Individual Fairness in Graph Decomposition

Kamesh Munagala [*1] **Govind S. Sankar** [*1]

Abstract

In this paper, we consider classic randomized low diameter decomposition procedures for planar graphs that obtain connected clusters which are cohesive in that close-by pairs of nodes are assigned to the same cluster with high probability. We require the additional aspect of *individual fairness* – pairs of nodes at comparable distances should be separated with comparable probability. We show that classic decomposition procedures do not satisfy this property. We present novel algorithms that achieve various trade-offs between this property and additional desiderata of connectivity of the clusters and optimality in the number of clusters. We show that our individual fairness bounds may be difficult to improve by tying the improvement to resolving a major open question in metric embeddings. We finally show the efficacy of our algorithms on real planar networks modeling congressional redistricting.

1. Introduction

In this paper, we study a clustering problem of partitioning users located in a metric space into components, each of diameter $O(R)$ for some input parameter R. Such partitioning has been widely studied as *low diameter decomposition* (LDD) and has myriad applications. In addition to being a stand-alone clustering procedure, they are, for example, used as a subroutine in metric embeddings (Bartal, 1996; Fakcharoenphol et al., 2004).

In this paper, we consider the question of fairness in how different pairs of nodes are separated in the decomposition. In keeping with terminology in the fairness literature, we term this "individual fairness" of the decomposition.

1.1. Motivation for Individual Fairness

As motivation, consider the clustering problem of opening schools or libraries and assigning users to these locations. In such assignments, it is desirable to preserve communities in the sense that nearby users should be assigned to the same cluster. For instance, in Congressional redistricting (Altman, 1998; Cohen-Addad et al., 2018), it is preferable to keep communities from being fractured (Brubach et al., 2020). Similarly, when assigning students to schools, students who live close to each other prefer to be assigned to the same school. This is termed *community-cohesion* (Ashlagi & Shi, 2014). Of course, no deterministic solution can guarantee community cohesion for all user pairs since some close-by user pairs will be deterministically split. In view of this impossibility, we take the approach of Brubach et al. (2020); Harris et al. (2019) and consider solutions that are a distribution over clusterings. In such a distribution, different pairs of users are separated with different probabilities, and this probability will typically increase with the distance between the pair. This naturally leads to the notion of *pairwise envy-freeness* – pairs of users with comparable distance should be separated with comparable probability. This ensures no pair of users envies a comparable pair in terms of how cohesive the pair is in the decomposition.

As another motivation, in Congressional redistricting, one pertinent question is to generate a *random sample* of representative maps to audit a given map for unfair gerrymandering (DeFord et al., 2021). Ideally, in this random sampling distribution, the marginal probabilities of separation of pairs of precincts of comparable distance should be comparable. This means the probability that any pair of precincts is separated should be *both* lower and upper bounded by a value that depends on the distance between them.

We term this notion of fairness as *individual fairness* due to its connection to a corresponding notion in fair machine learning (Dwork et al., 2012). The main goal of our paper is to study the existence and computation of LDDs that are individually fair.

1.2. Randomized LDD in Planar Graphs

Before proceeding further, we present a formal model for randomized LDD. We consider the canonical case where the users (resp. neighborhoods or other entities) are nodes

[1] Computer Science Department, Duke University, Durham NC 27708-0129, USA. Correspondence to: Kamesh Munagala <kamesh@cs.duke.edu>, Govind S. Sankar <govind.subash.sankar@duke.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

of a planar graph $G(V, E)$ and the distance function $d(u, v)$ is the unweighted shortest path distance between nodes u and v. We now define an LDD as follows:

Definition 1.1 (Low Diameter Decomposition)**.** Given a graph $G(V, E)$ with shortest path distance $d(\cdot, \cdot)$, and given a parameter R (where potentially $R \geq n$), a low-diameter decomposition (LDD) is a (possibly randomized) partition of V into disjoint clusters $\pi_1, \pi_2, \ldots, \pi_r$, so that for j and $u, v \in \pi_j$, $d(u, v) = O(R)$.

In the applications previously mentioned, planar graphs are a natural model. For instance, in political redistricting, nodes correspond to precincts, and edges connect nodes whose corresponding precincts are adjacent. Similarly, in school assignment, nodes correspond to neighborhoods that cannot be split, with edges between nodes corresponding to adjacent neighborhoods. If the LDD partitions the planar graph into connected regions, these regions will also be connected when the nodes are expanded into the corresponding precincts or neighborhoods. This property is an advantage of planar graphs over point set models, where a contiguous set of nodes need not be connected when they are expanded. Finally, the radius constraint on the LDD ensures the expanded regions are sufficiently compact.

Now consider a pair (u, v) of users separated by distance $d(u, v)$. Several randomized LDD constructions lead to cohesive communities by making the probability that (u, v) are separated an increasing function of $d(u, v)$. This ensures close-by pairs are together with a larger probability than pairs further away. The natural benchmark for this probability is the quantity $\rho_{uv} = \frac{d(u,v)}{R}$, as the following example shows.

Example 1. Consider a grid graph where vertex v with coordinates (i, j) has edges to the vertices $(i-1, j)$, $(i+1, j)$, $(i, j-1)$, and $(i, j+1)$. A simple LDD is to choose (ℓ, m) uniformly at random from $[R] \times [R]$ and cut along the x-axis at $\{\ldots, \ell-2R, \ell-R, \ell, \ell+R, \ldots\}$ and the y-axis at $\{\ldots, m-2R, m-R, m, m+R, \ldots\}$. Then, the probability any pair of vertices (u, v) is separated is at most ρ_{uv}.

Indeed, for any planar graph, the classic procedure of Klein et al. (1993) (KPR, see Algorithm 1) provides a practical LDD method for a planar graph into connected components of diameter $O(R)$ so that the probability of separation of any pair (u, v) is $O(\rho_{uv})$.

1.3. Individual Fairness in LDD

We now come to the main focus of this paper: individual fairness. For pairs of nodes (u_1, v_1) and (u_2, v_2) with $d(u_1, v_1) \approx d(u_2, v_2)$, both pairs should be separated with comparable probability, so that neither pair envies the other. Ideally, (u, v) are separated with probability $\Theta(\rho_{uv})$.

In Example 1, the probability that (u, v) are separated is at least $\rho_{uv}/2$, so that this probability is indeed $\Theta(\rho_{uv})$. This may give us hope that the classical KPR procedure also recreates such an individual fairness guarantee for general planar graphs. Note that the classical analysis implies an upper bound of $O(\rho_{uv})$ on the probability (u, v) is separated, but no corresponding lower bound.

Existing Methods can Fail. As our first contribution (Theorem 1.2), we somewhat counter-intuitively show that the KPR procedure (Algorithm 1) cannot yield an individual fairness guarantee – for any distance d, some pair (u, v) with distance $d(u, v) = d$ is never separated.

Theorem 1.2 (Proved in Appendix B)**.** *For any $n \in \mathbb{N}$ and any distance $1 \leq d \leq \frac{\sqrt{n}}{3}$, there exists a planar graph G_d and vertices $u, v \in V(G)$ with $d(u, v) = d$ such that Algorithm 1 with parameter $R > 2d$ never separates u and v.*

This means that though the procedure guarantees that the probability that (u, v) is separated is $O(\rho_{uv})$, the actual probability of separation – zero – need not be comparable to this quantity. Therefore, different vertex pairs of comparable distance can be separated with vastly different probabilities, and this holds even when the vertex pair is an edge.

1.4. Achieving Individual Fairness: Our Results

Given this negative result, our main contribution is a set of algorithms for randomized planar graph partitioning that achieve various levels of individual fairness. We study the trade-offs between three natural desiderata in randomized LDDs – each part in the decomposition should be a connected component, these components should have non-trivial size, and the partitioning should be individually fair.

We first state the above desiderata formally:

(CON): Each cluster forms a single connected component. This is important in applications such as school assignment or political redistricting (Duchin, 2018).

(COMP): The LDD should optimally compress the graph. For a general unweighted planar graph with n nodes, we say that a procedure satisfies (COMP) if (i) the expected number of components produced is $O(n/R)$;[1] and (ii) for any outcome of randomness, the algorithm should achieve non-trivial compression: When $R = \frac{D}{\beta}$, where D is the diameter of the graph and $\beta \geq 1$, there always exist $\Omega(\beta)$ components with diameter $\Omega(R)$.

(f, g)**-(IF):** This condition captures individual fairness. The probability that (u, v) are separated is at most $f(\rho_{uv})$ and at least $g(\rho_{uv})$, for scalar functions f, g. For an arbitrary function h, we say that $f(\rho) =$

[1] This bound cannot be improved for a line graph.

$O(h(\rho))$ if $f(\rho_{uv}) = O(h(\rho_{uv}))$ for every pair u, v; similarly for g. In the ideal case, as in Example 1, $\Pr[(u, v) \text{ is cut}] = \Theta(\rho_{uv})$ for all u, v, so that $f(\rho) = O(\rho)$ and $g(\rho) = \Omega(\rho)$ simultaneously.

Note that the KPR procedure (Algorithm 1) achieves (CON), (COMP), and $f(\rho) = O(\rho)$. However, by Theorem 1.2, it has $g(\rho) = 0$, and is hence not individually fair.

Fairness for Edges. We first show that if we only seek to be individually fair for pairs $(u, v) \in E$, that is, edges or $d(u, v) = 1$, then there is a modification of the KPR procedure via random vertex weights (Algorithm 2) that preserves (COMP) and (CON), and guarantees that any edge (u, v) is separated with probability $\Theta\left(\frac{1}{R}\right)$. We state the theorem in a bit more generality below. We show a tighter bound for the case of $d(u, v) = 1$ since we present experiments on this case in Section 7.

Theorem 1.3 (Proved in Section 3)**.** *There is an LDD procedure (Algorithm 2) that produces connected components of diameter at most $43 \cdot (R + 1)$, satisfies (COMP), and every pair $(u, v) \in V \times V$ satisfies*

$$\frac{1}{2R} \le \Pr[u, v \textit{ separated}] \le \begin{cases} 3\rho_{uv} + \frac{3}{R} & \textit{if } d(u, v) > 1 \\ \frac{3}{R} & \textit{if } d(u, v) = 1 \end{cases}.$$

Lower Bound on Individual Fairness. Moving on to pairs (u, v) separated by general distances, we show that obtaining an individual fairness guarantee of $\Theta(\rho_{uv})$ resolves a major conjecture in metric embeddings and is hence likely difficult. Formally, we tie individual fairness in LDD precisely to the distortion of embedding the graph metric into ℓ_1. Such an embedding is a function h that maps the vertices v of G to s-dimensional points $h(v)$ (for some s) with the guarantee that the ℓ_1 distance between $h(u)$ and $h(v)$ is at least $d(u, v)$, and at most $\alpha \cdot d(u, v)$. The parameter $\alpha \ge 1$ is termed the *distortion*.

Theorem 1.4 (Proved in Section 4)**.** *If there is an LDD procedure for planar graph*[2] *G that with any parameter*[3] *$R = \omega(n)$ that achieves $f(\rho)/g(\rho) \le \gamma$ and $f(\rho) = O(\rho)$, then there is an ℓ_1 embedding of G with distortion $O(\gamma)$.*

The best-known distortion bound for embedding a planar graph into ℓ_1 is $O(\sqrt{\log n})$, and it is a major open question to improve this to a constant. This implies obtaining the "optimal bound" – $f(\rho) = O(\rho)$ and $g(\rho) = \Omega(\rho)$ simultaneously resolves a major conjecture, since by Theorem 1.4, it implies a constant distortion embedding of G into ℓ_1.

