mathcal{S} \leftarrow \emptyset$ .
- 3: Let  $\mathcal{T} \leftarrow \emptyset$ .
- 4: **while**  $G_i$  is non-empty **do**
- 5:   Let  $C_i$  be a connected component of  $G_i$ .
- 6:   Pick  $u_i \in C_i$ . Let  $T_i$  be a tree rooted at  $u_i$  that consists of shortest paths (in  $G_i$ ) from  $u_i$  to the closest vertex of  $N(S)$  for each supernode  $S \in \mathcal{S}_{C_i}$ .
- 7:   Let  $R_i$  be a random variable drawn independently from the distribution  $\text{Texp}_{[0,1/8]}(16r)$ .
- 8:   Let  $S_i \leftarrow B_{G_i}(T_i, R_i\Delta)$  be a neighborhood of  $T_i$ .
- 9:   Add  $S_i$  to  $\mathcal{S}$ .
- 10:   Add  $T_i$  to  $\mathcal{T}$ .
- 11:    $G_{i+1} \leftarrow G_i \setminus S_i$ .
- 12:    $i \leftarrow i + 1$ .
- 13: **end while**
- 14: **return** Create-Balls( $G, \mathcal{T}, \mathcal{S}, \Delta, r$ ).

---

---

**Algorithm 2** Create-Balls( $G, \mathcal{T}, \mathcal{S}, \Delta, r$ )

---

- 1:  $P = \emptyset$ .
- 2: **for**  $i = 1, \dots, |\mathcal{T}|$  **do**
- 3:   Let  $N_i = \{v_1, \dots, v_{|N_i|}\}$  be a  $\Delta/8$ -net of  $T_i$ .
- 4:   **for**  $j = 1, \dots, |N_i|$  **do**
- 5:     Let  $\alpha_j$  be a random variable drawn independently from distribution  $\text{Texp}_{[1/4,1/2]}(80r)$
- 6:     Add  $B_{G_i}(v_j, \alpha_j\Delta) \setminus \cup P$  as a cluster to the partition  $P$ .
- 7:   **end for**
- 8: **end for**
- 9: **return**  $P$ .

---

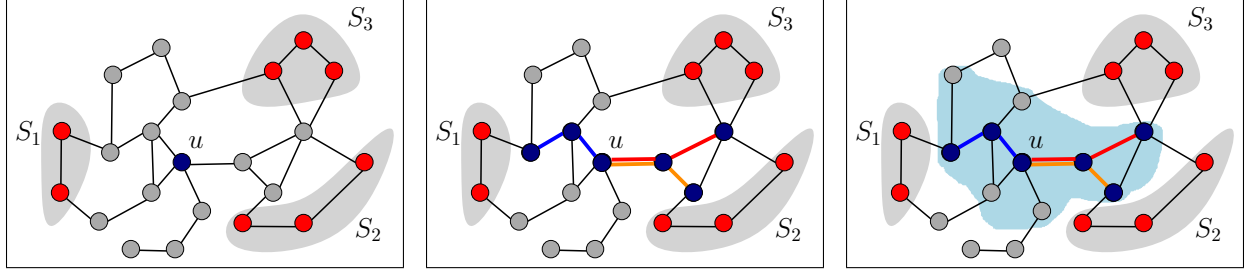


Figure 4.1: An iteration of the algorithm. On the left, there are three supernodes  $S_1, S_2, S_3$  neighboring the current component with  $u$  as a root. In the middle, we have a tree  $T_4$  comprised of three shortest paths from  $u$ . On the right, the new supernode  $S_4$  which is a 1-neighborhood of  $T_4$  (observe that this neighborhood is taken in the connected component containing  $u$ ).

constructed in step  $i$  in the component  $C_i$ . Let  $C$  be some connected component of  $G_{i+1}$ , and  $S, S' \in \mathcal{S}_{|C}$ . If  $C \cap C_i = \emptyset$  then  $C$  is a component of  $G_i$  as well; moreover, as  $S_i \subseteq C_i$  it must be that  $S_i \approx C$  so neither of  $S, S'$  can be  $S_i$ , and hence we can use the induction hypothesis to infer that  $S \sim S'$ . On the other hand, suppose that  $C \subseteq C_i$ . There are two cases: if  $S_i \notin \{S, S'\}$  we have  $S \sim S'$  by the induction hypothesis on  $C_i$ . On the other hand, suppose  $S_i = S$  (w.l.o.g.). Recall that  $T_i$  was chosen so that it contains a neighbor of every supernode in  $\mathcal{S}_{|C_i}$  and  $T_i \subseteq S_i$ , we have that  $S_i \sim S'$ .  $\square$

**Invariant 1** implies that for each connected component  $C$ , contracting the supernodes of  $\mathcal{S}_{|C}$  yields a  $K_{|\mathcal{S}_{|C}|}$  minor, so we obtain the following corollary.

**Corollary 11.** *If  $G$  excludes  $K_{r+1}$  as a minor, then for every time step  $i$ , the connected component  $C_i$  has  $|\mathcal{S}_{|C_i}| \leq r$ . In particular, the tree  $T_i$  is made up of at most  $r$  shortest paths in  $G_i$ .*

**Claim 12.** *The algorithm above generates a  $\Delta$ -bounded partition of  $G$ .*

*Proof.* First we prove that we generate a partition. Indeed, we delete supernodes from the graph, and recurse on the remaining components, so we need to show that vertices within the supernodes are contained in some cluster. Consider a vertex  $x$  in supernode  $S_i$ . By definition,  $d_{G_i}(x, T_i) \leq \Delta/8$ . Since  $N_i$  is a  $\Delta/8$ -net in  $T_i$ , some net point  $v_j \in N_i$  satisfies  $d_{G_i}(x, v_j) \leq \Delta/4$ . And since  $\alpha_j \geq 1/4$ , the ball  $B_{G_i}(v_j, \alpha_j \Delta)$  contains  $x$ . Hence each vertex within the deleted supernode is contained in some cluster, and we get a partition of  $G$ . Moreover, each cluster is a ball of radius at most  $\alpha_j \Delta \leq \Delta/2$  (and hence diameter at most  $\Delta$ ) in  $G_i$ . Finally, distances in  $G_i$  are no smaller than those in  $G$ .  $\square$

