A Constant Factor Approximation for Navigating Through Connected Obstacles in the Plane

Neeraj Kumar* Daniel Lokshtanov* Saket Saurabh† Subhash Suri*

August 18, 2020

Abstract

Given two points s and t in the plane and a set of obstacles defined by closed curves, what is the minimum number of obstacles touched by a path connecting s and t? This is a fundamental and well-studied problem arising naturally in computational geometry, graph theory (under the names Min-Color Path and Minimum Label Path), wireless sensor networks (Barrier Resilience) and motion planning (Minimum Constraint Removal). It remains NP-hard even for very simple-shaped obstacles such as unit-length line segments. In this paper we give the first constant factor approximation algorithm for this problem, resolving an open problem of [Chan and Kirkpatrick, Theoretical Computer Science, 2014] and [Bandyapadhyay et al., Computational Geometry, 2020]. We also obtain a constant factor approximation for the Minimum Color Prize Collecting Steiner Forest where the goal is to connect multiple request pairs $(s_1, t_1), \ldots, (s_k, t_k)$ while minimizing the number of obstacles touched by any (s_i, t_i) path plus a fixed cost of w_i for each pair (s_i, t_i) left disconnected. This generalizes the classic Steiner Forest and Prize-Collecting Steiner Forest problems on planar graphs, for which intricate PTASes are known. In contrast, no PTAS is possible for Min-Color Path even on planar graphs since the problem is known to be APX-hard [Eiben and Kanj, ACM Transactions on Algorithms, 2020]. Additionally, we show that generalizations of the problem to disconnected obstacles in the plane or connected obstacles in higher dimensions are strongly inapproximable assuming some well-known hardness conjectures.

*Department of Computer Science, University of California, Santa Barbara, USA {neeraj@cs.ucsb.edu, daniello@ucsb.edu, suri@cs.ucsb.edu}

†Institute of Mathematical Sciences, Chennai, India and Department of Informatics, University of Bergen, Norway. saket@imsc.res.in

1 Introduction

We consider the following generalization of the undirected, unweighted s-t-Shortest Path problem, called Min-Color Path. Given an undirected graph $G = (V, E)$ with a coloring function $\sigma : V \to 2^{[m]}$ associated with its vertices, where $[m] = \{1, 2, \dots, m\}$ denotes the set of colors and $2^{[m]}$ denotes the family of all possible subsets of $[m]$, the goal is to find an s-t path π that minimizes the number of distinct colors it touches. Formally, given a path π in G, if $\sigma(\pi) = \bigcup_{v\in\pi} \sigma(v)$ is set of colors on the path, then the goal is to find an s-t path π minimizing $|\sigma(\pi)|$. In other words, if each color represents an obstacle, then the Min-Color Path problem for a s-t path with the least number of obstacles on it. This is a well-studied problem, first introduced by Jacob et al. [JKK+99], and independently re-discovered and studied in several application domains, including wireless sensor networks [BK09, CK12, CK14, KLA07, KLSS18] and robot motion planning [EL13, Hau14, KB17, XM16].

Unlike the s-t-Shortest Path problem, Min-Color Path is computationally intractable. It is known to be NP-hard [CDKM00] as well as hard from the perspective of parameterized algorithms [EK20]. It is also known that, unless P = NP, Min-Color Path cannot be approximated within a factor of $O(2^{\log^{1-\delta} n})$, for $\delta = 1/\log\log^c n$ and some constant $c < 1/2$, even when the graph is planar [CDKM00]. In particular, Kumar [Kum19] showed that, assuming the stronger Dense vs Random conjecture [CDM17], there is no polynomial time algorithm with approximation factor less than $n^{(\frac{1}{8}-\epsilon)}$, for any fixed $\epsilon > 0$. There are only some limited results in the positive direction. In the case where the colors are on the *edges* instead of vertices and each edge has *only* one color, Hassin et al. [HMS07] presented a $O(\sqrt{n})$-approximation algorithm. For the general Min-Color Path in vertex-colored graphs, the best bound known is an $O(\sqrt{n})$ approximation due to Bandyapadhyaya et al. [BKSV18].