Positive Results for General Distances. Given the above difficulty, we show algorithms that achieve increasingly stronger individual fairness guarantees as we relax the requirements of (COMP) and (CON).

First, we show the following theorem that extends Theorem 1.3 to derive a lower bound on separation probability for non-adjacent vertex pairs.

Theorem 1.5 (Proved in Section 5)**.** *There is an algorithm (Algorithm 3) that, for any $\epsilon > 0$, achieves (CON), (COMP), $f(\rho) = O(\rho)$, and $g(\rho) = \Omega(\epsilon^2 \cdot \rho^{2+\epsilon})$ for any $d(u, v)$.*

The above theorem is technically the most intricate. We extend the classical analysis in (Klein et al., 1993) by designing a novel modification of their algorithm. Our analysis hinges on showing the existence of a $K_{3,3}$ minor (and hence a contradiction) if a vertex pair (u, v) is not separated with sufficient probability. At a high level, this is similar to the analysis in (Klein et al., 1993). However, they show such a minor for deriving the diameter of the decomposition. In contrast, in our case, the diameter follows from their argument but we need the minor construction to show the lower bound. This makes the details of our construction different.

We next show in Section 6 that the above bound on g can be improved if we violate (COMP).

Theorem 1.6 (Proved in Section 6)**.** *For any constant $\epsilon > 0$, there is an LDD procedure (Algorithm 4) that achieves (CON), $f(\rho) = O(\rho)$, and $g(\rho) = \Omega(\epsilon \cdot \rho^{1+\epsilon})$. A similar procedure also gives $f(\rho) = O(\rho)$ and $g(\rho) = \Omega\left(\frac{\rho}{\log R}\right)$.*

The algorithm that shows the above theorem runs the classical KPR algorithm with a random choice of radius parameter. Such a procedure will preserve connectivity, but the random choice of radius leads to sub-optimal compression, even for a line.

We finally consider the implications of violating both (CON) and (COMP). We show the following theorem.

Theorem 1.7 (Proved in Appendix C)**.** *For any planar graph and parameter R, there is an LDD (Algorithm 5) with $f(\rho) = O(\rho)$ and $g(\rho) = \Omega\left(\frac{\rho}{\sqrt{\log R}}\right)$.*

The algorithm for showing the above theorem involves embedding the planar graph into the ℓ_2 metric and projecting the embedding onto a line, reminiscent of locality-sensitive hashing methods for the ℓ_2 metric. Such an approach does not preserve either connectivity or (COMP). Note that by Theorem 1.4, improving the above result for $R = \omega(n)$ would make progress on the conjecture of optimally embedding planar graphs into ℓ_1.

The trade-offs above are summarized in Table 1. Note that the gap between f and g improves as we relax the other desiderata of (CON) and (COMP).

[2] This theorem holds for arbitrary metric spaces.

[3] Note that Definition 1.1 allows for the parameter R to be larger than the diameter of G. In this regime, the constraint on the diameter is vacuous, and the only requirement is to satisfy the bounds on the separation probabilities.

	(CON)	(COMP)	$g(\rho)$
KPR (Alg. 1)	✓	✓	–
Theorem 1.5 (Alg. 3)	✓	✓	$\Omega(\epsilon^2 \cdot \rho^{2+\epsilon}), \epsilon > 0$
Theorem 1.6 (Alg. 4)	✓	–	$\Omega(\epsilon \cdot \rho^{1+\epsilon}),\ \epsilon > 0$ $\Omega\left(\frac{\rho}{\log R}\right)$
Theorem 1.7 (Alg. 5)	–	–	$\Omega\left(\frac{\rho}{\sqrt{\log R}}\right)$

Table 1: Trade-offs in Desiderata. In all cases, $f(\rho) = O(\rho)$. The notation Ω_ϵ hides terms dependent on ϵ.

Empirical Results. We perform experiments on real-world planar graphs modeling precinct connectivity in redistricting applications in the states of North Carolina, Maryland, and Pennsylvania in the United States. We consider individual fairness on edges (Theorem 1.3 and Algorithm 2). We show an approach to establish that Algorithm 2 is more individually fair than Algorithm 1 (KPR) across these datasets. Focusing on Algorithm 2, we also show that empirically, the radii of the clusters are much smaller than the theoretical bounds (Theorem 2.1). Further, the number of components satisfies the $O(n/R)$ bound on the number of clusters from (COMP) not just in expectation but almost always.

1.5. Related Work

Low Diameter Decomposition. Randomized low diameter decomposition of metric spaces is widely used as a subroutine for embedding the metric space into a simpler space, such as trees or Euclidean metrics, with provably low distortion, meaning that no distance is stretched too much. For general metric spaces, the work of Bartal (1996); Fakcharoenphol et al. (2004) present LDD procedures that achieve a probability of separation $O(\rho_{uv} \cdot \log n)$, where n is the number of points in the metric space. For graphs excluding a K_r minor, which includes planar graphs, the algorithm of Klein et al. (1993) has been a predominant tool. Abraham et al. (2019) devise an entirely different framework using 'cop decompositions' that yields improved separation probability bounds for large r. This, however, does not yield any improvement for planar graphs ($r = 5$), so we focus on the simpler KPR algorithm in this paper. We refer the reader to the excellent survey by Matoušek (2002) for various embedding results and their connection to LDD. In contrast, we consider low diameter decomposition as a stand-alone clustering concept and study individual fairness in the separation probabilities it generates and how this fairness trades off with other natural clustering desiderata.

Fairness. Over the last decade, fairness has become an important algorithmic objective in machine learning (Chouldechova & Roth, 2018). Fair matchings (Huang et al., 2016), rankings (Celis et al., 2017), and bandit problems (Joseph et al., 2016) have been considered, among others. Following the work of Chierichetti et al. (2017), fair clustering has seen much interest (Backurs et al., 2019; Bera et al., 2019; Chen et al., 2019; Esmaeili et al., 2020).

Within the literature on fairness, machine learning literature has considered the notion of individual fairness (Dwork et al., 2012), where the goal is to find a randomized solution (classifier, regression) such that close-by data points see similar outcomes. This has inspired similar models for problems such as auctions (Chawla & Jagadeesan, 2022) and assignments (Devic et al., 2023; Shen et al., 2023). We extend this literature to consider fairness in separation probability in randomized clustering.

Clustering. Our problem is closely related to classical k-center clustering, where the goal is to open at most k such locations and assign users to these locations. Though this problem is NP-HARD, several efficient algorithms are known that achieve a 2-approximation when the points lie in a metric space (Hochbaum & Shmoys, 1985; Hsu & Nemhauser, 1979). Of particular interest is the work of Brubach et al. (2020) (see also (Miller et al., 2013)) that performs randomized k-center clustering and guarantees a probability of separation of $O(\rho_{uv} \cdot \log k)$. However, their work does not guarantee individual fairness for pairs of users with comparable distances. We use this line of work as motivation for studying randomized partitioning and seeking individual fairness guarantees. In contrast with k-center clustering, we do not enforce a limit k on the number of clusters since no non-trivial fairness guarantee can be obtained with this restriction.[4]

2. The KPR Partitioning Algorithm

For the sake of completeness, we present the KPR algorithm (Klein et al., 1993) as Algorithm 1. This achieves a randomized LDD with the additional properties that: (a) each cluster π_j is a connected component; (b) for any $u, v \in \pi_j$, the shortest path from u to v in the subgraph induced by π_j has length $O(R)$; and (c) the probability that u, v belong to different clusters (or are "separated") is $O(\rho_{uv})$. Recall that $\rho_{uv} = \frac{d(u,v)}{R}$, where $d(u, v)$ is the shortest path distance (number of edges) between u and v in G.

First, we present some terminology. We term the maximum shortest path length within the subgraph induced by a cluster as its "diameter". For a BFS tree rooted at vertex r, we say

[4] For example, consider 4 points located on the real line at $0, 1, x, x + 1$. If we are only allowed to open 2 clusters, then as $x \to \infty$, the pairs of close-by points cannot be separated without drastically affecting the cluster radii.

that r is at level 0, the children of r are at level 1, and so on. We say an edge (u, v) is at level i if u is in level i and v is at level $i+1$ or vice versa. We call an iteration of the loop in step 1 one 'phase' of Algorithm 1.

Algorithm 1 KPR Decomposition (Klein et al., 1993)

```
Input: Integer $R$, planar graph $G = (V, E)$.
 1: Set $F \leftarrow \{\}$.
 2: for $i = \{1, 2, 3\}$ do
 3:   for every connected component $C$ of $G \setminus F$ do
 4:     Perform BFS from an arbitrary root $r$ in $C$.
 5:     Sample $k$ uniformly at random from $\{0, 1, \dots R-1\}$.
 6:     for every edge $e$ at level $\ell \equiv k \mod R$ from $r$ do
 7:       $F \leftarrow F \cup \{e\}$
 8:     end for
 9:   end for
10: end for
```

Algorithm 1 can also be viewed in the following way. At step 1, define V_δ as the vertices whose levels lie in $[\delta R + k, (\delta+1)R + k)$ for integers δ. Then, the algorithm can instead remove all edges leaving V_δ in step 1, for all δ.

Theorem 2.1 ((Klein et al., 1993; Goemans, 2006)). *Algorithm 1 partitions the vertices of G into clusters, each of which is a disjoint connected component of diameter at most $43 \cdot R$. Further, for any $(u, v) \in V \times V$:*

$$\Pr[(u, v) \text{ are separated}] \leq 3 \cdot \rho_{uv}.$$

The procedure satisfies (CON) by definition. Lemma 2.2 shows that it satisfies (COMP). In Appendix B, we show that it does not satisfy individual fairness, since there exist graphs where a pair of vertices are never separated. The following lemma is proved in Appendix A.

Lemma 2.2 ((COMP)). *For any planar graph with n vertices and diameter D, the expected number of components produced by Algorithm 1 is at most $3n/R$. Further, regardless of the randomness, there always exist $\lfloor \beta/4 \rfloor$ components with diameter at least $R/8$ assuming $R = \frac{D}{\beta}$.*

3. Individual Fairness for Edges

We will now prove Theorem 1.3 via Algorithm 2. The algorithm is similar to Algorithm 1, except it adds random weights to each vertex and cuts either above or below the vertex in the BFS tree depending on the vertex weight.

The diameter guarantee follows from the proof of Theorem 2.1. Each phase of Algorithm 2 has at most $R+1$ levels in the BFS tree, and we reuse the bounds from Algorithm 1 with parameter $R+1$ instead of R. It is easy to check that the procedure satisfies (CON), and a similar argument to Lemma 2.2 shows (COMP).

Algorithm 2 KPR algorithm with random vertex weights

```
Input: Integer $R$, planar graph $G = (V, E)$.
 1: Set $F \leftarrow \{\}$.
 2: for $i = \{1, 2, 3\}$ do
 3:   for every connected component $C$ of $G \setminus F$ do
 4:     Perform BFS from an arbitrary root $r$ in $C$.
 5:     Sample $\theta$ uniformly at random from $\{0, 1, \dots R-1\}$.
 6:     for every vertex $v$ at level $\ell_v \equiv \theta \mod R$ do
 7:       Independently choose $b_v \in \{0, 1\}$ uniformly at random.
 8:       $\ell_v \leftarrow \ell_v + b_v$.
 9:     end for
10:     For non-negative integers $q$, define $V_q = \{v \mid qR + \theta < \ell_v \leq (q+1)R + \theta\}$.
11:     Let $E_q$ denote edges with one end-point in set $V_q$, and let $E' = \cup_{q \geq 0} E_q$.
12:     $F \leftarrow F \cup E'$.
13:   end for
14: end for
```

To upper-bound the separation probability for an edge, consider the BFS tree just before executing Step (2) in some phase of the algorithm. If u, v are on the same level, then that level is $\theta \mod R$ with probability $\frac{1}{R}$ in this phase. If this does not happen, u and v will not be separated in this phase. Suppose u and v are not on the same level. Without loss of generality, assume that u is in level ℓ and v is in level $\ell + 1$. If $\ell \equiv \theta \mod R$ (which happens with probability $\frac{1}{R}$), then $b_u = 0$ with probability $1/2$ and (u, v) are subsequently separated. This event, therefore, happens with probability $\frac{1}{2R}$. Similarly, if $\ell + 1 \equiv \theta \mod R$, then (u, v) are subsequently separated if $b_v = 1$, which again happens with probability $\frac{1}{2R}$. In all other events, (u, v) are not separated. Therefore, in each phase, the probability that u and v are separated is at most $\frac{1}{R}$, and the upper bound follows by taking the union bound over three phases. The bound for more general distances is analogous.