**Lemma 13.** *For  $r \geq 4$ , and any  $\gamma \leq 1/160$ , the probability that a ball  $B_z$  of radius  $\gamma \Delta$  is cut by the above process is*

$$\Pr[B_z \text{ cut}] \leq 1 - e^{-320r\gamma}.$$

*Proof.* First observe that the process defined in Algorithm 1 is a threatening skeleton-process, with the sequence of graphs  $G_0, G_1, \dots$  as defined in the algorithm and with  $A_i = T_i$ ,  $K_i = S_i$ ,  $l = 0, u = 1/8$ ,  $s = r$  and  $b = 2s$ . Recall that  $B_z = B_G(z, \gamma \Delta)$ , and set  $u' = 1/2$  so that  $\mathcal{J}_z = \{T_i \mid d_{G_i}(z, T_i) \leq (u' + \gamma)\Delta\}$  (we choose this  $u'$  to accommodate the cutting process which

will be conducted with this parameter). By [Invariant 1](#) we get that for all  $i \in \mathbb{N}$ ,  $|\mathcal{S}_{C_i}| \leq r$ , so by [Lemma 7](#) (using that  $\gamma \leq 1/160$ ),

$$\mathbb{E}[|\mathcal{J}_z|] \leq 3e^{(2r+1) \cdot (u'+\gamma)/u} \leq 10e^{10r} . \quad (4.13)$$

For each  $i$  such that  $T_i \in \mathcal{J}_z$ , let  $U_i = \{v \in N_i \mid d_{G_i}(v, z) \leq (1/2 + \gamma)\Delta\}$  be the net points in  $N_i$  that are sufficiently close to threaten  $B_z$  (note that by the choice of  $u'$  this is indeed the case), and denote  $\mathcal{T}_z = \cup_{i|T_i \in \mathcal{J}_z} U_i$ . By [Corollary 11](#),  $T_i$  is comprised of at most  $r$  shortest paths, and we claim that on each shortest path there can be at most 10 points that are in  $U_i$ . This is because the distance between any two consecutive net points on a path is at least  $\Delta/8$ , and if there are  $q > 10$  points, because this is a shortest path, the distance from the first point to the last is at least  $(q - 1) \cdot \Delta/8 > (1 + 2\gamma)\Delta$ . The triangle inequality implies that it can't be that both are within  $(1/2 + \gamma)\Delta$  from  $z$ . We conclude that for all  $i$  (with  $T_i \in \mathcal{J}_z$ ) we have  $|U_i| \leq 10r$ , thus by [\(4.13\)](#)

$$\tau := \mathbb{E}[|\mathcal{T}_z|] \leq 10r \cdot 10e^{10r} = 100r \cdot e^{10r} . \quad (4.14)$$

Next, we show that our **Create-Balls** algorithm generates a cutting skeleton-process. Simply take the sequence  $G_0, \dots, G_0, G_1, \dots, G_1, G_2, \dots$ , where each  $G_i$  is taken  $|N_i|$  times. Then the skeleton sets  $A$  are in fact singletons: for each  $i$  we will take  $|N_i|$  sets - the points of  $N_i$ , to be these singletons. The parameters for the exponential distribution are  $l = 1/4$ ,  $u = 1/2$  and  $b = 20r$ . To see the cutting property of [Definition 6](#), note that once we move from the graph  $G_i$  to  $G_{i+1}$ ,  $G_{i+1}$  will contain all the points yet uncovered by clusters, because we already observed in [Claim 12](#) that once all the points of  $N_i$  create a cluster, the supernode  $S_i$  is completely covered (recall  $G_{i+1} = G_i \setminus S_i$ ). Finally, applying [Lemma 9](#), we obtain that the probability that  $B_z$  is cut is at most

$$(1 - e^{-2b\gamma/(u-l)}) \left(1 + \frac{\tau}{e^b - 1}\right) = (1 - e^{-160r\gamma}) \left(1 + \frac{100r \cdot e^{10r}}{e^{20r} - 1}\right) .$$

It holds that  $\frac{100r \cdot e^{10r}}{e^{20r} - 1} \leq e^{-r}$  for  $r \geq 4$ , and this completes the proof as

$$(1 - e^{-160r\gamma}) \cdot (1 + e^{-r}) \leq (1 - e^{-160r\gamma}) \cdot (1 + e^{-160r\gamma}) = 1 - e^{-320r\gamma} ,$$

using that  $\gamma \leq 1/160$ . □

## 5 A Strong-Diameter Partition

In the previous section, we saw how to get a weak-diameter partition for minor-free graphs. In this section, we give a strong-diameter guarantee with a slightly weaker padding parameter of  $(O(r^2), O(1/r^2))$  instead of  $O(r)$ . However, this is still an exponential improvement over the best previous padding for such strong-diameter partitions of minor-free graphs.

### 5.1 The Algorithm

The algorithm for strong-diameter partitions is similar in spirit to that of [Section 4.1](#) for weak-diameter partitions, but there are some crucial differences that we highlight here.

At a high level, the algorithm works as follows: in each step, pick a connected component of the remaining graph, and find (in a specific way) a shortest path  $P$  in this component. Delete



a random neighborhood of  $P$  from the graph, and recurse on each connected component of the graph, if any. Each such random neighborhood is decomposed into small diameter regions using cones centered at some of  $P$ 's points. A key property to ensure the padding guarantee is that each node is expected to be close to few of these paths. We show that this property holds, otherwise we can construct a  $K_{r+1}$ -minor in  $G$ .

The algorithm again maintains a set of paths (instead of trees), and associated supernodes that will be used in the construction. These will be denoted as  $P_{ij}$  and  $S_i$  respectively, and supernode  $S_i$  consists of the union of neighborhoods of the paths  $P_{ij}$ . The main difference from the weak-diameter construction is that instead of building a shortest-path tree all at once, we build a “tree” one path at a time, and remove a neighborhood of the path from the graph before constructing the subsequent paths.

Let us describe the  $i$ -th iteration of the algorithm. Let  $\mathcal{S} \subseteq V$  be the set containing all the supernodes created so far. Let  $C_i$  be a connected component in the graph  $G_i = G \setminus (\cup \mathcal{S})$ . Let  $\mathcal{S}_{|C_i} = \{S \in \mathcal{S} : S \sim C_i\}$  be the set of supernodes that have a neighbor in component  $C_i$ . We pick an arbitrary vertex  $u_i$  from  $C_i$  and build a supernode  $S_i$ . Again, the intuition behind the construction is that we wish for the new supernode to “touch” every supernode  $S \in \mathcal{S}_{|C_i}$  (i.e.,  $S_i \sim S$ ). However, this is done slightly differently from [Section 4.1](#). At the first iteration ( $j = 1$ ) we create a shortest path  $P_{ij}$  from  $u_i$  to some supernode  $S \in \mathcal{S}_{|C_i}$ , and remove a random neighborhood  $S_{ij}$  from the graph to obtain  $G_{i(j+1)}$ . This neighborhood  $S_{ij}$  is defined as all the vertices within distance  $R_{ij} \cdot \Delta$  of  $P_{ij}$  (in the current component  $C_{ij}$ ), where  $R_{ij} \propto \text{Texp}_{[0,1/4]}(8(r^2 + r))$ . We increase the iteration counter  $j$  and continue in this manner on every connected component of  $G_{ij}$  that is contained in  $C_i$ , until the new supernode  $S_i = \cup_j S_{ij}$  touches every supernode  $S \in \mathcal{S}_{|C}$  for every connected component  $C \subseteq C_i$  in the remaining graph  $G_{ij}$ .