Due to the intractability of Min-Color Path, research has focused on more tractable versions of the problem, and many variants of this problem have been explored in different communities. Perhaps the most studied variant is the Barrier Resilience problem [BK09, CK12, CK14, KLA07, KLSS18] in wireless networks, where the task is to connect two given points s and t in the plane by a path while touching as few obstacles as possible. Because the obstacles model ranges of wireless sensors, they are typically assumed to be disc-shaped. It is a well-known open problem whether Barrier Resilience is polynomial time solvable when all obstacles are unit discs [KLSS18]. However, the problem is known to be NP-hard for many other obstacle shapes, including unit-length line segments, rectangles with aspect ratio close to one, or discs with two different sizes [TK11, EL13, EGKY18, KLSS18]. Chan and Kirkpatrick [CK14, CK12] gave a simple constant factor approximation for the unit disc case and left as an open problem whether there exists a constant factor approximation algorithm for the case when obstacles are general discs.

It is easy to see that Barrier Resilience is a special case of Min-Color Path. Consider the plane arrangement of the discs, and introduce a vertex for each face (connected region of the plane disjoint from disc boundaries) and a vertex for each intersection point of two or more disc boundaries. Next connect by edges all pairs of faces that share a common boundary arc, and each face vertex with all the intersection points on its boundary. Associate a unique color to each disc, and assign to each vertex the colors of its incident discs. We now have a natural correspondence between geometric paths in the plane and paths (walks) in the graph such that the colors on the path are precisely the obstacles touched by the geometric path.

In Computational Geometry and robotics, a variant of the Min-Color Path problem has been studied under the name Minimum Constraint Removal [EL13, EL20, Hau14, KB17, XM16]. The goal again is to find a path from a start point s to a target point t in the presence of obstacles, which are typically modeled as simple closed curves in the plane (although higher dimensional variants are also considered) [Hau14]. To date, the best approximation algorithm for MINIMUM CONSTRAINT REMOVAL in the plane is a factor $O(\sqrt{n})$-approximation algorithm by Bandyapadhyaya et al. [BKSV18], who pose the existence of a better approximation algorithm as an open problem.

The reduction from BARRIER RESILIENCE to MIN-COLOR PATH also works for MINIMUM CONSTRAINT REMOVAL, with one caveat: the number of faces and intersection points in the arrangement of the obstacles should be polynomial in the input size because that determines the size of the graph for MIN-COLOR PATH. For most reasonable obstacle models, such as polygons or low-degree splines, this is indeed the case. Starting with MINIMUM CONSTRAINT REMOVAL the instances of MIN-COLOR PATH produced by this reduction have the following properties: (a) the graph G is planar, and (b) the instances are *color connected* (the set of vertices containing any color is connected). Indeed, it is easy to see that for this class of instances MIN-COLOR PATH can be reduced back to MINIMUM CONSTRAINT REMOVAL in the plane: For every color pick a spanning tree and make an obstacle that traces the outline of the spanning tree in the plane. Our main result is a constant factor approximation algorithm for planar and color-connected instances of MIN-COLOR PATH, resolving the open problems of [BKSV18, CK12, CK14].

Theorem 1. *There exists a polynomial time $O(1)$-approximation algorithm for* MIN-COLOR PATH *on color-connected planar graphs.*

Theorem 1 immediately implies a polynomial time $O(1)$-approximation algorithm for BARRIER RESILIENCE and for MINIMUM CONSTRAINT REMOVAL when the obstacles are sufficiently well behaved (such as general polygons or splines). We complement our results by showing that Theorem 1 is in some ways the best one can hope—any attempt at a significant generalization or improvement runs into a wall of computational intractability. First, while the precise constant of our approximation ratio may be improved somewhat, the problem does not admit a polynomial time approximation scheme unless $\mathsf{P} = \mathsf{NP}$. Indeed, MIN-COLOR PATH on color-connected planar graphs is at least as hard to approximate as VERTEX COVER, even when the input graph has treewidth 3 (see Eiben and Kanj [EK20]). This rules out a better than 2-approximation algorithm assuming the Unique Games Conjecture [KR08] (and better than $\sqrt{2}$ assuming $\mathsf{P} \neq \mathsf{NP}$ [KMS18]). Further, both color-connectivity *and* planarity are necessary to get a constant factor approximation. We show in Section 7 that MIN-COLOR PATH is hard to approximate within a factor of $\max\{m^{1-\epsilon},\ n^{1/4-\epsilon}\}$ assuming a conjecture on the hardness of DENSEST SUBGRAPH [CDM17]. The hardness results hold even for very special classes of instances, namely, diamond paths. Specifically, G is a *diamond path* if it can be obtained by replacing each edge uv of a path by any number of degree 2 vertices adjacent to u and v (See Figure 6). We prove the following hardness result, complementing Theorem 1.