We will only consider the first phase to get the lower bound on the probability of separation. Assume without loss of generality that u is at least as close to the root of the BFS tree as v. If u and v are at the same level ℓ at the beginning of Step (2), then with probability $\frac{1}{R}$, θ is chosen as $\ell \mod R$. In this event, u and v are separated if $b_u \neq b_v$, which happens with probability $\frac{1}{2}$. This gives us a probability of $\frac{1}{2R}$ that they are separated. Similarly, if $\ell_u < \ell_v$ at the beginning of Step (2), then with probability $\frac{1}{2R}$, $\theta \equiv \ell_v \mod R$ and $b_u = 0$, in which case u and v are separated. This completes the proof of Theorem 1.3.

4. Lower Bound on Individual Fairness

We next prove Theorem 1.4. Let $n = |V|$. Suppose for any parameter $R = \omega(n)$, there is an LDD procedure that produces clusters such that the probability that u, v are separated lies in $\left[c_1 \cdot \frac{d(u,v)}{R}, c_2 \cdot \frac{d(u,v)}{R}\right]$ where $c_2 = O(1)$. Note that assuming $R = \omega(n)$ means the diameter of the resulting clusters are irrelevant, and all we care about is the separation probability.

Create an embedding h of G into ℓ_1 as follows. Sample $\eta = m \cdot R$ such partitions $\{\sigma_i\}_{i=1}^{\eta}$, where $m = \omega(n^2)$. For each cluster C in partitioning σ_i, choose $q_C^i = +1/m$ with probability $1/2$ and $q_C^i = -1/m$ with probability $1/2$. For each $v \in C$, set $t_{iv} = q_C^i$. The embedding has η dimensions and the i^{th} dimension of $h(v)$ is t_{iv}.

We have the following lemma for any $(u, v) \in V \times V$:

Lemma 4.1. *For any (u, v), with probability $1 - \frac{1}{m}$,*

$$\frac{c_1}{2} \cdot d(u,v) \cdot \le \|h(u) - h(v)\|_1 \le 4c_2 \cdot d(u,v).$$

Proof. Let h_{ui} be the i^{th} coordinate of $h(u)$. In each dimension, if (u, v) are separated, they get different tokens with probability $1/2$, in which case their distance is $2/m$ in that dimension. Otherwise, their distance is zero. This implies

$$\mathsf{E}[|h_{ui} - h_{vi}|] = \frac{\Pr[(u,v) \text{ are separated }]}{m}.$$

Therefore,

$$\begin{aligned}\mathsf{E}[\|h(u) - h(v)\|_1] &= \eta \cdot \frac{\Pr[(u,v) \text{ are separated }]}{m} \\ &= c \cdot d(u,v) \;\text{ for } c \in [c_1, c_2].\end{aligned}$$

Let $\mu_{uv} = \mathsf{E}[\|h(u) - h(v)\|_1]$. Note now that $\|h(u) - h(v)\|_1$ is the sum of R independent random variables each in $\{0, 2/m\}$ and further that $\mu_{uv} \ge c_1 \cdot d(u,v) \ge d(u,v) \ge 1$. Using Chernoff bounds, we have:

$$\Pr[\|h(u) - h(v)\|_1 \notin \mu_{uv} \cdot (1 \pm 0.5)] \le 2 \cdot e^{-m/8} \le \frac{1}{m}$$

for large enough m. □

Since $m = \omega(n^2)$, the previous lemma implies that with constant probability, for all $(u, v) \in V \times V$, we have

$$\frac{c_1}{2} \cdot d(u,v) \cdot \le \|h(u) - h(v)\|_1 \le 4c_2 \cdot d(u,v).$$

Scaling the coordinates up by $2/c_1$, this yields a non-contractive embedding into ℓ_1 with distortion $O(c_2/c_1)$, completing the proof of Theorem 1.4.

5. Individual Fairness for General Distances

We now prove Theorem 1.5 via Algorithm 3. The key idea of the algorithm is to use two cut sequences in each phase of Algorithm 1.

Algorithm 3 KPR algorithm with two cuts per phase

```
Input: Integer $R$, planar graph $G = (V, E)$.
 1: Run the iteration of Algorithm 1 once (for $i = 1$) on $G$ from an arbitrary root $r_0$.
 2: Let $F$ be the set of removed edges.
 3: for $i = \{1, 2\}$ do
 4:   for every connected component $C$ of $G \setminus F$ do
 5:     Perform BFS with root $r_i$ set to be the vertex closest to the previous root $r_{i-1}$ used for creating the component, breaking ties arbitrarily.
 6:     Sample $k_1$ uniformly at random from $\{0, 1, \ldots R - 1\}$.
 7:     Sample $\kappa$ from density $\frac{\epsilon}{\kappa^{1-\epsilon} R^{\epsilon}}$; set $k_2 = \lceil \kappa \rceil$.
 8:     for every edge $e$ at levels $\ell \equiv k_1 \mod R$ or $\ell \equiv k_1 + k_2 \mod R$ from $r_i$ do
 9:       $F \leftarrow F \cup \{e\}$.
10:     end for
11:   end for
12: end for
```

To prove Theorem 1.5, first, note that it satisfies (CON) by definition. The diameter guarantee of $43 \cdot R$ follows directly from Theorem 2.1. Simply observe that if we just consider the cuts induced by k_1 in Step 3, we have effectively run three phases of Algorithm 1. Because of k_2, we make some additional cuts in the second and third phases, which can only decrease the diameter.

Lemma 5.1. $\Pr[u, v$ *are separated*$] = O\left(\frac{d(u,v)}{R}\right)$.

Proof. We use union bounds. The three cuts made in the first step and by k_1 in Step 3 add $3 \cdot \rho_{uv}$ to the probability of separation. We now bound the probability that u, v are separated by k_2 in Step 3. Consider the path of length $d = d(u,v)$ between u, v in the tree. They are separated only if at least one edge of this path is at some level $\ell \equiv k_1 + k_2 \mod R$. Suppose that the edges of this path lie between levels ℓ_1 and ℓ_2 where we know that $\ell_1 - \ell_2 \le d - 1$. If $\ell_1 \equiv k_1 + x \mod R$, then the probability that some edge of this path is cut is the probability that $\kappa \in [x, x + \ell_2 - \ell_1) \subseteq [x, x + d)$. This happens with probability

$$\int_x^{x+d} \frac{\epsilon}{\kappa R^{\epsilon}} d\kappa = \frac{\epsilon((x+d)^{\epsilon} - x^{\epsilon})}{R^{\epsilon}}.$$

Since k_1 is chosen uniformly at random, x is also chosen uniformly at random from $\{0, 1, \ldots, R - 1\}$. Thus, a cut is made with probability

$$\sum_{x=0}^{R-1} \frac{\epsilon((x+d)^\epsilon - x^\epsilon)}{R^{1+\epsilon}} \leq \int_0^R \frac{\epsilon((x+d)^\epsilon - x^\epsilon)}{R^{1+\epsilon}} dx$$
$$\leq \frac{\epsilon((R+d)^{1+\epsilon} - R^{1+\epsilon})}{(1+\epsilon)R^{1+\epsilon}}.$$

Choosing $\epsilon \leq 1$, this is at most $2d/R = 2\rho_{uv}$, giving an overall probability of separation of at most $5\rho_{uv}$. □

We prove the following lemma in Appendix A, and it is similar to Lemma 2.2.

Lemma 5.2. *Algorithm 3 satisfies (COMP).*

We will now show the following theorem, which yields Theorem 1.5 for $d(u,v) \geq 59$.

Theorem 5.3. *For $d(u,v) \geq 59$, Algorithm 3 satisfies*

$$\Pr[u, v \textit{ are separated}] \geq \frac{\epsilon^2}{16} \cdot \left(\frac{\lfloor d(u,v)/59 \rfloor}{R}\right)^{2+2\epsilon}.$$

For (u,v) with $d(u,v) < 59$, we first run Algorithm 2 and note that Theorem 1.3 implies $\Pr[u, v \text{ separated}] = \Omega(1/R) = \Omega(\rho_{uv}) = \Omega(\rho_{uv}^{2+2\epsilon})$. We subsequently run Algorithm 3 on each resulting component, hence yielding Theorem 1.5.

5.1. Proof of Theorem 5.3

Fix some pair (u,v) of vertices and let $d = d(u,v)$. We assume d is a multiple of 59; otherwise, we can use $59\lfloor \frac{d}{59} \rfloor$ as a lower bound for d. First, note that we can assume $d \leq 43 \cdot R$, else u, v are separated with probability 1. Define $\hat{\rho} = \frac{\rho_{uv}}{59}$ and $\hat{d} = \frac{d}{59}$. From above, we get $\hat{\rho} \leq 1$. Further, assume w.l.o.g. that (u,v) are not cut in Step (3).

Conditioned on this event, let G_1 denote the component to which (u,v) belong. In the next iteration (Step (3) for $i = 1$), let the root chosen for this component be r_1, and the corresponding BFS tree be T_1. Let $d_1(x,y)$ denote the distance between x, y in G_1 and let $\ell_1(u) = d_1(r_1, u)$ denote the level of the tree T_1 (measured from the root) that u belongs to.

Events L_1 and E_1: Let L_1 be the event that $|\ell_1(u) - \ell_1(v)| < \hat{d}$. If L_1 does not happen, then (u,v) are cut in Step (3) for $i = 1$ with probability at least $\hat{\rho}$ and we are done. We proceed assuming event L_1 happens and define the first sandwich event E_1 as follows.

Definition 5.4. In event E_1, in Step (3) for $i = 1$, the (k_1, k_2) chosen satisfy:

- $k_1 \equiv \ell \mod R$ for some ℓ satisfying
$$\min(\ell_1(u), \ell_1(v)) \geq \ell \geq \min(\ell_1(u), \ell_1(v)) - \hat{d}$$
- $k_1 + k_2 \equiv \ell \mod R$ for some ℓ satisfying
$$\max(\ell_1(u), \ell_1(v)) \leq \ell \leq \max(\ell_1(u), \ell_1(v)) + \hat{d}$$

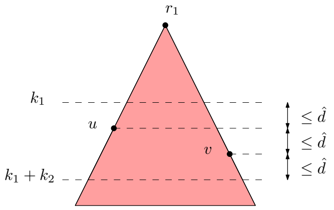


Figure 1: An example of the event $E_1 \cap L_1$ in T_1 (red). Both the cuts made by k_1, k_2 are within $2\hat{d}$ levels of both u, v, which are themselves within $\hat{d}$ levels of each other.