Finally, each such neighborhood  $S_{ij}$  is partitioned to “cones”. Each cone  $B$ , centered at some (yet uncovered) point  $c \in P_{ij}$ , consists of the (yet uncovered) points in  $S_{ij}$  whose distance to  $c$  is not “much larger” than their distance to  $P_{ij}$ . The notion of being “much larger” is determined by a random variable  $\alpha$  drawn independently and uniformly from  $[\Delta/8, \Delta/4]$ . The algorithms are formally presented as Algorithms 3 and 4 respectively. Observe that the subroutine **Create-Cones** is invoked in line 13 of **Strong-Random-Partition**.

## 5.2 The Analysis

We begin by arguing that the algorithm creates a partition  $\mathcal{C}$  with strong-diameter  $\Delta$ . The following properties will be useful.

**Proposition 14.** *For any  $S$  and  $P$  obtained during the run of the algorithm **Create-Cones**:*

- *If  $u, v \in S$  are such that a shortest path from  $u$  to  $P$  contains  $v$ , and  $v \in B$  for a cone  $B$ , then also  $u \in B$ .*
- *If  $u, v \in S$  are such that a shortest path from  $u$  to  $c$  contains  $v$ , and  $u \in B$  for a cone  $B$  centered at  $c$ , then also  $v \in B$ .*

*Proof.* Let  $c \in P$  be the center of the cone  $B$ . We begin by proving the first item: Since  $v \in B$  we have that  $d_S(v, c) - d_S(v, P) \leq \alpha\Delta$ . Since  $v$  is on the shortest path from  $u$  to  $P$ ,  $d_S(u, P) =$

---

**Algorithm 3** Strong-Random-Partition( $G, \Delta, r$ )

---

- 1: Let  $G_0 \leftarrow G$ ,  $i \leftarrow 0$ .
- 2: Let  $\mathcal{S} \leftarrow \emptyset$ .
- 3: Let  $\mathcal{C} \leftarrow \emptyset$ .
- 4: **while**  $G_i$  is non-empty **do**
- 5:   Select a connected component  $C_i$  of  $G_i$ , and pick  $u_i \in C_i$ .
- 6:   Let  $W = \{u_i\}$ .
- 7:   Let  $j = 1$  and  $G_{ij} = G_i \setminus W$ .
- 8:   **while** there exist a connected component  $C_{ij}$  in  $G_{ij}$  and a supernode  $S \in \mathcal{S}_{|C_{ij}}$  such that  $C_{ij} \sim S$  and  $C_{ij} \sim W$  but  $W \not\sim S$  **do**
- 9:     Choose  $u \in N(W) \cap C_{ij}$ .
- 10:     Let  $P_{ij}$  be a shortest path (in  $G_{ij}$ ) from  $u$  to  $N(S)$ .
- 11:     Let  $R_{ij}$  be a random variable drawn independently from the distribution  $\text{Texp}_{[0,1/4]}(8(r^2 + r))$ .
- 12:     Let  $S_{ij} \leftarrow B_{G_{ij}}(P_{ij}, R_{ij}\Delta)$  be a neighborhood of  $P_{ij}$ .
- 13:     **Create-Cones**( $S_{ij}, P_{ij}, \mathcal{C}$ ).
- 14:      $W \leftarrow W \cup S_{ij}$ .
- 15:      $G_{i(j+1)} \leftarrow G_{ij} \setminus S_{ij}$ .
- 16:      $j \leftarrow j + 1$ .
- 17:   **end while**
- 18:   Set  $S_i = W$ , and add  $S_i$  to  $\mathcal{S}$ .
- 19:    $G_{i+1} \leftarrow G_i \setminus S_i$ .
- 20:    $i \leftarrow i + 1$ .
- 21: **end while**
- 22: **return**  $\mathcal{C}$

---

---

**Algorithm 4** Create-Cones( $S, P, \mathcal{C}$ )

---

- 1: **while**  $P \neq \emptyset$  **do**
- 2:   Choose  $c \in P$ .
- 3:   Choose  $\alpha \in [1/8, 1/4]$  uniformly at random.
- 4:   Let  $B = \{u \in S \mid d_S(u, c) - d_S(u, P) \leq \alpha\Delta\}$ .
- 5:   Add  $B$  to  $\mathcal{C}$ .
- 6:   Set  $S \leftarrow S \setminus B$ .
- 7:   Set  $P \leftarrow P \setminus B$ .
- 8: **end while**

---

$d_S(u, v) + d_S(v, P)$  and thus

$$\begin{aligned} d_S(u, c) - d_S(u, P) &\leq (d_S(u, v) + d_S(v, c)) - (d_S(u, v) + d_S(v, P)) \\ &= d_S(v, c) - d_S(v, P) \\ &\leq \alpha\Delta, \end{aligned}$$

which implies that  $u \in B$ .

The second item is proved in a similar manner: Since  $u \in B$  we have that  $d_S(u, c) - d_S(u, P) \leq \alpha\Delta$ . Since  $v$  is on the shortest path from  $u$  to  $c$ ,  $d_S(v, c) = d_S(u, c) - d_S(u, v)$  and thus

$$\begin{aligned} d_S(v, c) - d_S(v, P) &\leq (d_S(u, c) - d_S(u, v)) - (d_S(u, P) - d_S(u, v)) \\ &= d_S(u, c) - d_S(u, P) \\ &\leq \alpha\Delta, \end{aligned}$$

which implies that  $v \in B$ . □

**Lemma 15.** *Each cone  $B$  created in the algorithm has  $\text{diam}(G[B]) \leq \Delta$ .*

*Proof.* Recall that each neighborhood  $S$  of a shortest path  $P$  contains points within distance at most  $\Delta/4$  from  $P$ . Let  $S$  be the remaining part after some cones have been created, and  $P$  is the remaining path. The first property in [Proposition 14](#) implies that the shortest path from any  $u \in S$  to  $P$  is fully contained in  $S$ , and thus

$$d_S(u, P) \leq \Delta/4. \tag{5.15}$$

Consider a certain cone  $B$  centered at  $c \in P$ , and by definition of  $B$ , for each  $u \in B$ ,

$$d_S(u, c) \leq \alpha\Delta + d_S(u, P) \stackrel{(5.15)}{\leq} \Delta/4 + \Delta/4 = \Delta/2. \tag{5.16}$$

By the second property of [Proposition 14](#), if  $u \in B$  then surely any  $v \in S$  on the shortest path from  $u$  to  $c$  will also be in  $B$ , so  $d_B(u, c) \leq \Delta/2$  as well, and thus  $\text{diam}(G[B]) \leq \Delta$ .