Theorem 2. *Assuming* DENSE VS RANDOM *conjecture* [CDM17]*, one cannot approximate* MIN-COLOR PATH *within ratio $O(m^{1-\epsilon})$ or $O(n^{1/4-\epsilon})$ in polynomial time, for any $\epsilon > 0$, where m is the number of colors and n is the number of vertices in G. The bounds hold even on the following two restricted classes of instances.*

1. *G is a diamond path.*
2. *G has a vertex v so that $G - v$ is a diamond path and (G, σ) is color-connected.*

Although these lower bounds rule out any significant improvement of Theorem 1, we do generalize the result in a different way, by designing a constant-factor approximation for a MIN-COLOR extension of the classical PRIZE COLLECTING STEINER FOREST problem. In this extension, we are given an undirected graph G, vertex coloring σ, multiple source-destination connection pairs $(s_1, t_1), \ldots, (s_k, t_k)$, and a cost w_i for failing to connect the ith pair. The objective is to minimize total number of colors used in all the paths plus the total cost of all the pairs left disconnected. The special case with $w_i = \infty$, forcing all pairs to be connected, is the Min-Color Steiner Forest problem.

Theorem 3. *There exists a polynomial time $O(1)$-approximation algorithm for* Prize Collecting Min-Color Steiner Forest *on color-connected planar graphs.*

While Theorem 3 implies Theorem 1, in the interest of a cleaner presentation we will prove Theorem 1 first, and then point out the (minor) modifications needed for the generalization. The Prize Collecting Min-Color Steiner Forest problem on color-connected planar graphs generalizes classic Steiner Forest and Prize-Collecting Steiner Forest problems on planar graphs. Approximation algorithms for the Steiner Tree problem and its several variants, such as Steiner Forest, Group Steiner Tree, Prize Collecting Steiner Forest, have been widely studied on general graphs and planar graphs. In fact, just on planar graphs there have been significant amount of work done on these problems to design PTASes for them [HKKN12, BDHM16, BHM11, BCE+11, EKM12, BKK07, BKM09, DHK14]. We believe that the ease with which Theorem 1 generalizes to Theorem 3 nicely demonstrates the versatility of our methods.

Our Methods. Given a graph G and a pair of vertices s and t, let $d(s,t)$ denote the length (number of edges) of the shortest path between s and t in G. An *s-t-separator* is a vertex set S such that s and t are in distinct connected components of $G - S$. Shortest paths in (unweighted) graphs admit the following simple min-max theorem: $d(s,t) = |\mathcal{X}| + 1$, where $\mathcal{X}$ is a maximum cardinality set of pairwise vertex-disjoint s-t-separators. The main combinatorial insight behind our algorithm is that a similar (approximate) min-max theorem can be proved for planar and color-connected instances of Min-Color Path. For any set $C \subseteq [m]$ of colors, we define its *host vertex-set* $V(C) \subseteq V$ to be the set of vertices that contain a color in C. That is, $V(C) = \{v \in V \mid \sigma(v) \cap C \neq \emptyset\}$. A set of colors $S \subseteq [m]$ is an *s-t color separator* if $V(S)$ is an s-t-separator (see Figure 1 for an example).

Let $\mathcal{F}$ be the set of all s-t color separators in G. A *fractional packing* of color separators is an assignment $0 \leq y_S \leq 1$ for each color separator S in G, such that for every color c we have

$$\sum_{\substack{S \in \mathcal{F} \\ c \in S}} y_S \leq 1$$

The *value* of a fractional packing is $\sum_{S \in \mathcal{F}} y_S$.