In other words, the first sandwich event happens when the first cut (corresponding to k_1) for that iteration $i = 1$ is made within $\hat{d}$ levels above the higher of the levels of u, v in T_1 and the second cut (corresponding to k_2) in that iteration is made within $\hat{d}$ levels below the level of the lower of u, v in T_1. This is illustrated in Figure 1. Observe that for a sandwich event to happen, k_1 must take one of $\hat{d}$ possible values, which happens with probability $\hat{\rho}$, and κ must lie in the range $[\max(\ell(u), \ell(v)) - k_1, \max(\ell(u), \ell(v)) - k_1 + \hat{d})]$. Since $\max(\ell(u), \ell(v)) - k_1 \in [\hat{d}, 2\hat{d}]$, this probability is at least $\frac{(3\hat{d})^\epsilon - (2\hat{d})^\epsilon}{R^\epsilon} \geq \frac{\epsilon}{4} \cdot \hat{\rho}^\epsilon$. This gives an overall probability of $\Pr[E_1] = \frac{\epsilon}{4} \cdot \hat{\rho}^{1+\epsilon}$. Conditioned on $L_1 \cap E_1$, the pair (u,v) could still be connected. If they are not connected, then we have shown the theorem. Suppose they are still connected, then let the resulting component be G_2. Define the root r_2 of the BFS tree T_2 on G_2 analogous to above, and let $\ell_2(x)$ denote the level of node $x \in V(G_2)$ in this tree.

Events L_2 and E_2: We assume the event L_2 that $|\ell_2(u) - \ell_2(v)| < \hat{d}$ happens. Suppose not, then (u,v) are cut in Step (3) for $i = 2$ with probability at least $\hat{\rho}$. Combining this with the bound for $\Pr[E_1]$ above, we are done. Therefore, assume L_2 happens. Then, define the second sandwich event E_2 analogous to Definition 5.4, with $i = 2$ and ℓ_2 used instead of $i = 1$ and ℓ_1. We have,

$$\Pr[E_2|E_1] = \frac{\epsilon}{4} \cdot \hat{\rho}^{1+\epsilon} \implies \Pr[E_1 \cap E_2] \geq \frac{\epsilon^2}{16} \cdot \hat{\rho}^{2+2\epsilon}.$$

The following lemma completes the proof of Theorem 5.3.

Lemma 5.5 (Proved in Appendix D)**.** *Conditioned on $E_1 \cap E_2$ (assuming $L_1 \cap L_2$ happens), u and v are always split.*

We prove Lemma 5.5 by contradiction. Suppose not and that u, v lie in the same connected component after the two phases. Let G_3 be this connected component. Our approach will be to exhibit a $K_{3,3}$ minor in the graph, contradicting the planarity of G. This construction is the technical heart of the proof, and we present in Appendix D.

6. Individual Fairness without (COMP)

We now prove Theorem 1.6 via Algorithm 4. The key idea is to run Algorithm 1 with the parameter drawn from a distribution depending on R instead of the parameter being R directly.

```
Algorithm 4 KPR algorithm with random diameter
Input: Integer $R$, $\alpha \in \mathbb{R}_{\geq 0}$, planar graph $G = (V, E)$
Sample $r \in (0, R]$ according to density $\frac{(1+\alpha)r^{\alpha}}{R^{1+\alpha}}$.
Run Algorithm 1 with the parameter $\lceil r \rceil$ in place of $R$.
```

Lemma 6.1. *Algorithm 2 produces connected components of diameter $O(R)$ and for every $u, v \in V(G)$, satisfies*

$$\frac{\rho_{uv}}{43} \leq \Pr[u, v \textit{ are separated}] \leq 3\rho_{uv} \log\left(\frac{1}{\rho_{uv}}\right)$$

if $\alpha = 0$ and if $\alpha > 0$, satisfies

$$\left(\frac{\rho_{uv}}{43}\right)^{1+\alpha} \leq \Pr[u, v \textit{ are separated}] \leq 3\rho_{uv}\left(1 + \frac{1}{\alpha}\right).$$

Proof. The diameter bound follows directly from Theorem 2.1 since the largest parameter with which we run Algorithm 1 is R. Similarly, the connectivity guarantee follows since Algorithm 1 finds connected components.

For the lower bound, observe that if $43r < d(u, v)$, then they are separated with probability 1 from the statement of Theorem 2.1. This happens with probability

$$\int_0^{\frac{d(u,v)}{43}} \frac{(1+\alpha)r^{\alpha}}{R^{1+\alpha}} dr = \left(\frac{d(u, v)}{43R}\right)^{1+\alpha}.$$

For the upper bound, we consider them to be separated with probability 1 if $r < 3d(u, v)$. If $r \geq 3d(u, v)$, then they are separated with probability $\frac{3d(u,v)}{r}$ from Theorem 2.1. This gives us an overall probability of

$$\int_0^{3d(u,v)} \frac{(1+\alpha)r^{\alpha}}{R^{1+\alpha}} dr + \int_{3d(u,v)}^{R} \frac{3d(u, v)}{r} \frac{(1+\alpha)r^{\alpha}}{R^{1+\alpha}} dr$$
$$= \left(\frac{3d(u, v)}{R}\right)^{1+\alpha} + \frac{3(1+\alpha)d(u, v)}{R^{1+\alpha}} \int_{3d(u,v)}^{R} r^{\alpha-1} dr.$$

If $\alpha > 0$, the above expression is

$$(3\rho)^{1+\alpha} + \frac{3\rho(1+\alpha)}{\alpha R^{\alpha}} \left(R^{\alpha} - (3d(u, v))^{\alpha}\right) \leq \frac{3\rho(1+\alpha)}{\alpha},$$

where $\rho = \rho_{uv}$. If $\alpha = 0$, the probability is $3\rho + \log\left(\frac{1}{3\rho}\right) \leq 3\rho \log\left(\frac{1}{\rho}\right)$, completing the proof. □

To obtain the guarantee in Theorem 1.6, we modify Algorithm 4 as follows:

1. For constant $\alpha > 0$, we run Algorithm 1 with probability $1 - \alpha$ and Algorithm 4 with probability α. This yields $g(\rho) = \Omega(\alpha\rho^{1+\alpha})$ and $f(\rho) \leq 3\rho$.

2. When $\alpha = 0$, we run Algorithm 1 with probability $1 - \frac{1}{\log R}$ and otherwise run Algorithm 4. This again yields $g(\rho) = \Omega(\rho / \log R)$ and $f(\rho) = O(\rho)$.

We note that though Algorithm 4 produces $O(n/R)$ components in expectation, it does not satisfy (COMP) since when $r = o(R)$, the maximum diameter is $O(r) = o(R)$.

7. Empirical Results on Precinct Maps

In this section, we run Algorithm 2 on precinct data obtained from the MGGG Redistricting Lab (MGGG Lab). This is naturally modeled as a planar graph where the vertices correspond to precincts of a state, and edges denote adjacency of precincts. We ran experiments on the states of North Carolina (NC), Pennsylvania (PA), and Maryland[5] (MD). The sizes of the graphs are listed in Table 2.

State	Nodes	Edges	Diameter
North Carolina	2692	7593	68
Maryland	1809	4718	66
Pennsylvania	9255	25721	89

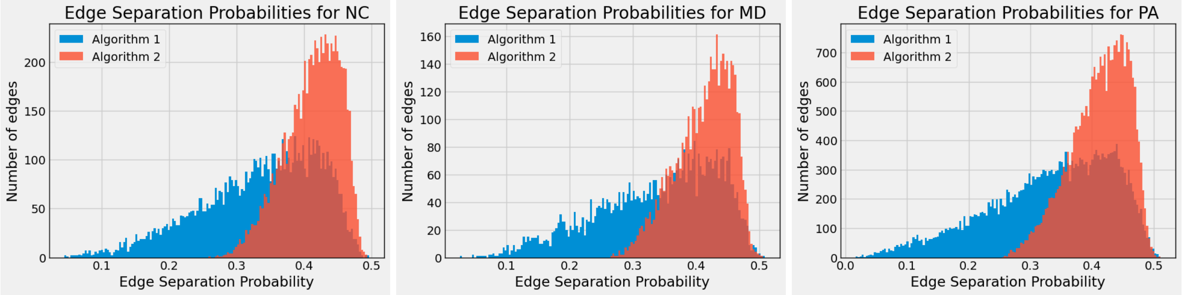
Table 2: Sizes of the graphs from MGGG Lab.

We implemented Algorithm 2 with parameter $R = 5$ and compared it with Algorithm 1 with the same R. We present the results for individual fairness, as well as the number and diameter of the clusters below. We omit a detailed runtime analysis of the algorithms as they exclusively entail Breadth-First Search traversals.

We focus on Algorithm 2 since the real-world graphs have relatively small distances, and Algorithms 3 and 4 provide weaker guarantees in this regime.

Individual Fairness. For Algorithm 2, for any edge $(u, v) \in E$, Theorem 1.3 yields $\Pr[(u, v) \text{ separated}] \in \left(\frac{1}{2R}, \frac{3}{R}\right) = (0.1, 0.6)$, which bounds individual fairness by a factor of 6 for this algorithm. Empirically, we find that Algorithm 2 outperforms Algorithm 1 significantly on this metric. We plot the histograms of the edge separation probabilities in Figure 2, showing clearly that Algorithm 2 is more individually fair than Algorithm 1. Additionally, we obtain a stronger guarantee in the lower bound at virtually no cost to the upper bound.

[5] The MD graph from (MGGG Lab) is disconnected. It was modified to connect precincts that shared a water border.



(a) (b) (c)

Figure 2: Histograms of edge separation probabilities across 3000 runs of Algorithm 2 on NC, MD, and PA.

In our experiments, we choose the roots for the BFS trees uniformly at random. Our results show that even with such a strengthened version of Algorithm 1, it is more unfair than Algorithm 2. This is in line with our result showing that randomness cannot help Algorithm 1 (see Appendix B.3).

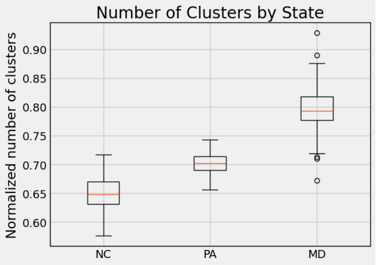


Figure 3: Box plot of the number of clusters across 200 runs of Algorithm 2 on NC, MD, and PA. The *normalized* number of clusters refers to the number of clusters as a fraction of the baseline $\frac{n}{R}$.

Number and diameter of clusters. For Algorithm 2, we can adapt Lemma 2.2 to show that the expected number of clusters is at most $\frac{6n}{R}$. We plot the number of clusters for 200 runs in Figure 3, and note that the empirically observed value is consistently smaller than n/R.

We next plot the maximum diameter of a cluster in each of the 200 runs of Algorithm 2 in Figure 4. Although the theoretical guarantee on the diameter of the largest cluster is $43R = 215$, as stated in Theorem 2.1, the largest diameter of any cluster is empirically much smaller.

8. Conclusion

We briefly mention some extensions. Analogous to Klein et al. (1993), our results naturally extend to graphs excluding a $K_{r,r}$ minor, with the bounds worsening as r increases. Similarly, for general metric spaces, we can plug existing low diameter decompositions into Algorithm 4 or Algorithm 5, losing an additional $O(\log n)$ factor in $f(\rho)$ compared to the bounds in Table 1; see Appendix C for such a result. It would be interesting to define analogs for (CON) and (COMP) for general metrics and study their trade-offs with individual fairness, analogous to our results for planar graphs in Sections 3 and 5. Exploring whether our bounds in Table 1 are improvable for planar graphs would also be an interesting direction. Our work makes edge separation probabilities proportional; defining some notion of Pareto-optimality in conjunction with this would be interesting.