Finally, it remains to see that **Create-Cones** generates a partition of  $S$  (i.e. that the clusters it creates cover  $S$ ), and this can be verified by the first property of [Proposition 14](#). If for  $u \in S$  there is a shortest path from  $u$  to  $P$  ending at  $v \in P$ , then whenever  $v$  is covered by a cone,  $u$  must be covered as well (the algorithm does not stop until  $P = \emptyset$ ). □

For a time step  $i$ , we say that  $W$  is the *working supernode*, and at the end of this step it will become the supernode  $S_i$ . Note that  $W$  induces a connected subgraph, because we always choose a vertex  $u$  in  $N(W)$  to be a start of the next path. We denote by  $G_{i0} = G_i$ . The following invariant holds for each time step  $i$ :

**Invariant 2.** *For every  $i, j \geq 0$ , every connected component  $C$  of  $G_{ij}$  satisfies that if  $S, S' \in \mathcal{S}_C$  then  $S \sim S'$ .*

*Proof.* Assume inductively that the invariant holds until time step  $i$  at iteration  $j$ . First consider the case  $j > 0$ , then as  $G_{ij}$  is obtained from  $G_{i(j-1)}$  by removing some vertices, and the set of supernodes remains unchanged, the invariant will still hold: Every connected component  $C$  of  $G_{ij}$

is a subset of a connected component  $D$  of  $G_{i(j-1)}$ , in particular  $\mathcal{S}_C \subseteq \mathcal{S}_D$ , and so any pair of supernodes  $S, S' \in \mathcal{S}_C$  is also in  $\mathcal{S}_D$  and thus  $S \sim S'$ .

For the case  $j = 0$ , a new supernode  $S_{i-1}$  was just introduced, but the termination condition of line 9. guarantees that for any connected component  $C$  in  $G_i$ , any supernode  $S \in \mathcal{S}_C$  must have  $S \sim S_{i-1}$ .  $\square$

**Corollary 16.** *If  $G$  excludes  $K_{r+1}$  as a minor, then for every time step  $i$  and iteration  $j$ , the connected component  $C_{ij}$  has  $|\mathcal{S}_{C_{ij}}| \leq r$ . Moreover, fix some  $z \in V$ . If  $P_{i1}, \dots, P_{il}$  are the shortest paths chosen while creating  $S_i$  in the components containing  $z$ , then  $l \leq r$ .*

*Proof.* If  $|\mathcal{S}_{C_{ij}}| = q$ , then using [Invariant 2](#), contracting each supernode in  $\mathcal{S}_{C_{ij}}$  will yield a  $K_q$  minor, so it must be that  $q \leq r$ . To see the second part of the assertion, note that each  $P_{ij}$  will connect the component containing  $z$  with some supernode  $S \in \mathcal{S}_{C_{ij}}$ , so that  $S_{ij} \sim S$ . Finally, as  $|\mathcal{S}_{C_{ij}}| \leq r$ , there can be at most  $r$  such paths.  $\square$

**Lemma 17.** *For  $\gamma \leq 1/r^2$ , the probability that a ball  $B_z$  of radius  $\gamma\Delta$  is cut by the above process is*

$$\Pr[B_z \text{ is cut}] \leq O(\gamma r^2) .$$

*Proof.* First observe that our algorithm is a threatening skeleton-process with parameters  $l = 0$ ,  $u = 1/4$ ,  $s = r^2 + r$ ,  $b = 2s$  and the  $G_i$  (respectively  $A_i, K_i$ ) are the  $G_{ij}$  (resp.  $P_{ij}, S_{ij}$ ) ordered lexicographically. By [Invariant 2](#) we get that for all  $i, j \in \mathbb{N}$ ,  $|\mathcal{S}_{C_{ij}}| \leq r$ . By [Corollary 16](#), each of these supernodes  $S \in \mathcal{S}_{C_{ij}}$  can have at most  $r$  paths that were built in a component containing  $C_{ij}$ , so it may contribute at most  $r$  to the number of sets in  $\mathcal{K}_{C_{ij}}$ , to a total of  $r^2$ . We must also add in the (at most)  $r$  paths of the current working supernode, to obtain that  $|\mathcal{K}_{C_{ij}}| \leq s$ . For  $u' = u$ , we set  $\mathcal{T}_z = \{P_{ij} \mid d_{G_{ij}}(P_{ij}, z) \leq (u + \gamma)\Delta\}$ , and let  $\tau = \mathbb{E}[|\mathcal{T}_z|]$ . With this we may apply [Lemma 7](#) to infer that

$$\tau \leq 3e^{(2s+1) \cdot (u'+\gamma)/u} = 3e^{(2s+1) \cdot (1+\gamma/u)} .$$

Next, we show that our process is also a cutting skeleton-process, with the graph sequence  $G_{ij}$  and the skeletons are the  $P_{ij}$ , ordered lexicographically. The parameters are the same as before:  $l = 0$ ,  $u = 1/4$  and  $b = 2s$  (this is the exact same process, after all). The condition that the graph sequence contains every uncovered point is trivial by definition of  $G_{ij}$ . By [Lemma 9](#) we obtain that the probability that  $B_z$  is cut is at most

$$\begin{aligned} & (1 - e^{-2b\gamma/(u-l)}) \left( 1 + \frac{\tau}{e^b - 1} \right) \\ & \leq (1 - e^{-20r^2\gamma}) \cdot (1 + 9e^{10r^2\gamma}) \\ & = O(\gamma r^2) , \end{aligned} \tag{5.17}$$

where the last equality follows as  $\gamma \leq 1/r^2$ . In what follows we bound the probability of event  $\mathcal{E}_{\text{cone}}$ , which is the event that the ball  $B_z$  is cut in the **Create-Cones** procedure, while conditioning that it was not cut while creating the  $S_{ij}$ . Let  $S = S_{ij}$  be the set that contains  $B_z$ , which was built around the path  $P = P_{ij}$ . Let  $c_1, \dots, c_k$  be the centers chosen in **Create-Cones**( $S, P, \mathcal{C}$ ). We claim that there can be at most 9 of them that may cut  $B_z$ . To see this, observe that each cone contains a ball of radius at least  $\Delta/8$ , and since  $P$  is a shortest path, in any set of 10 centers there are two centers  $c_g, c_h$  such that  $d_S(c_g, c_h) \geq 9\Delta/8 > 2(1/2 + \gamma)\Delta$ . By the triangle inequality it must be