Observe that a family $\mathcal{X}$ of s-t color separators so that no color appears in two different separators in $\mathcal{X}$ corresponds to a fractional packing of value $|\mathcal{X}|$. Since every s-t path P and s-t color separator S satisfy $\sigma(P) \cap S \neq \emptyset$, it is easy to see that for every s-t path P and every fractional packing $\{y_S\}$ of s-t color-separators, the number of colors on P is at least the value of $\{y_S\}$. The engine behind Theorem 1 is that the inequality holds in the other direction as well, up to constant factors.

Theorem 4. *Let (G, σ) be an instance of* Min-Color Path*, OPT be the minimum number of colors on an s-t-path, and v be the maximum value of a packing of s-t-color separators. Then $OPT = \Theta(v)$.*

We remark that Theorem 4 can easily be seen to be false for the instances constructed in our reductions in Theorem 2, barring us from generalizing it to instances that are either non-planar or not color-connected. Theorem 4 is the engine behind the algorithm of Theorem 1, not in the sense that we first prove Theorem 4 and then use it to prove Theorem 1, but in the sense that we conjectured Theorem 4 as a natural extension of the min-max relation for shortest paths, and that the methods for proving Theorem 4 led to a proof of Theorem 1 as well.

Finding the maximum value fractional packing of color separators can be seen as a Linear Program (LP). The proofs both of Theorem 1 and Theorem 4 are by first solving and then rounding its dual LP. This LP has a polynomial number of variables (one for every color), but an exponential number of constraints (one per color separator). In particular, for each color $i \in \{1, 2, \ldots, m\}$, we associate a variable $0 \le x_i \le 1$ that indicates whether or not color i is included in the solution. We then have the following formulation, which we will refer to as the HITTING-LP.

$$\text{minimize} \quad \sum_{i \in [m]} x_i$$

such that,

for all s-t color separators $S \in \mathcal{F}$: $$\sum_{j \in S} x_j \ge 1 \qquad (1)$$

The strong duality theorem of Linear Programming [PS82] implies that v is equal to the optimum of HITTING-LP. An inspection of HITTING-LP shows that it deals with fractional *hitting sets* for the set of all s-t color separators. More formally, a *color-hitting-set* is a set C of colors so that for all s-t color separators $S \in \mathcal{F}$, $C \cap S \neq \emptyset$. Clearly integral solutions to HITTING-LP correspond to color-hitting sets. It turns out that color-hitting sets are strongly tied to minimum color s-t paths: a color set C is a color hitting set if and only if there exists an s-t path π such that $\sigma(\pi) \subseteq C$ (see Lemma 1 for a short proof).

Thus proving Theorem 4 amounts to upper bounding the integrality gap of HITTING-LP by a constant. Making such a proof algorithmic, in the sense of making a polynomial time procedure to find an optimal fractional solution to HITTING-LP, and rounding it to an integral solution with only a constant factor loss in the objective function, is then sufficient to obtain a polynomial time approximation algorithm for MIN-COLOR PATH and prove Theorem 1.

The key insight (and main technical part) of our rounding algorithm is that in the color intersection graph (that has a vertex for every color and an edge between two colors if they are incident to a common face), if we consider the shortest path metric where the distance between two adjacent colors is the average of the LP-values of the corresponding x_c variables, then no ball of radius 0.4 can contain a color separator. With this insight, applying the bounded diameter decomposition of Leighton and Rao [LR99] would immediately yield an $O(\log n)$ approximation algorithm for MIN-COLOR PATH, and a proof of Theorem 4, but with a $O(\log n)$ gap instead of $O(1)$. To get rid of the $O(\log n)$ factor we observe that the color intersection graph is a region graph over a planar graph (see Section 4 for a definition) and apply the improved bounded diameter decomposition of Lee [Lee17] for such graphs.