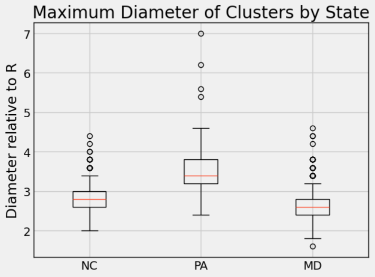


Figure 4: Box plot of the maximum diameters of clusters for the states of NC, MD, and PA across 200 clusterings. The diameters are shown as a fraction of R.

Our techniques can be viewed as randomly compressing a graph while preserving fairness among compressed pairs. This can be used as a pre-processing step to solve more complex optimization problems on the entire graph more efficiently. As an example, in Congressional redistricting, we may want to partition the graph into k parts each with an equal population, while preserving individual fairness. Though no worst-case fairness guarantees are possible for fairness in this setting, one can design efficient heuristics to optimize fairness for the compressed graph, and this is an avenue for future work. Similarly, exploring individual fairness for other optimization problems beyond graph clustering would be interesting.

Acknowledgment. This work is supported by NSF grant CCF-2113798.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Abraham, I., Gavoille, C., Gupta, A., Neiman, O., and Talwar, K. Cops, robbers, and threatening skeletons: Padded decomposition for minor-free graphs. *SIAM Journal on Computing*, 48(3):1120–1145, 2019. doi: 10.1137/17M1112406. URL https://doi.org/10.1137/17M1112406.

Altman, M. Modeling the effect of mandatory district compactness on partisan gerrymanders. *Political Geography*, 17(8):989–1012, 1998.

Ashlagi, I. and Shi, P. Improving community cohesion in school choice via correlated-lottery implementation. *Operations Research*, 62(6):1247–1264, 2014.

Backurs, A., Indyk, P., Onak, K., Schieber, B., Vakilian, A., and Wagner, T. Scalable fair clustering. In *International Conference on Machine Learning*, pp. 405–413. PMLR, 2019.

Bartal, Y. Probabilistic approximation of metric spaces and its algorithmic applications. In *Proceedings of 37th Conference on Foundations of Computer Science*, pp. 184–193, 1996. doi: 10.1109/SFCS.1996.548477.

Bera, S., Chakrabarty, D., Flores, N., and Negahbani, M. Fair algorithms for clustering. In Wallach, H., Larochelle, H., Beygelzimer, A., d Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/fc192b0c0d270dbf41870a63a8c76c2f-Paper.pdf.

Bourgain, J. On lipschitz embedding of finite metric spaces in hilbert space. *Israel Journal of Mathematics*, 52:46–52, 1985. URL https://api.semanticscholar.org/CorpusID:121649019.

Brubach, B., Chakrabarti, D., Dickerson, J., Khuller, S., Srinivasan, A., and Tsepenekas, L. A pairwise fair and community-preserving approach to k-center clustering. In *International Conference on Machine Learning*, pp. 1178–1189. PMLR, 2020.

Celis, L. E., Straszak, D., and Vishnoi, N. K. Ranking with fairness constraints. *arXiv preprint arXiv:1704.06840*, 2017.

Chawla, S. and Jagadeesan, M. Individual fairness in advertising auctions through inverse proportionality. In Braverman, M. (ed.), *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPIcs*, pp. 42:1–42:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

Chen, X., Fain, B., Lyu, L., and Munagala, K. Proportionally fair clustering. In *International Conference on Machine Learning*, pp. 1032–1041. PMLR, 2019.

Chierichetti, F., Kumar, R., Lattanzi, S., and Vassilvitskii, S. Fair clustering through fairlets. *Advances in Neural Information Processing Systems*, 30, 2017.

Chouldechova, A. and Roth, A. The frontiers of fairness in machine learning. *ArXiv*, abs/1810.08810, 2018.

Cohen-Addad, V., Klein, P. N., and Young, N. E. Balanced centroidal power diagrams for redistricting. In *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '18, pp. 389–396, New York, NY, USA, 2018. Association for Computing Machinery.

DeFord, D., Duchin, M., and Solomon, J. Recombination: A family of markov chains for redistricting. *Harvard Data Science Review*, 3(1), 3 2021.

Devic, S., Kempe, D., Sharan, V., and Korolova, A. Fairness in matching under uncertainty. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 7775–7794. PMLR, 2023.

Duchin, M. Gerrymandering metrics: How to measure? what's the baseline? *arXiv*, 1801.02064, 2018.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. S. Fairness through awareness. In *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*, pp. 214–226, 2012.

Esmaeili, S., Brubach, B., Tsepenekas, L., and Dickerson, J. Probabilistic fair clustering. *Advances in Neural Information Processing Systems*, 33:12743–12755, 2020.

Fakcharoenphol, J., Rao, S., and Talwar, K. A tight bound on approximating arbitrary metrics by tree metrics. *Journal of Computer and System Sciences*, 69(3):485–497, 2004.

Goemans, M. X. Lecture notes. https://math.mit.edu/˜goemans/18409-2006/lec9.pdf, 2006.

Harris, D. G., Li, S., Pensyl, T., Srinivasan, A., and Trinh, K. Approximation algorithms for stochastic clustering. *Journal of Machine Learning Research*, 20(153):1–33, 2019. URL http://jmlr.org/papers/v20/18-716.html.

Hochbaum, D. S. and Shmoys, D. B. A best possible heuristic for the k-center problem. *Mathematics of Operations Research*, 10(2):180–184, 1985. ISSN 0364765X, 15265471. URL http://www.jstor.org/stable/3689371.

Hsu, W.-L. and Nemhauser, G. L. Easy and hard bottleneck location problems. *Discrete Applied Mathematics*, 1(3):209–215, 1979. ISSN 0166-218X. doi: https://doi.org/10.1016/0166-218X(79)90044-1. URL https://www.sciencedirect.com/science/article/pii/0166218X79900441.

Huang, C.-C., Kavitha, T., Mehlhorn, K., and Michail, D. Fair matchings and related problems. *Algorithmica*, 74 (3):1184–1203, 2016.

Joseph, M., Kearns, M., Morgenstern, J., Neel, S., and Roth, A. Fair algorithms for infinite and contextual bandits. *arXiv preprint arXiv:1610.09559*, 2016.

Klein, P., Plotkin, S. A., and Rao, S. Excluded minors, network decomposition, and multicommodity flow. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '93, pp. 682–690, New York, NY, USA, 1993. Association for Computing Machinery. ISBN 0897915917. doi: 10.1145/167088.167261. URL https://doi.org/10.1145/167088.167261.

Matoušek, J. *Embedding Finite Metric Spaces into Normed Spaces*, pp. 355–400. Springer New York, New York, NY, 2002.

MGGG Lab. Mggg state shapefiles. https://github.com/mggg-states. Accessed January 2024.

Miller, G. L., Peng, R., and Xu, S. C. Parallel graph decompositions using random shifts. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '13, pp. 196–203, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450315722. doi: 10.1145/2486159.2486180. URL https://doi.org/10.1145/2486159.2486180.

Rao, S. Small distortion and volume preserving embeddings for planar and euclidean metrics. In *Proceedings of the Fifteenth Annual Symposium on Computational Geometry*, SCG '99, pp. 300–306, New York, NY, USA, 1999. Association for Computing Machinery. ISBN 1581130686. doi: 10.1145/304893.304983. URL https://doi.org/10.1145/304893.304983.

Shen, Z., Wang, Z., Zhu, X., Fain, B., and Munagala, K. Fairness in the assignment problem with uncertain priorities. In Agmon, N., An, B., Ricci, A., and Yeoh, W. (eds.), *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023, London, United Kingdom, 29 May 2023 - 2 June 2023*, pp. 188–196. ACM, 2023.

A. Omitted Proofs

A.1. Proof of Lemma 2.2

Let C_1, C_2, C_3 be the random variables denoting the additional number of components generated in phases $1, 2, 3$ respectively, and let $C = C_1 + C_2 + C_3$. Consider the first phase of Algorithm 1 and the associated BFS tree. We can pretend the edges not in the tree don't exist; this only decreases the number of components generated. Let $a_1, a_2, \ldots, a_q$ denote the number of edges in levels $1, 2, \ldots, q$ in the tree. We pad this sequence with zeros so that the q is a multiple of R. We have $\sum_{i=1}^{q} a_i = n - 1 \leq n$. If a level j is cut, the number of additional components created is precisely a_j. Since any level is chosen for cut with probability $1/R$, we have $\mathsf{E}[C_1] \leq n/R$.

Suppose we have components of size $m_1, m_2, \ldots, m_t$ at the end of the first phase. Applying the above argument again to each of these components, we have $\mathsf{E}[C_2|C_1] \leq \sum_{i=1}^{t} m_i/R \leq n/R$, since $\sum_{i=1}^{t} m_i \leq n$. A similar argument holds for $\mathsf{E}[C_3]$, completing the proof of the first part by linearity of expectation.

For the second part, suppose $R = D/\beta$. Then, the BFS tree in the first phase has a depth of at least $D/2$, so that at least $\lfloor \beta/4 \rfloor$ components have a diameter of at least $R/2$. For each of these components, the BFS tree now has a depth at least $R/2$, so at least one component from each has a diameter of at least $R/4$. Similarly for the third phase, where we obtain a diameter at least $R/8$, completing the proof.

A.2. Proof of Lemma 5.2

The proof is similar to that of Lemma 2.2. The same arguments follow for the first cut created. For the cuts in Step 3, we separately argue below. Let $a_1, a_2, \ldots, a_q$ denote the number of edges in levels $1, 2, \ldots, q$ of the tree. The expected number of additional components generated by the cut corresponding to k_1 is at most $\frac{\sum_{i=1}^{q} a_i}{R} \leq \frac{n}{R}$.

To bound the expected number of components generated by the cut corresponding to k_2, observe that the unconditional probability that any edge of the BFS tree is cut is again $1/R$. Formally, the expected number of components is:

$$\sum_{x=0}^{R-1} \Pr[k_1 = x] \sum_{i=1}^{q} a_i \Pr[k_2 \equiv x - i \mod R | k_1 = x]$$
$$= \sum_{i=1}^{q} a_i \sum_{x=0}^{R-1} \frac{1}{R} \Pr[k_2 \equiv x - i \mod R | k_1 = x] = \frac{\sum_{i=1}^{q} a_i}{R} \leq \frac{n}{R}$$

The second requirement of (COMP) follows similarly as the proof of Lemma 2.2, with the observation that we only have $\frac{2D}{R}$ cuts per phase in total for any outcome of randomness.

B. Lower Bound for KPR: Proof of Theorem 1.2

B.1. Proof of Theorem 1.2 for $d = 1$

We start with the case of $d = 1$ in Figure 5. We present a constant-sized subgraph, but the arguments remain the same if we attach the graph to an arbitrary star graph with root r_0. We also consider adversarial choices of roots in Algorithm 1, but we show later that a similar argument works for random choices of roots. Let r_0 be the choice of the root. In all subsequent subgraphs, if r_0 is not present then r_1 is chosen as the root. If neither r_0 nor r_1 are present, r_2 is chosen as the root.

Let G_0 be the initial graph and let T_0 be the BFS tree from r_0. The first observation is that u, v, r_1, r_2 belong to the same level of T_0. Thus, no cut can separate these vertices. The only cut that modifies the graph cuts just below r_0. For the next phase, the component containing u, v is a 4-cycle with a diagonal joining u, v. Let this graph be G_1.

In G_1, r_1 is chosen as the new root. The BFS tree T_1 from r_1 consists of r_1 at level 0, u, v at level 1, and r_2 at level 2. Observe that u, v cannot be separated in G_1 because they are again at the same level.