that at least one of them is more than  $(1/2 + \gamma)\Delta$  away from  $z$ . Finally, by [Lemma 15](#) any cone centered at  $c$  may only contain points at distance at most  $\Delta/2$  from  $c$  (see [\(5.16\)](#)), so it may not be the first to cut  $B_z$ . As  $\alpha$  is chosen uniformly from an interval of size  $\Delta/8$ , the probability that a ball of radius  $\gamma\Delta$  will be cut is at most  $2\gamma\Delta/(\Delta/8) = 16\gamma$ . By a simple union bound,

$$\Pr[\mathcal{E}_{\text{cone}} \mid B_z \subseteq S] < 144\gamma ,$$

which is dominated by [\(5.17\)](#), thus the final bound is

$$\Pr[B_z \text{ is cut}] \leq O(\gamma r^2) .$$

□

## 6 Bounded Tree-width Graphs

In this section we prove the second part of [Theorem 4](#), that any graph with tree-width at most  $r$  admits an efficient  $(O(r), O(1/r))$ -padded strong-diameter partition scheme.

Since graphs of tree-width  $r$  are  $K_{r+2}$ -minor-free, the result of [Section 4](#) already implies a (weak diameter) probabilistic partition which is  $O(r)$ -padded. The purpose of this section is to show a *strong-diameter*  $(O(r), O(1/r))$ -padded partition for graphs of bounded tree-width. We will use the same framework as the previous sections, and exploit the special structure of bounded tree-width graphs.

**Definition 18.** *A graph  $G = (V, E)$  has tree-width  $r$  if there exists a collection of sets  $I = \{X_1, \dots, X_k\}$  with each  $X_i \subseteq V$ , and a tree  $T = (I, F)$ , such that the following conditions hold:*

- $\cup_{i \in [k]} X_i = V$ ,
- For all  $i \in [k]$ ,  $|X_i| \leq r + 1$ ,
- For all  $\{u, v\} \in E$ , there exists  $i \in [k]$  such that  $u, v \in X_i$ ,
- For all  $u \in V$ , the tree nodes containing  $u$  form a connected subtree of  $T$ .

**Corollary 19.** *Let  $U$  be a bag in the tree-decomposition  $T = (I, F)$  of  $G = (V, E)$ . Then if  $U_1, U_2 \in I$  lie in different connected components of  $T \setminus \{U\}$ , and  $x_1 \in U_1 \setminus U$ ,  $x_2 \in U_2 \setminus U$ , then  $x_1, x_2$  are in different connected components of  $G \setminus U$ .*

### 6.1 The Algorithm

Let  $G = (V, E)$  be a graph of tree-width  $r - 1$ , and let  $T$  be its tree-decomposition so that each bag has at most  $r$  vertices, and  $T$  has an arbitrary root  $R$ . The height of a tree node  $U$ ,  $h(U)$ , is its distance in  $T$  from the root  $R$ . For a vertex  $u \in V$  let  $h(u)$  denote the minimal height of a tree node  $U$  containing  $u$ , and denote by  $b(u) = U$  the node achieving this minimum. Order the vertices of the graph  $(v_1, \dots, v_n)$  such that for all  $1 \leq i < j \leq n$ ,  $h(v_i) \leq h(v_j)$ . In the  $i$ -th iteration of the algorithm we will have a graph  $G_i$  (initially  $G_1 = G$ ), and if  $v_i \in G_i$  we shall create a cluster  $S_i = B_{G_i}(v_i, R_i\Delta)$ , where  $R_i \propto \text{Texp}_{[0,1/2]}(8r)$ . Then set  $G_{i+1} = G_i \setminus S_i$  and continue. If  $v_i \notin G_i$  then we do nothing in this iteration.

---

**Algorithm 5** Tree-width-Partition( $G, \Delta, r$ )

---

```
1: Set  $\mathcal{S} = \emptyset$ .
2: Let  $G_1 \leftarrow G$ .
3: for  $i = 1, \dots, n$  do
4:   if  $v_i \in G_i$  then
5:     Let  $R_i \propto \text{Texp}_{[0,1/2]}(8r)$ .
6:     Let  $S_i = B_{G_i}(v_i, R_i\Delta)$ .
7:     Add  $S_i$  to  $\mathcal{S}$ .
8:     Set  $G_{i+1} \leftarrow G_i \setminus S_i$ .
9:   else
10:    Set  $G_{i+1} \leftarrow G_i$ .
11:   end if
12: end for
13: return  $\mathcal{S}$ .
```

---

## 6.2 The Analysis

Fix some  $z \in V$ ,  $\gamma = O(1/r)$  and  $B_z = B_G(z, \gamma\Delta)$ . Let  $U = b(z) \in I$  be the tree node containing  $z$  such that  $h(z) = h(U)$ . The first observation is that when analyzing the probability that  $B_z$  is cut, we may restrict our attention to vertices  $v \in V$  whose  $b(v)$  lies on the path from  $R$  to  $U$  in  $T$ . The reason is that if  $b(v_i)$  is not on this path, then if  $C \in I$  is the least common ancestor of  $U$  and  $b(v_i)$  in  $T$ , we claim that  $G_i$  does not contain any vertex from  $C$ . To see this, note that by the choice of ordering all vertices in  $C$  appear before  $v_i$ , and thus either created a cluster or were removed from the graph. By [Corollary 19](#)  $z$  and  $v_i$  are in different component of  $G_i$ , so  $S_i$  cannot be the first to cut  $B_z$ .

Consider then the process restricted to the vertices contained in bags on the path from  $R$  to  $U$  (we may assume w.l.o.g that these appear first in the ordering). For any  $i \in [n]$ , denote by  $C_i$  the connected component in  $G_i$  that contains  $z$ , and let  $\mathcal{S}_{|C_i} = \{S_j : S_j \sim C_i\}$ .