The final hurdle to obtaining a polynomial time approximation algorithm for MIN-COLOR PATH is that HITTING-LP contains an exponential number of constraints, thus our algorithm can not explicitly construct the LP from the instance and run a solver on it. Instead we will use the ellipsoid method with a separation oracle [PS82]. A separation oracle for HITTING-LP amounts to: given as input a planar graph G, coloring function σ, and assignment x_c for every color c, to determine whether there exists a color separator S such that $\sum_{c \in S} x_c < 1$. This is equivalent to the problem of finding a minimum weight color separator. This problem (or rather its equivalent formulation as finding a minimum cardinality subset of polygonal obstacles whose union separate two points s and t in the plane) is known as 2-POINT SEPARATION, for which a polynomial time algorithm was given quite recently by Cabello and Giannopoulos [CG16]. We were not aware of this algorithm and designed our own polynomial time algorithm for 2-POINT SEPARATION, which turns out to be substantially shorter and (in our opinion) simpler.

We believe that Theorem 4 together with the methods for rounding algorithm for HITTING-LP will find further applications for obstacle removal problems. We do not obtain an explicit bound on the constant in our approximation ratio - in particular it depends on constants hidden in the big-oh notation in the region decomposition theorem of Lee [Lee17]. It is likely that the "correct" constant for Theorem 4, and therefore also for an approximation algorithm based on HITTING-LP to be much smaller. For instances that arise from practical applications, rather than worst case instances, the ratio may be smaller still. Therefore it is interesting both to pin down the "right" constant factor approximation ratio for MIN-COLOR PATH for planar color connected graphs, as well as evaluate the performance of heuristic algorithms for MIN-COLOR PATH based on rounding HITTING-LP on practical instances.

Organization. Our paper is organized as follows. In Section 2 we introduce some basic definitions and outline a roadmap for our algorithm. One of the key insights here is an alternative characterization of MIN-COLOR PATH in terms of *color separators* and finding a minimum size color set that *hits* all color separators. We use this characterization to obtain a linear program for MIN-COLOR PATH on color-connected planar graphs (abbreviated as PLANAR-CONN-MCP). The next Section 3 gives some structural properties of color separators that we use both to solve HITTING-LP and to round it. In Section 4, we will show how to round a solution for this LP to obtain our approximation algorithm. Our rounding scheme makes use of well-known small diameter graph decompositions due to Leighton and Rao [LR99], and Klein, Plotkin and Rao [KPR93]. Section 5 gives proof of Lemma 8 which establishes relationship between color separator and the corresponding separating cycle in the dual. Section 6 gives an alternate polynomial time algorithm for 2-POINT SEPARATION, which is used as for separation oracle. Finally, we discuss some improved hardness of approximations bounds in Section 7. These bounds are based on the so-called DENSE VS RANDOM conjecture [CDM17]. We conclude the paper with some interesting open problems in Section 8.

To follow the main algorithmic result of the paper, a reader may omit Sections 3, and 5 in the first pass, and the hardness result of Section 7 can be read independently.

2 Basic Definitions and Roadmap

In this section, we will introduce the notion of color separators and use them to outline a roadmap for our algorithm. We begin by setting up some basic definitions.

Let (G, σ) denote a *colored graph* that consists of a planar graph $G = (V, E)$ and a coloring function $\sigma : V \to 2^{[m]}$ that satisfies color-connectivity. If $\sigma(v) = \emptyset$ for some $v \in V$, we say that v is a *white vertex*. Without loss of generality, we can assume that both s and t are white vertices. This holds because every s-t path must include all colors from both s and t. For any set $C \subseteq [m]$ of colors, we define its *host vertex-set* $V(C) \subseteq V$ to be the set of vertices that contain a color in C. That is, $V(C) = \{v \in V \mid \sigma(v) \cap C \neq \emptyset\}$. Assuming that s and t are connected in G, we can now define color separators formally as follows.

Definition 1 (Color Separator)**.** *A set of colors $S \subseteq [m]$ is an s-t* color separator *if s and t are disconnected in $G - V(S)$. (See Figure 1 for an example.)*

Let $\mathcal{F}$ to be the set of all s-t color separators of G, then we say that a set of colors C is a *color-hitting set* if $C \cap S \neq \emptyset$ for all s-t color separators $S \in \mathcal{F}$. We have the following lemma.