Subsequently, since $R > 2$, we only have two possible cuts: A cut that separates level 0 from levels 1 and 2, and a cut that separates level 2 from levels 0 and 1. In either case, the component containing u, v is a 3-cycle with u, v, and one of r_1, r_2. Without loss of generality, we consider the latter case so that we get a graph G_2 in which we choose r_2 as the final root. The

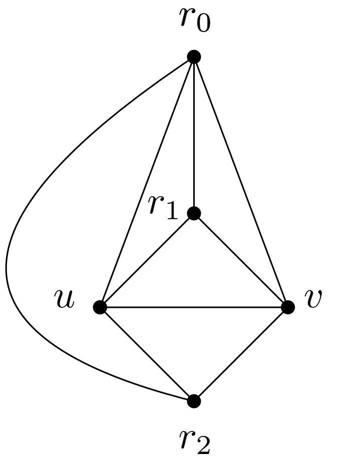


Figure 5: Counterexample for $d = 1$.

BFS tree from r_2 in G_2 again has u, v at the same level. Hence, u and v are never separated.

B.2. Proof of Theorem 1.2 for $d > 1$

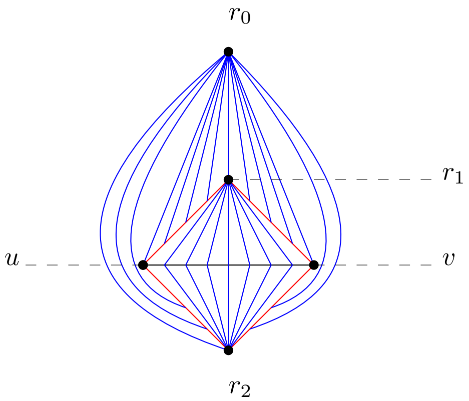


Figure 6: Counterexample for $d \geq 2$. Each blue line is a path of length d.

We now prove the statement for general $d \geq 2$. We describe the construction of the graph we use in Figure 6.

- Each blue line in the graph is a path of length d.
- The straight line path between u, v that does not go through r_1, r_2 is of length d. Each vertex in this path is connected to r_1, r_2 via paths of length d.
- u, v, r_1, r_2 are all at distance d from r_0.
- The red paths between u, v and r_1, r_2 are paths of length d. Each vertex in this path is connected to r_0 via a path of length d. This is possible to do in a planar graph, as shown in the figure.

The construction contains $\leq 6d^2 + d \leq 9d^2$ vertices. We can arbitrarily attach a star graph with root r_0 to increase the number of vertices. However, the number of vertices n must satisfy $n \geq 9d^2$.

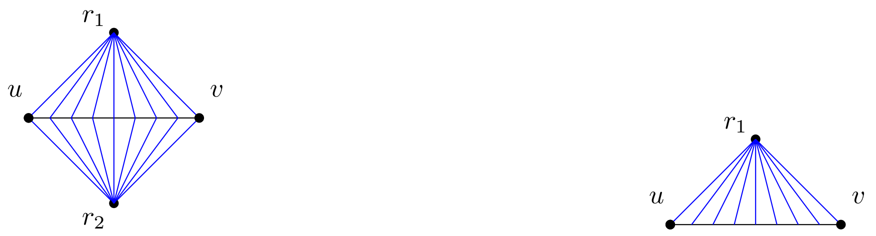
Let r_0 be the choice of the root. In all subsequent subgraphs, if r_0 is not present then r_1 is chosen as the root. If neither r_0 nor r_1 are present, r_2 is chosen as the root.

Let G_0 be the initial graph and let T_0 be the BFS Tree from r_0. Observe that by construction, all the vertices on the red paths at level d in T_0. Thus, they must all be in the same subgraph as u, v for the second phase. Let G_1 be the component containing u, v in the second phase. We argue that G_1 is roughly of two 'types', depending on where the cut is made in the first phase:

Case 1. The cut is made above level d. This ensures that all vertices at levels $\geq d$ are in G_1. Let the resulting graph be G_1^a. Note that $r_0 \notin V(G_1^a)$.

Case 2. The cut is made below level d. This ensures that all vertices at levels $\leq d$ are in G_1. Let the resulting graph be G_1^b.

Note that the T_0 has at most $2d$ levels since all vertices of the graph are within $2d$ levels of r_0. Furthermore, no cut can remove a vertex both below level d and above level d from G_1 because $R > 2d$. We now consider these two cases, ignoring the case where the graph is not modified because that can only decrease the probability that u, v are separated. The next two lemmas complete the proof of Theorem 1.2.



(a) Graph G_1^a.

(b) Graph G_2^a.

Figure 7: Case 1 in the proof of Theorem 1.2.

Lemma B.1. *u and v cannot be separated by the final two iterations ($i = 2, 3$) of Algorithm 1 run on G_1^a.*

Proof. In this case, the relevant part of the graph is shown in Figure 7a. Any vertex not shown is a vertex above level d in G_0, and does not lie in any path from r_1 to any of the vertices in G_1^a. We ignore them for ease of exposition, but it can be verified that they do not affect our arguments, specifically because they do not affect the positions of u and v in the BFS trees or their connectivity.

Now, r_1 is chosen as the root. Let G_2^a be the component containing u, v in the next phase. Like before, only two kinds of cuts can modify the graph.

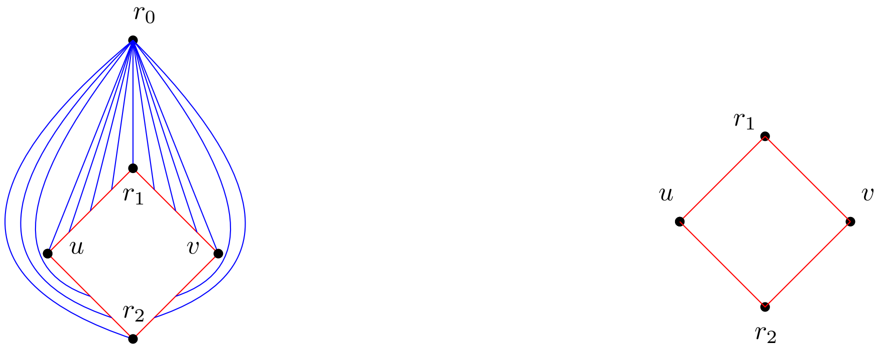
1. If the cut is made above level d, r_2 lies in G_2^a and r_1 is not in G_2^a.
2. If the cut is made below level d, r_1 lies in G_2^a and r_2 is not in G_2^a.

Because of symmetry, we can assume wlog that the cut is made below level d. Regardless of where the cut is, the vertices and edges in Figure 7b are preserved in G_2^a. There may be some extra vertices that can safely be ignored again. We again choose r_1 as the root. Observe that every vertex in the straight-line path (colored red) from u to v is at distance d from r_1. Thus, no cut can separate the vertices in this path, and u, v must necessarily remain connected after any cut. □

Lemma B.2. *u and v cannot be separated by the final two iterations ($i = 2, 3$) of Algorithm 1 run on G_2^a.*

Proof. The relevant part of the graph for phase 2 is of the form in Figure 8a. There are several paths from r_1 or r_2 to u or v that may or may not be present in G_1^b but are not pictured. For our arguments, we only need to observe that the *shortest* paths from r_1 and r_2 to u and v are preserved in G_1^b.

In G_1^b, making cuts below level d does not affect any of the relevant u, v paths. Thus, we assume that the cut is made above level d, giving us the graph G_2^b. The relevant vertices are pictured in Figure 8b. Like before, we can restrict ourselves to this graph because the remaining edges do not affect the BFS trees that we construct. For the next phase, we choose r_1 as the root. The key observation is that there is a path from u to v entirely above level d in this tree through r_1, and there is a disjoint path from u to v entirely below level d through r_2. We now reuse the observation that any cut can either remove vertices below level d or vertices above level d, but not both. Thus, u and v remain connected regardless of the cut. □



(a) Graph G_1^b.

(b) Graph G_2^b.

Figure 8: Case 2 in the proof of Theorem 1.2.

B.3. Additional Impossibility Results

Algorithm 1 with additional iteration. The proof of Theorem 1.2 describes a pair u, v that remain connected after 3 phases of KPR. One might wonder if it is possible to do better with additional phases. Indeed, it turns out that we can obtain a non-zero probability of separation. This forms the basis of Algorithm 3. However, we also observe that simple modifications to Algorithm 1 can only give a very weak lower bound on the probability of separation.

Let KPR$_+$ denote Algorithm 1 with one additional phase. That is, we add one more iteration to step 1.

Corollary B.3. *For any $n \in \mathbb{N}$ and $1 \leq d \leq \frac{\sqrt{n}}{3}$, there exists a planar graph G_d and vertices $u, v \in V(G)$ with $d(u, v) = d$ such that KPR$_+$ with adversarial choices of roots in each component and parameter $R > 2d$, separates u, v with probability at most $8\rho_{uv}^4$.*

Proof. By running through the proof of Theorem 1.2 again, we can see that if, in any of the phases, the cut is made beyond the first $2d$ levels of the BFS Tree, then u, v will not be separated even in 4 phases. For example, if in the first phase, the cut is made beyond level $2d$ of the BFS Tree, then u, v cannot be separated in the remaining three phases because of Theorem 1.2. Similarly, if this happens in the second or third phases, u, v will never be separated in the remaining phases.

Thus, we assume that for the first three phases, the cut is within the first $2d$ levels of the BFS tree. This happens with probability $\left(\frac{2d}{R}\right)^3 = 8\rho_{uv}^3$. The final observation is that u, v are connected by a path of length d in the fourth phase regardless of whether the graph in the third phase is G_2^a or G_2^b. Thus, they are at most d levels apart in any BFS tree and can be separated by probability at most ρ_{uv}. This gives us an overall probability of separation of at most $8\rho_{uv}^4$. □

Random choices of roots. The original algorithm of (Klein et al., 1993) chooses the nodes arbitrarily. A natural question is whether considering adversarial choices of roots is too pessimistic and whether the 'bad' case in Theorem 1.2 can be avoided by a simple scheme such as choosing the root uniformly at random.

Unfortunately, randomization does not provide any strength beyond possibly providing a $\frac{1}{\mathsf{poly}(n)}$ lower bound. Recall that in the proof of Theorem 1.2, all the adversary needs is to enforce a preference ordering between the roots: If r_0 is present in a component, it must be chosen as the root. If r_0 is not present but r_1 is, then it must be chosen. Finally, if neither r_0 nor r_1 is present, r_2 should be chosen as the root. We argue that such a preference order can be enforced even in the presence of uniform sampling of roots.

Corollary B.4. *For any $n \in \mathbb{N}$ and $1 \leq d \leq n^{\frac{1}{8}}$, there exists a planar graph G_d and vertices $u, v \in V(G)$ with $d(u, v) = d$ such that Algorithm 1 with uniformly random choices of roots in each component and parameter $R > 2d$ separates u, v with probability only $O\left(n^{-\frac{1}{4}}\right)$.*

Proof. We take the construction in Theorem 1.2. Suppose that it has m vertices. We add star graphs with m^4, m^3, and m^2 nodes with roots r_0, r_1, r_2 respectively. Let the total number of nodes now be $n = O(m^4)$. Note that choosing a leaf attached to r_i is equivalent to choosing r_i for the purposes of the proof since the BFS tree from that node is similar to that of the BFS tree from r_i.