**Claim 20.** For any  $i \in [n]$ ,  $|\mathcal{S}_{|C_i}| \leq 2r$ .

*Proof.* Let  $R = U_1, \dots, U_k = U$  be the sequence of bags from the root to  $U$  in the tree-decomposition. For any  $j \in [k]$ , let  $i_j \in [n]$  be the minimal such that  $U_j \cap V(G_{i_j}) = \emptyset$ . We prove that  $|\mathcal{S}_{|C_{i_j}}| \leq r$ , by noting that there are at most  $r$  supernodes that can intersect  $U_j$  (as  $|U_j| \leq r$ ). If a supernode  $S_h$  does not intersect  $U_j$ , then since this supernode is not centered at some vertex of  $U_{j'}$  for  $j' > j$  (using the ordering and the minimality of  $i_j$ ), then by [Corollary 19](#) there is no path from  $z$  to  $N(S_h)$  in  $G_{i_j}$ . Since there are at most  $r$  new supernodes created between time  $i_j$  to  $i_{j+1}$  (as each bag is covered after at most  $r$  clusters are formed), the claim follows.  $\square$

Observe that the algorithm generates a threatening skeleton-process with the sequence  $G_1, \dots$ , the skeletons are  $A_i = \{v_i\}$ ,  $K_i = S_i$ ,  $l = 0$ ,  $u = 1/2$ ,  $s = 2r$  and  $b = 4r$ . Let  $u' = u$  and  $\mathcal{J}_z = \{v_i \mid d_{G_i}(z, v_i) \leq (u + \gamma)\Delta\}$ . By [Claim 20](#) we may apply [Lemma 7](#) and obtain that

$$\tau \leq 3e^{(4r+1) \cdot (1+\gamma/u)}. \quad (6.18)$$

Finally, as our process can also be made to be a cutting skeleton-process, as long as we omit the steps in which  $v_i \notin G_i$  (note that the next  $i$  for which  $v_i \in G_i$  may depend on previous random

choices of  $R_j$  for  $j < i$ , but this is allowed), and with  $l = 0$ ,  $u = 1/2$  and  $b = 4r$ . Applying [Lemma 9](#), we obtain that the probability that  $B_z$  is cut is at most

$$(1 - e^{-2b\gamma}) \left(1 + \frac{\tau}{e^b - 1}\right) \leq (1 - e^{-8r\gamma}) \cdot 9e^{8r\gamma} = O(\gamma r),$$

using that  $\gamma \leq 1/r$ .

### 6.3 Bounded Pathwidth Graphs

A graph has path-width  $r$  if it has a tree decomposition of width  $r$  such that the tree is a path. The following result was communicated to us by James R. Lee and Anastasios Sidiropoulos.

- Any graph  $G$  on  $n$  vertices and path-width  $r$  admits an efficient  $O(\log r)$ -padded strong-diameter partition scheme.

We provide a sketch of the proof. First decompose the graph into shortest paths as follows: as long as the graph is not empty, in each connected component, pick a shortest path between a vertex in the first bag to a vertex in the last bag. Remove this path from the graph, and continue on the connected components that remain. Since any such path must use some vertex in every bag, it follows that the path-width decreases by at least 1 in each iteration. We thus obtain a cop-decomposition of width 1 and depth  $r$ . We now apply our method, and the number of threateners is only  $O(r)$ , which implies the result

Note that every graph  $G$  on  $n$  vertices and tree-width  $r$  has path-width at most  $O(r \log n)$  (this follows because it has a tree-decomposition of depth  $O(\log n)$ , see e.g. [\[GTW13\]](#)). An immediate corollary is a  $O(\log r + \log \log n)$ -padded strong-diameter partition for graphs of tree-width  $r$ .

## 7 Bounded Euler-Genus Graphs

In this section we prove the third part of [Theorem 4](#), that any graph with Euler-genus at most  $g$  admits an efficient  $O(\log g)$ -padded strong-diameter partition scheme. We assume here that the graph  $G$  is embedded without any edge crossing on some closed surface  $\Sigma$  (compact and without boundary), which can be orientable or non-orientable, of Euler characteristic  $2 - g$ .

The Euler characteristic is the value  $\chi(\Sigma) = n - e + f$  where  $n, e, f$  are respectively the number of nodes, edges, and faces of the embedding of  $G$  on  $\Sigma$ . If  $\Sigma$  is orientable then  $g$  must be even and  $\Sigma$  homeomorphic to a sphere with  $g/2$  “handles”. And if  $\Sigma$  is non-orientable, then it is homeomorphic to a sphere with  $g$  “cross-caps”. The Euler-genus of  $G$  is the smallest  $g$  such that it can be embedded on a surface of Euler characteristic  $2 - g$ . So, it generalizes the classical notion of genus of a graph (for orientable surfaces) and the non-orientable genus of a graph. Planar graphs have Euler-genus 0.

Using the Fundamental Cycle Method based on BFS trees (see [\[MT01, Lemma 4.2.4 and Theorem 4.3.2\]](#)), we have the following lemma (see also [\[IS07, CdV10\]](#)):

**Lemma 21.** *If  $G$  is a Euler-genus  $g$  graph, there exists a cycle  $A$  comprised of two shortest paths emanating at a common root, such that  $G \setminus A$  has Euler-genus at most  $g - 1$  (this is at most  $g - 2$  if  $A$  is two-sided).*

This fits nicely in the bounded threateners program: Our algorithm will iteratively take such a cycle  $A$ , create a random buffer  $S$  around it, and recurse on the connected components of  $G \setminus S$ .

The base case is when the component is planar, then we may apply our strong-diameter padding algorithm. Formally, in iteration  $i$  take a connected component  $C_i$  in  $G_i$ , if  $C_i$  is not planar, find a cycle  $A_i$  as in [Lemma 21](#). Let  $S_i = B_{G_i}(A_i, R_i\Delta)$  where  $R_i \propto \text{Texp}_{[0,1/4]}(8 \log g)$ , set  $G_{i+1} = G_i \setminus S_i$ . Each  $S_i$  is partitioned to clusters by iteratively taking cones centered at some of the points of  $A_i$ . If  $C_i$  is planar, invoke the decomposition scheme of [Section 5](#).

---

**Algorithm 6** Genus-Partition( $G, \Delta, g$ )

---

```

1: Let  $G_0 \leftarrow G$ ,  $i = 0$ .
2: Let  $\mathcal{C} \leftarrow \emptyset$ .
3: while  $G_i$  is non-empty do
4:   Let  $C_i$  be a connected component of  $G_i$ .
5:   if  $C_i$  is planar then
6:     Let  $P_i$  be a partition obtained by invoking Strong-Random-Partition( $C_i, \Delta, 5$ ). Add the
       clusters of  $P_i$  to  $\mathcal{C}$ .
7:     Set  $G_{i+1} \leftarrow G_i \setminus \cup P_i$ .
8:   else
9:     Let  $A_i$  be cycle as in Lemma 21.
10:    Let  $R_i \propto \text{Texp}_{[0,1/4]}(8 \log g)$ .
11:    Let  $S_i = B_{G_i}(A_i, R_i\Delta)$ .
12:    Create-Cones( $S_i, A_i, \mathcal{C}$ ). Add the resulting clusters to  $\mathcal{C}$ .
13:    Set  $G_{i+1} \leftarrow G_i \setminus S_i$ .
14:   end if
15:    $i \leftarrow i + 1$ .
16: end while
17: return  $\mathcal{C}$ .