Lemma 1. *Let C be a set of colors. Then C is a color-hitting set if and only if there exists an s-t path π such that $\sigma(\pi^*) \subseteq C$.*

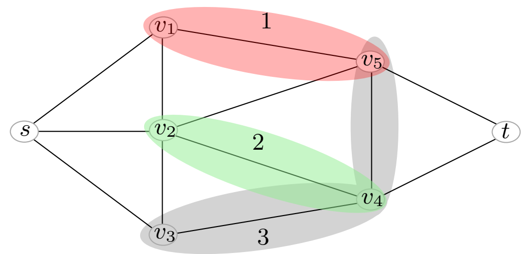


Figure 1: An instance of Planar-Conn-MCP *with three colors. Color sets* $\{1, 2\}$ *and* $\{3\}$ *are both color separators.*

Proof. For the reverse direction, consider a path π such that $\sigma(\pi) \subseteq C$. Every s-t color separator S must intersect $\sigma(\pi)$, because otherwise π is an s-t path in $G - V(S)$, contradicting that S is a color separator. So S must have non-empty intersection with $\sigma(\pi)$ and therefore also with C.

Suppose C is a color-hitting set. Define the coloring function σ' as follows, for every vertex v, $\sigma'(v) = \sigma(v) \setminus C$. We claim that G contains a path π such that $\sigma'(\pi) = \emptyset$. Suppose this is not true, that is, every s-t path in (G, σ') contains at least one color. Then the set of all remaining colors $S' = \bigcup_{v \in V} \sigma'(v)$ is also a color separator. However, we have that $C \cap S' = \emptyset$ which contradicts that C is a color-hitting set. Therefore, $\sigma'(\pi) = \emptyset$, which gives $\sigma(\pi) \subseteq C$. □

An s-t path that uses minimum number of colors is called a *min-color path*. An immediate corollary of Lemma 1 is that the minimum number of colors on an s-t path is the same as the size of the smallest color separator.

Lemma 2. *let* π^* *be a min-color path, and let* C^* *be the smallest color-hitting set, then* $|\sigma(\pi^*)| = |C^*|$.

The above lemma shows that computing a min-color path is equivalent to computing the color-hitting set of smallest cardinality. In the following, we use this equivalence to obtain a linear program for min-color path.

An LP Formulation. For each color $i \in \{1, 2, \ldots, m\}$, we associate a variable $0 \leq x_i \leq 1$ that indicates whether or not color i is included in the solution. We then have the following formulation which we will refer to as Hitting-LP.

$$\text{minimize} \quad \sum_{i \in [m]} x_i$$

such that,

$$\text{for all } s\text{-}t \text{ color separators } S \in \mathcal{F}: \quad \sum_{j \in S} x_j \geq 1 \qquad (2)$$

It is easy to see that an integral solution to this LP is a color-hitting set of smallest cardinality. In order to obtain an approximation algorithm using this LP, we first need to be able to solve it in polynomial time. Although the LP may contain an exponential number of constraints, we can solve it in polynomial time using ellipsoid method provided the existence of a polynomial time *separation oracle*. We use a weighted version of color separators to define a *min-color separator* which will serve as the separation oracle for Hitting-LP.

Definition 2 (Min-Color Separator)**.** *Suppose each color* $i \in [m]$ *has a non-negative weight* w_i *associated with it. Then, a* min-color separator *of* G *is a color separator* $S \subseteq [m]$ *that minimizes the weight* $w(S) = \sum_{i \in S} w_i$.

In Section 6, we will present a polynomial time algorithm for computing a min-color separator. Here we state the resulting lemma.

Lemma 3. *A min-color separator can be computed in polynomial time on planar graphs with color connectivity.*

We note that the problem of computing a min-color separator is essentially the same as the 2-Point Separation problem studied by Cabello and Giannopoulos [CG16], and therefore we could also use the polynomial time algorithm from [CG16] for the separation oracle. However, we did not know about that result and independently obtained an alternative polynomial time algorithm for min-color separator, which is arguably much simpler. Using either of these algorithms as the separation oracle, we obtain the next lemma.