We now follow the proof of Theorem 1.2 and argue that the roots are chosen as required with high probability. In the first phase, the probability that r_0 or a leaf attached to it is not chosen is $\frac{m^3+m^2+m}{m^4+m^3+m^2+m} \le \frac{1}{m}$. In the second phase, if the graph was not modified in the first phase, the same calculation above applies. Similarly, in G_1^b, the probability that r_0 is not chosen is again at most $\frac{1}{m}$. In G_1^a, the probability that r_1 is not chosen is at most $\frac{m^2+m}{m^3+m^2} \le \frac{1}{m}$. A similar calculation can be done for both graphs G_2^a and G_2^b in the third phase. By union bounds, the proof is complete. □

C. Individual Fairness without (CON): Proof of Theorem 1.7

We now prove Theorem 1.7. Recall that an embedding is a function h that maps the vertices v of G to s-dimensional points $h(v)$ (for some s) with the guarantee that the ℓ_1 distance between $h(u)$ and $h(v)$ is at least $d(u,v)$, and at most $\alpha \cdot d(u,v)$. The parameter $\alpha \ge 1$ is termed the *distortion*. The algorithm for embedding planar graphs into ℓ_1 (Rao, 1999) has the following guarantee:

Lemma C.1 ((Rao, 1999)). *Given a planar graph with diameter Δ, there is an embedding h into Euclidean space such that for any pair of vertices (u,v):*

$$d(u,v) \le \|h(u)-h(v)\|_2 \le O(\sqrt{\log \Delta}) \cdot d(u,v).$$

Algorithm 5 LDD via Euclidean Embedding

```
Input: Integer $R$, planar graph $G=(V,E)$
Run Algorithm 1 to obtain clusters of diameter $O(R)$.
for every connected component $C$ found do
    Embed $C$ into a Euclidean metric via the algorithm in (Rao, 1999). Let $h(v)$ be the embedding of $v \in V$ and let the dimension be $\gamma$.
    Let $\vec{x}$ be a $\gamma$-dimensional vector whose coordinates are i.i.d. $\mathcal{N}(0,1)$.
    Let $q(v) = h(v) \cdot \vec{x}$.
    Choose $\theta \in [0, R \cdot \sqrt{\log R}]$ uniformly at random.
    For $\ell \in \mathbb{Z}$, place all vertices $v$ s.t. $q(v) \in [\theta + \ell \cdot R\sqrt{\log R}, \theta + (\ell+1)R\sqrt{\log R})$ in sub-partition $C(\ell)$.
    Output the set $\{C(\ell)\}$ of sub-partitions.
end for
```

The algorithm for showing Theorem 1.7 is shown in Algorithm 5. Let $d'(u,v) = \|h(u)-h(v)\|_2$. Note that $\mathsf{E}[|q(v)-q(u)|] = d'(u,v)$, so that

$$\Pr[(u,v) \text{ are separated in Algorithm 5 }] = \frac{d'(u,v)}{R\sqrt{\log R}}.$$

Observe that (u,v) are separated in Algorithm 5 with probability $O(\rho_{uv})$ or by using Lemma C.1 and observing that $\Delta = O(R)$, in Algorithm 5 with probability $O(\rho_{uv})$. This shows $f(\rho) = O(\rho)$. The lower bound follows by observing that (u,v) are separated in Algorithm 5 with probability at least $\rho/\sqrt{\log R}$ by the above equation. The algorithm does not satisfy (CON) since any two vertices have a non-zero probability of being in the same cluster because of the unbounded support of the normal distribution. The algorithm does not satisfy (COMP) either since, for any graph, there is a non-zero probability of separating the vertices in n clusters. The proof of Theorem 1.7 now follows.

General Metrics. The above algorithm also extends to general metric spaces if we replace Algorithm 5 with the low diameter decomposition procedure in (Fakcharoenphol et al., 2004; Miller et al., 2013). This guarantees probability of separation $f(\rho) = O(\rho \cdot \log n)$. Now we embed the metric into Euclidean space suffering distortion $O(\log n)$ (Bourgain, 1985), and finally choose $\theta \in [0,R]$ at random before performing Algorithm 5. This guarantees $g(\rho) = \rho$ and $f(\rho) = O(\rho \cdot \log n)$.

D. Completing Proof of Theorem 5.3: Proof of Lemma 5.5

At a high level, our proof outline is similar to that for showing the bounded diameter of clusters in Algorithm 1 in (Goemans, 2006), where the overall goal is to show a proof by contradiction by exhibiting a $K_{3,3}$ minor. We also borrow some of their

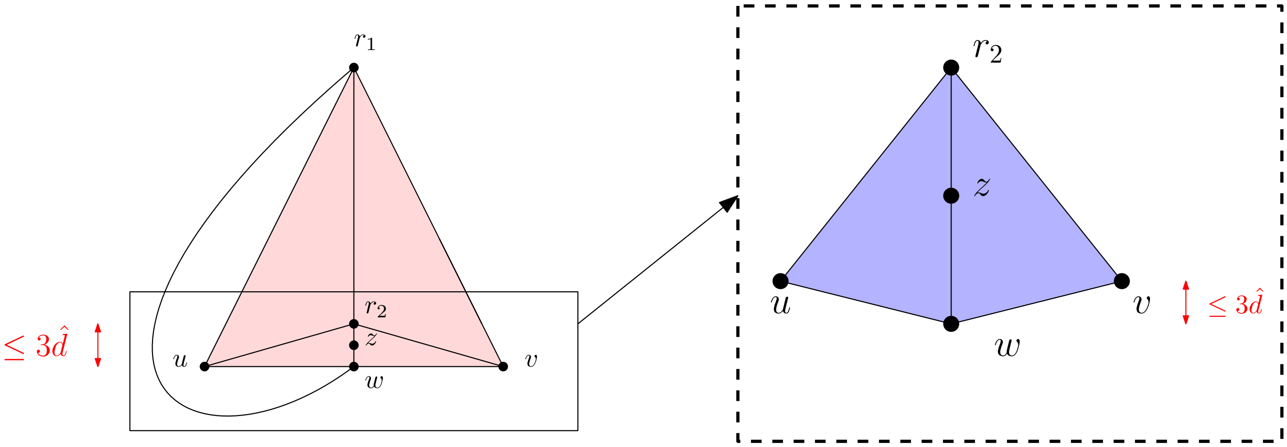


Figure 9: The relative positions of the special vertices in T_1 (red) and T_2 (blue inset). All of u, v, w, z, r_2 are within $3\hat{d}$ levels of each other in T_1 by virtue of them being in G_2. In T_2 , u, v, w are within $3\hat{d}$ levels of each other by virtue of them being in G_3.

notation, though the meaning of the notation is slightly different in our context, as are the details of the construction and analysis. For instance, our supernodes that form the $K_{3,3}$ minor are easily defined with respect to a few special vertices, as opposed to those in (Klein et al., 1993), which involve many auxiliary nodes. We also only need to look at two BFS trees instead of three, as in (Klein et al., 1993), although we make more cuts.

Another important distinction is that our analysis inherently requires the randomness of Algorithm 3. For Algorithm 1, the proof that the clusters have a small diameter only hinges on cutting the BFS tree into slices of width R. In our case, the trees we analyze possess the necessary properties only if we condition on the events E_1, E_2, L_1 and L_2.

D.1. Special Vertices

We first show some simple properties of the graphs G_1, G_2, G_3. The first claim follows from the definition of the BFS tree T_1, and the occurrence of event $E_1 \cap E_2 \cap L_1 \cap L_2$.

Claim D.1. *For any $x \in V(G_2)$, $|\ell_1(x) - \ell_1(u)|$ and $|\ell_1(x) - \ell_1(v)|$ are both at most $2\hat{d}$.*

Similarly, the next claim follows since L_2 happens and u, v are still connected in G_3.

Claim D.2. *For any $x \in V(G_3)$, $|\ell_2(x) - \ell_2(u)|$ and $|\ell_2(x) - \ell_2(v)|$ are at most $2\hat{d}$.*

Since the events L_1 and L_2 happen, and since $d_i(u, v) \geq d$, and we defined $\hat{d} = \frac{d}{59}$, we have:

Claim D.3. *For $i \in \{1, 2\}$, we have: $\ell_i(u)$ and $\ell_i(v)$ are at least $28\hat{d}$.*

We now find two special vertices, w and z. Let P be the shortest path between u, v in G_3. Let $w \in P$ be such that $d_2(u, w) \geq \frac{d}{2} - 1 > 28\hat{d}$ and $d_2(v, w) \geq \frac{d}{2} - 1 > 28\hat{d}$. By Claim D.2 and Claim D.3, we have $\ell_2(w) \geq 25\hat{d}$. Consider the path from r_2 to w in T_2. Let z be a vertex at distance $9\hat{d}$ from w along this path. Note that z does not belong to G_3. Since $\ell_2(w) \geq 25\hat{d}$, we know that z exists. The construction of these vertices is illustrated in Figure 9.

Using Claim D.2 with $x = w$ and noting that $\ell_2(z) = \ell_2(w) - 9\hat{d}$, we obtain:

Claim D.4. *$|\ell_2(z) - \ell_2(u)|$ and $|\ell_2(z) - \ell_2(v)|$ are at most $11\hat{d}$.*

D.2. Construction of $K_{3,3}$ Minor

First, the following is a straightforward property of BFS trees that we will use repeatedly.

Claim D.5. *For $i \in \{1, 2\}$, consider a shortest path Q in G_i from r_i to any vertex x. For any $x' \in Q$, if both $x, x' \in G_{i+1}$, then every vertex between x, x' in Q also lies in G_{i+1}.*

In the narrative below, we assume $x \in \{u, v, z\}$. Define $P_1(x)$ as the shortest path from r_1 to x in G_1. Further, define

$$P_1^-(x) = \{x' \in P_1(x) \mid x' \in G_1 \setminus G_2\}$$
$$P_1^+(x) = \{x' \in P_1(x) \mid x' \in G_2\}.$$

Using Claim D.5, for $x \in \{u, v, z\}$, $P_1^-(x)$ and $P_1^+(x)$ are a partition of $P_1(x)$ into two subpaths. Similarly, for $x \in \{u, v, z\}$, we define $P_2(x)$ as the shortest path from r_2 to x in G_2. Let

$$P_2^-(x) = \{x' \in P_2(x) \mid \ell_2(x') < \ell_2(x) - 4\hat{d}\}$$
$$P_2^+(x) = \{x' \in P_2(x) \mid \ell_2(x') \geq \ell_2(x) - 4\hat{d}\}.$$

That is, we partition the path into two subpaths based on the distance from r_2. The $4\hat{d}$ vertices closest to x along the path lie in $P_2^+(x)$. Again, by Claim D.5, these form sub-paths of $P_2(x)$.

Recall that we had assumed for contradiction the existence of a path P from u to v that survived $E_1 \cap E_2$, and that passes through w. For $y \in \{u, v\}$, we define $P_w(y)$ as the subpath of P from w to x in G_2. Define $P_w(z)$ as the path from w to z in T_2. Recall that this is part of the shortest path from r_2 to w in G_2.

We again partition the first $4\hat{d}$ vertices of these three paths from $x \in \{u, v, z\}$ as $P_w^+(x)$ and the remaining part as $P_w^-(x)$. Since $P_w(u)$ and $P_w(v)$ are shortest paths in G_3, we have

$$P_w^-(x) = \{x' \in P_w(x) \mid d_3(x', w) < d_3(x, w) - 4\hat{d}\}$$
$$P_w^+(x) = \{x' \in P_w(x) \mid d_3(x', w) \geq d_3(x, w) - 4\hat{d}\}.$$

for $x \in \{u, v\}$. Similarly, since $P_w(z)$ is a shortest path in G_2, we have

$$P_w^-(z) = \{x' \in P_w(z) \mid d_2(x', w) < d_2(z, w) - 4\hat{d}\}$$
$$P_w^+(z) = \{x' \in P_w(z) \mid d_2(x', w) \geq d_2(z, w) - 4\hat{d}\}.$$

We have the following useful properties of these partitions.