```

---

We now turn to analyzing the algorithm. The fact that the resulting partition is strong-diameter  $\Delta$ -bounded follows from the fact that **Strong-Random-Partition** generates strong-diameter  $\Delta$ -bounded clusters, and by [Lemma 15](#), the cones are also strong-diameter  $\Delta$ -bounded (the proof of that lemma never used that  $P$  is a shortest path, we only need that any point in  $S_i$  is within distance  $\Delta/4$  from  $A_i$ ).

Fix some  $z \in V$ ,  $\gamma \leq \delta$  for sufficiently small constant  $\delta$  (which is independent of  $g$ ), and set  $B_z = B_G(z, \gamma\Delta)$ .

**Lemma 22.** *The probability that the ball  $B_z$  is cut by the above process is*

$$\Pr[B_z \text{ is cut}] \leq 1 - e^{-O(\gamma \log g)} .$$

*Proof.* Let  $\mathcal{E}_{\text{genus}}$  be the event that  $B_z$  is first cut by some set  $S_i$ . Divide the event  $\neg \mathcal{E}_{\text{genus}}$  into  $\mathcal{F}_{\text{cone}} = \{\exists i, B_z \subseteq S_i\}$  and  $\mathcal{F}_{\text{planar}} = \{\exists i, B_z \subseteq C_i \wedge C_i \text{ is planar}\}$ . Let  $\mathcal{E}_{\text{cone}}$  be the event that  $\mathcal{F}_{\text{cone}}$  holds and also  $B_z$  is first cut by a cone in the **Create-Cones**( $S_i, A_i, \mathcal{C}$ ), and finally let  $\mathcal{E}_{\text{planar}}$  be the event that  $\mathcal{F}_{\text{planar}}$  holds and also  $B_z$  is cut while calling **Strong-Random-Partition** on a planar component containing  $B_z$ . We will bound each of the  $\mathcal{E}$  events separately.

Assume w.l.o.g that non-planar components are chosen first, then the process until time  $T$  (where all components are planar) is a cutting skeleton-process, with the graph sequence  $G_1, \dots$ , the skeletons  $A_i$  and  $K_i = S_i$ , the parameters are  $l = 0$ ,  $u = 1/4$  and  $b = 2 \log g$ . Let  $\mathcal{T}_z =$



$\{A_i : i \in [T], d_{G_i}(A_i, z) \leq (1/4 + \gamma)\Delta\}$ . Note that by [Lemma 21](#) there can be at most  $g$  iterations (on components containing  $z$ ) in which  $z$  lies in a non-planar component, so  $|\mathcal{T}_z| \leq g$ . By [Lemma 9](#)

$$\Pr[\mathcal{E}_{\text{genus}}] \leq (1 - e^{-16\gamma \log g}) \cdot (1 + g/(e^{2\log g} - 1)) \leq 1 - e^{-32\gamma \log g},$$

using that  $\gamma \leq 1/32$ . If  $\Pr[\neg \mathcal{E}_{\text{genus}}] = p$ , then  $p \geq e^{-32\gamma \log g}$  and if  $p_{\text{cone}} = \Pr[\mathcal{F}_{\text{cone}}]$  and  $p_{\text{planar}} = \Pr[\mathcal{F}_{\text{planar}}]$  then since the events  $\mathcal{F}_{\text{cone}}, \mathcal{F}_{\text{planar}}$  are disjoint, we have that

$$p = p_{\text{cone}} + p_{\text{planar}}. \tag{7.19}$$

By the first assertion of [Theorem 4](#), there is a large constant  $C$  such that

$$\Pr[\mathcal{E}_{\text{planar}}] = p_{\text{planar}} \cdot O(\gamma) = p_{\text{planar}}(1 - e^{-C\gamma}),$$

since  $\gamma$  is sufficiently small.

Finally, we bound the probability of event  $\mathcal{E}_{\text{cone}}$ . Conditioning on  $B_z \subseteq S_i$  for some  $i$ , we use a similar argument as in the proof of [Lemma 17](#), here we claim that there can be at most 18 centers whose cone may intersect  $B_z$ . This is because if there are more, at least 10 of them lie on one of the two shortest path  $A_i$  is comprised of, and using the argument appearing in the proof of [Lemma 17](#), it cannot be that all of them threaten  $B_z$ . Since  $\alpha$  is chosen uniformly from an interval of length  $\Delta/8$ , the probability that any cone cuts  $B_z$  is at most  $2\gamma\Delta/(\Delta/8)$ , thus by a union bound, using that  $C$  is large enough,

$$\Pr[\mathcal{E}_{\text{cone}}] = p_{\text{cone}} \cdot O(\gamma) = p_{\text{cone}}(1 - e^{-C\gamma}).$$

Combining the three bounds, we obtain that the probability that  $B_z$  is cut is at most

$$\begin{aligned} \Pr[\mathcal{E}_{\text{genus}}] + \Pr[\mathcal{E}_{\text{cone}}] + \Pr[\mathcal{E}_{\text{planar}}] &\leq 1 - p + p_{\text{cone}}(1 - e^{-C\gamma}) + p_{\text{planar}}(1 - e^{-C\gamma}) \\ &\stackrel{(7.19)}{=} 1 - p \cdot e^{-C\gamma} \\ &\leq 1 - e^{-32\gamma \log g} \cdot e^{-C\gamma} \\ &= 1 - e^{-O(\gamma \log g)}. \end{aligned}$$

□

## 8 Further Directions

A clear open problem is to improve the  $O(r)$ -padded partition scheme for  $K_r$ -minor-free graphs to the optimal  $O(\log r)$ . A first step might be proving such a result for graphs of tree-width  $r$  (recall that such graphs have a strong-diameter  $O(\log r + \log \log n)$ -padded partition).

### Acknowledgments

We are grateful to Alex Andoni and Daniel Berend for fruitful discussions. A. Gupta and C. Gavaille thank Microsoft Research SVC for their kind hospitality.

## References

- [ABN11] Ittai Abraham, Yair Bartal, and Ofer Neiman. Advances in metric embedding theory. *Advances in Mathematics*, 228(6):3026 – 3126, 2011.
- [AFH<sup>+</sup>04] Aaron Archer, Jittat Fakcharoenphol, Chris Harrelson, Robert Krauthgamer, Kunal Talwar, and Éva Tardos. Approximate classification via earthmover metrics. In *Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1079–1087, New York, 2004. ACM.
- [AGG<sup>+</sup>14] Ittai Abraham, Cyril Gavoille, Anupam Gupta, Ofer Neiman, and Kunal Talwar. Cops, robbers, and threatening skeletons: Padded decomposition for minor-free graphs. In *Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing, STOC '14*, pages 79–88, New York, NY, USA, 2014. ACM.
- [AGMW10] Ittai Abraham, Cyril Gavoille, Dahlia Malkhi, and Udi Wieder. Strong-diameter decompositions of minor free graphs. *Theory Comput. Syst.*, 47(4):837–855, 2010.
- [And86] Thomas Andreae. On a pursuit game played on graphs for which a minor is excluded. *J. Combin. Theory Ser. B*, 41(1):37–47, 1986.
- [AST90] Noga Alon, Paul Seymour, and Robin Thomas. A separator theorem for nonplanar graphs. *J. Amer. Math. Soc.*, 3(4):801–808, 1990.
- [Awe85] Baruch Awerbuch. Complexity of network synchronization. *J. ACM*, 32(4):804–823, October 1985.
- [Bar96] Y. Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science, FOCS '96*, pages 184–, Washington, DC, USA, 1996. IEEE Computer Society.
- [BK96] Hans Leo Bodlaender and Ton Kloks. Efficient and constructive algorithms for the pathwidth and treewidth of graphs. *Journal of Algorithms*, 21(2):358–402, 1996.
- [BLR10] Punyashloka Biswal, James R. Lee, and Satish Rao. Eigenvalue bounds, spectral partitioning, and metrical deformations via flows. *J. ACM*, 57(3), 2010.
- [BLS10] Glencora Borradaile, James R. Lee, and Anastasios Sidiropoulos. Randomly removing  $g$  handles at once. *Comput. Geom.*, 43(8):655–662, 2010.
- [BLT07] Costas Busch, Ryan LaFortune, and Srikanta Tirthapura. Improved sparse covers for graphs excluding a fixed minor. In *Proceedings of the twenty-sixth annual ACM symposium on Principles of distributed computing, PODC '07*, pages 61–70, New York, NY, USA, 2007. ACM.
- [CdV10] Éric Colin de Verdière. Shortest cut graph of a surface with prescribed vertex set. In *18<sup>th</sup> Annual European Symposium on Algorithms (ESA)*, volume 6347 of Lecture Notes in Computer Science, pages 100–111. Springer, September 2010.
- [CKR05] Gruiă Calinescu, Howard Karloff, and Yuval Rabani. Approximation algorithms for the 0-extension problem. *SIAM J. Comput.*, 34(2):358–372, 2004/05.

- [Die00] Reinhard Diestel. *Graph theory*, volume 173 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, second edition, 2000.
- [FHL08] Uriel Feige, MohammadTaghi Hajiaghayi, and James R. Lee. Improved approximation algorithms for minimum weight vertex separators. *SIAM J. Comput.*, 38(2):629–657, 2008.
- [FRT04] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *J. Comput. System Sci.*, 69(3):485–497, 2004.
- [FT03] Jittat Fakcharoenphol and Kunal Talwar. An improved decomposition theorem for graphs excluding a fixed minor. *RANDOM-APPROX*, pages 36–46, 2003.
- [GKL03] Anupam Gupta, Robert Krauthgamer, and James R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *FOCS*, pages 534–543, 2003.
- [GS01] Geoffrey R. Grimmett and David R. Stirzaker. *Probability and random processes*. Oxford University Press, New York, third edition, 2001.
- [GTW13] Anupam Gupta, Kunal Talwar, and David Witmer. Sparsest cut on bounded treewidth graphs: Algorithms and hardness results. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 281–290, New York, NY, USA, 2013. ACM.
- [IS07] Piotr Indyk and Anastasios Sidiropoulos. Probabilistic embeddings of bounded genus graphs into planar graphs. In *Symposium on Computational Geometry*, pages 204–209, 2007.
- [KLPT09] Jonathan A. Kelner, James R. Lee, Gregory N. Price, and Shang-Hua Teng. Higher eigenvalues of graphs. In *FOCS*, pages 735–744, 2009.
- [KMR08] Ken-ichi Kawarabayashi, Bojan Mohar, and Bruce A. Reed. A simpler linear time algorithm for embedding graphs into an arbitrary surface and the genus of graphs of bounded tree-width. In *49<sup>th</sup> Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 771–780. IEEE Computer Society Press, October 2008.
- [KPR93] Philip N. Klein, Serge A. Plotkin, and Satish Rao. Excluded minors, network decomposition, and multicommodity flow. In *STOC*, pages 682–690, 1993.
- [Lee13] James R. Lee. Open question recap, February 2013. <http://tcsmath.wordpress.com/2013/02/25/open-question-recap/>.
- [LGT12] James R. Lee, Shayan Oveis Gharan, and Luca Trevisan. Multi-way spectral partitioning and higher-order cheeger inequalities. In *STOC*, pages 1117–1130, 2012.
- [LN05] James R. Lee and Assaf Naor. Extending Lipschitz functions via random metric partitions. *Invent. Math.*, 160(1):59–95, 2005.
- [LS93] Nathan Linial and Michael Saks. Low diameter graph decompositions. *Combinatorica*, 13(4):441–454, 1993. (Preliminary version in *2nd SODA*, 1991).

- [LS10] James R. Lee and Anastasios Sidiropoulos. Genus and the geometry of the cut graph. In *SODA*, pages 193–201, 2010.
- [Mat02] Jiří Matoušek. *Lectures on discrete geometry*, volume 212 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2002.
- [Moh99] Bojan Mohar. A linear time algorithm for embedding graphs in an arbitrary surface. *SIAM Journal on Discrete Mathematics*, 12(1):6–26, 1999.
- [MT01] Bojan Mohar and Carsten Thomassen. *Graphs on Surfaces*. The Johns Hopkins university Press, 2001.
- [PRS94] Serge Plotkin, Satish Rao, and Warren D. Smith. Shallow excluded minors and improved graph decompositions. In *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (Arlington, VA, 1994)*, pages 462–470, New York, 1994. ACM.
- [Rab03] Yuri Rabinovich. On average distortion of embedding metrics into the line and into  $\ell_1$ . In *Proceedings of the thirty-fifth ACM symposium on Theory of computing*, pages 456–462. ACM Press, 2003.
- [Rao99] Satish B. Rao. Small distortion and volume preserving embeddings for planar and Euclidean metrics. In *SOCG*, pages 300–306, 1999.
- [Ree92] Bruce A. Reed. Finding approximate separators and computing treewidth quickly. In *24<sup>th</sup> Annual ACM Symposium on Theory of Computing (STOC)*, pages 221–228. ACM Press, 1992.
- [RS03] Neil Robertson and Paul D. Seymour. Graph minors. XVI. Excluding a non-planar graph. *Journal of Combinatorial Theory, Series B*, 89(1):43 – 76, 2003.
- [Sid10] Anastasios Sidiropoulos. Optimal stochastic planarization. In *FOCS*, pages 163–170, 2010.
- [WN11] Christian Wulff-Nilsen. Separator theorems for minor-free and shallow minor-free graphs with applications. In *FOCS*, pages 37–46, 2011.