Claim D.6. *For $x \in \{u, v, z\}$ and every $y \in \{1, 2, w\}$, every vertex in $P_y^+(x)$ is within distance $4\hat{d}$ of x in G_2.*

Proof. This is true by definition for $P_2^+(x)$. For $P_3^+(x)$, this also follows from the definition because the distance between pairs can only increase from G_2 to G_3. Consider some vertex $x' \in P_1^+(x)$. Applying Claim D.1 on both x' and x, we get that $|\ell_1(x') - \ell_1(x)| \leq 4\hat{d}$. But since $P_1^+(x)$ is a shortest path from r_1 to x, this means that the path from x to x' can be at most $4\hat{d}$ long since $\ell_1(x')$ increases linearly along the path from r_1 to x. Since this entire path survives in G_2 (from Claim D.5), it must also mean that the distance between x' and x in G_2 must also be at most $4\hat{d}$. □

We finally describe the $K_{3,3}$ minor. We set

- $A_1 = P_1^-(u) \cup P_1^-(v) \cup P_1^-(z)$
- $A_2 = P_2^-(u) \cup P_2^-(v) \cup P_2^-(z)$
- $A_3 = P_w^-(u) \cup P_w^-(v) \cup P_w^-(z)$
- $A_4 = P_1^+(u) \cup P_2^+(u) \cup P_w^+(u)$
- $A_5 = P_1^+(v) \cup P_2^+(v) \cup P_w^+(v)$
- $A_6 = P_1^+(z) \cup P_2^+(z) \cup P_w^+(z)$

The following lemma will complete the proof of Lemma 5.5.

Lemma D.7. *After contracting each A_i, $(\{A_1, A_2, A_3\}, \{A_4, A_5, A_6\})$ form a $K_{3,3}$ graph.*

D.3. Proof of Lemma D.7

The rest of the section is devoted to proving this lemma. First, it can be seen that each A_i is a connected subgraph. It is also clear that the required edges for the contracted graph to be a $K_{3,3}$ are present. We can think of representing the supernodes A_1 through A_6 by the vertices r_1, r_2, w, u, v, and z respectively. The path between r_1 and u is partitioned between A_1 and A_4, and hence, the supernodes have an edge between them. Similarly, the path between r_2 and u is partitioned between A_2 and A_4, and the path between w and u is partitioned between A_3 and A_4.

It remains to be argued that each A_i is disjoint from the others. The next set of claims will show that, hence completing the proof of Lemma D.7.

Claim D.8. *A_1 is disjoint from every other A_j.*

Proof. This follows because A_1 is the only set that lies in $G_1 \setminus G_2$. The others all lie in G_2. □

Claim D.9. *A_4, A_5, A_6 are all mutually disjoint.*

Proof. Assume for a contradiction that there is some $x \in A_4 \cap A_5$. Then $d_2(u,v) \leq d_2(u,x) + d_2(x,v) \leq 8\hat{d}$ from Claim D.6 which is a contradiction since $d_2(u,v) \geq d$. Instead, if there is some $x \in A_4 \cap A_6$, then we have $d_2(u,w) \leq d_2(u,x) + d_2(x,z) + d_2(z,w) \leq 13\hat{d}$ which is a contradiction since $d_2(u,w) \geq \frac{d}{2}$. Similarly if there is some $x \in A_5 \cap A_6$. □

Claim D.10. *A_2 is disjoint from A_4, A_5, A_6.*

Proof. We first prove this for A_4. The proof is symmetric for A_5. Assume for a contradiction that there is some $x \in A_2 \cap A_4$. We have 3 cases depending on what part of A_2 that x lies in.

- $x \in P_2^-(u)$. $P_2^-(u)$ is disjoint from $P_2^+(u)$ by definition so x cannot belong to it. By definition of $P_2^-(u)$ and because $P_2(u)$ is a shortest path from r_2 to u, we have $d_2(x,u) > 4\hat{d}$. However, this means that it cannot belong to $P_1^+(u)$ or $P_w^+(u)$ because it would contradict Claim D.6.
- $x \in P_2^-(v)$. Since $x \in A_4$, Claim D.6 implies that $d_2(x,u) \leq 4\hat{d}$ which implies $|\ell_2(x) - \ell_2(u)| \leq 4\hat{d}$. But since event L_2 happened, this then implies that $|\ell_2(x) - \ell_2(v)| \leq 5\hat{d}$. Since x lies on the shortest path from r_2 to v, this implies that $d_2(x,v) \leq 5\hat{d}$. Thus, we have $d_2(u,v) \leq d_2(x,v) + d_2(x,u) \leq 9\hat{d}$ which is a contradiction.
- $x \in P_2^-(z)$. This is similar to the case of $P_2^-(v)$. Since $x \in A_4$, Claim D.6 implies that $d_2(x,u) \leq 4\hat{d}$ which also implies $|\ell_2(x) - \ell_2(u)| \leq 4\hat{d}$. But Claim D.4 then implies that $|\ell_2(x) - \ell_2(z)| \leq 15\hat{d}$. Since x lies on the shortest path from r_2 to z, this implies that $d_2(x,z) \leq 15\hat{d}$. Thus, we have $d_2(u,w) \leq d_2(x,z) + d_2(x,u) \leq 19\hat{d}$ which is a contradiction since we defined w with $d_2(u,w) \geq 27\hat{d}$.

We now prove it for A_6. The proof is similar. Assume for a contradiction that $x \in A_2 \cap A_6$. We again have 3 cases depending on what part of A_2 that x lies in.

- $x \in P_2^-(z)$ is disjoint from $P_2^+(z)$ by definition. By definition of $P_2^-(z)$ and because $P_2(z)$ is a shortest path from r_2 to z, we have $d_2(x,z) > 4\hat{d}$. However, this means that it cannot belong to $P_1^+(z)$ or $P_w^+(z)$ because it would contradict Claim D.6.
- $x \in P_2^-(v)$. Since it also belongs to A_6, Claim D.6 implies that $d_2(x,z) \leq 4\hat{d}$ which also implies $|\ell_2(x) - \ell_2(z)| \leq 4\hat{d}$. But Claim D.4 then implies that $|\ell_2(x) - \ell_2(v)| \leq 15\hat{d}$. Since x lies on the shortest path from r_2 to v, this implies that $d_2(x,v) \leq 15\hat{d}$. Thus, we have $d_2(w,v) \leq d_2(x,v) + d_2(x,z) + d_2(z,w) \leq 28\hat{d}$ which is a contradiction to $d_2(u,w) > 28\hat{d}$.
- $x \in P_2^-(u)$. This is symmetric to the case of $x \in P_2^-(v)$.

This completes the proof. □

Claim D.11. *A_2 is disjoint from A_3.*

Proof. This primarily follows by observing the levels of the vertices in G_2. Assume for a contradiction that there is some $x \in A_2 \cap A_3$. We have 3 cases depending on what part of A_2 that x lies in.

- $x \in P_2^-(z)$. Note that x cannot also lie in $P_w^-(z)$ because $P_2(z)$ and $P_w(z)$ are partitions of $P_2(w)$. We also have $\ell_2(x) < \ell_2(z) - 4\hat{d} = \ell_2(w) - 13\hat{d}$ by definition. By applying Claim D.2 on w and combining this with the previous inequality, we get $\ell_2(x) < \ell_2(u) - 11\hat{d}$. This implies that x is not in G_3. If x also lied in $P_w^-(u)$ or $P_w^-(v)$, this would contradict Claim D.2 because $P_w(u)$ and $P_w(v)$ are paths in G_3.
- $x \in P_2^-(u)$. By definition, we have $\ell_2(x) < \ell_2(u) - 4\hat{d}$. If x also lied in $P_w^-(u)$ or $P_w^-(v)$, this would contradict Claim D.2 because $P_w(u)$ and $P_w(v)$ are paths in G_3.

  If $x \in P_w^-(z)$, then $d_2(x, w) \le 9\hat{d}$ because $P_w(z)$ is only that long. Since $P_w(z)$ is part of the shortest path from r_2 to w, we also have $|\ell_2(x) - \ell_2(w)| \le 9\hat{d}$. Applying Claim D.2 on w, we get $|\ell_2(x) - \ell_2(u)| \le 11\hat{d}$. But since x also lies on the shortest path from r_2 to u, we also get $d_2(x, u) \le 11\hat{d}$. Thus, we get $d_2(w, u) \le d_2(w, x) + d_2(x, u) \le 20\hat{d}$ which is a contradiction.
- $x \in P_2^-(v)$ follows similarly by symmetry.

□

Claim D.12. *A_3 is disjoint from A_4, A_5, A_6.*

Proof. We first show that A_3 is disjoint from A_4. The proof that it is disjoint from A_5 follows by symmetry. Assume for a contradiction that there is some $x \in A_3 \cap A_4$. We have 3 cases depending on what part of A_3 that x lies in.

- $x \in P_w^-(u)$. By definition, it is disjoint from $P_w^+(u)$. Suppose that x also lied in $P_1^+(u)$ or $P_2^+(u)$. This means that $x \in V(G_3)$. By Claim D.5, this means that every vertex between x and u in $P_1^+(u)$ (or $P_2^+(u)$) also lies in G_3 and hence, $d_3(u, x) \le 4\hat{d}$ by Claim D.6. This is a contradiction because $x \in P_w^-(u)$ implies that $d_3(x, u) > 4\hat{d}$ by definition.
- $x \in P_w^-(v)$. It must be disjoint from $P_w^+(u)$ because they are both disjoint subsets of the path $P_w(u) \cup P_w(v)$ from u to v. Suppose x also lied in $P_1^+(u)$ or $P_2^+(u)$. By Claim D.5, this means that every vertex between x and u in $P_1^+(u)$ (or $P_2^+(u)$) also lies in G_3 and hence, $d_3(u, x) \le 4\hat{d}$ by Claim D.6. This contradicts the fact that $P_w(u) \cup P_w(v)$ was a shortest path in G_3 because the path $u \to x \to v$ is strictly shorter, because $d_3(u, x) \le 4\hat{d} < \frac{d}{2} \le |P_w(u)|$ and $x \to v$ is a subpath of $P_w(v)$.
- $x \in P_w^-(z)$. Suppose x also lied in A_4. By Claim D.6, $d_2(u, x) \le 4\hat{d}$. Thus, we get $d_2(u, w) \le d_2(u, x) + d_w(x, w) \le 13\hat{d}$ since x lies on $P_w(z)$ which has length $9\hat{d}$. This is a contradiction.

Now, we show that A_3 is disjoint from A_6. Again, suppose there is some $x \in A_3 \cap A_6$. We have 3 cases.

- $x \in P_w^-(z)$. By definition, it is disjoint from $P_w^+(z)$. We have $d_2(x, z) > 4\hat{d}$ by definition. If x also lied in $P_1^+(z)$ or $P_2^+(z)$, then this contradicts Claim D.6.
- $x \in P_w^-(u)$. If $x \in P_w^-(u)$, then $x \in V(G_3)$ and hence Claim D.2 implies that $|\ell_2(x) - \ell_2(u)| \le 2\hat{d}$. But if $x \in A_6$, then Claim D.6 implies that $d_2(x, z) \le 4\hat{d} \implies |\ell_2(z) - \ell_2(x)| \le 4\hat{d}$. We also know that $\ell_2(z) = \ell_2(w) - 9\hat{d}$. Thus, it must be $|\ell_2(w) - \ell_2(x)| \ge 5\hat{d}$. Applying Claim D.2 on w gives us $|\ell_2(x) - \ell_2(w)| \ge 3\hat{d}$. This contradicts the earlier inequality.
- $x \in P_w^-(v)$ follows by symmetry.

This completes the proof. □

These claims complete the proof of Lemma D.7, hence that of Lemma 5.5, and hence Theorem 5.3.