Lemma 4. *The* Hitting-LP *can be solved in polynomial time on planar graphs with color-connectivity.*

The key challenge now is to *round* the fractional solution obtained by solving Hitting-LP to obtain an $O(1)$-approximation. We do this in the next two sections.

3 Properties of Color Separators

In this section, we will discuss some structural properties of color separators that we will use both to solve Hitting-LP and to round it. It is a well known fact [IS79] that an s-t cut in a planar graph corresponds to a *separating cycle*, which is a cycle in the dual graph that separates s from t. Inspired by this, we will establish an equivalence between color separators to a cycle in the dual graph. We begin by fixing an embedding of G and let $G^* = (V^*, E^*)$ be its dual graph. We then have the following lemma.

Lemma 5. *For every color separator S of graph G, there exists a non-empty family of separating cycles $\Gamma(S)$ in the dual graph G^*.*

Proof. Let $E(S) \subseteq E$ be the set of edges adjacent to vertices in $V(S)$. Consider any set $E_\gamma \subseteq E(S)$ such that removing E_γ from G disconnects s from t. That is, edges E_γ are *cut edges*. Given a set of cut edges, it is not hard to obtain a separating cycle γ in the dual graph : draw a simple closed curve enclosing one of s or t and only intersecting the cut edges.

Repeating this for all possible E_γ gives a family $\Gamma(S)$ of separating cycles. Note that $\Gamma(S)$ is non-empty because removing $V(S)$ separates s from t, so the set $E_\gamma = E(S)$ is a trivial cut edge set. □

Therefore, for each color separator S, we can associate a separating cycle $\gamma \in \Gamma(S)$ in the dual graph. Next, we assign colors to the vertices of the dual graph so that the colors on vertices of γ correspond to the colors in S.

Coloring the Dual Graph It will be convenient to first extend the coloring function σ to the edges of G and G^*. That is, for all $e = (u, v) \in E$, we assign $\sigma(e) = \sigma(u) \cup \sigma(v)$. We can then extend it to edges of the dual graph G^* in a natural way. That is, $\sigma(e^*) = \sigma(e)$ where e^* is the dual edge of e. Finally, we extend the coloring to dual vertices as $\sigma(v^*) = \bigcup_{e^* \in\ adj(v^*)} \sigma(e^*)$, where $adj(v^*)$ denotes the set of dual edges adjacent to v^*. We note the following.

Observation 1. *If $v^* \in V^*$ is a dual vertex and f is its corresponding face in G, then $\sigma(v^*) = \bigcup_{v \in \partial f} \sigma(v)$. Here ∂f is the set of vertices on the boundary of face f.*

Lemma 6. *The colored dual graph* (G^*, σ) *is color-connected.*

Proof. Let v_i^*, v_j^* be two vertices in the dual (faces of G). Consider a color $c \in \sigma(v_i^*) \cap \sigma(v_j^*)$. By color-connectivity of G, there must be a path $\pi = v_1 \to v_2 \to \cdots \to v_r$ in G from a vertex v_1 on the boundary of face v_i^* to a vertex v_r on boundary of face v_j^*. We want to find a color-connected path π^* from v_i^* to v_j^* in the dual.

We do an induction on number of vertices r on the path π. If v_2 also lies on face v_i^*, we can drop v_1 from π and we are done by induction. If v_2 lies on some other face v_k^*, we consider the clockwise order of edges $E_1 = adj(v_1)$. From our coloring scheme, it follows that all edges in E_1 contain color c and therefore, all their dual edges E_1^* will also contain color c. Since v_i^*, v_k^* are both faces adjacent to v_1, we can reach v_k^* from v_i^* by traversing edges of E_1^* in clockwise order. Moreover, all the dual vertices and edges in this path contain color c. Now, we can consider the subpath of π from v_2 to v_r and will again be done by induction. $\square$

Now that we have added colors to the dual graph G^*, we want to assign an ordering of colors of the separator S on the cycle γ. We do this by a *labeling function* $\lambda : E^* \to S$ that assigns to every edge $e^* \in \gamma$, a non-white color from $\sigma(e^*) \cap S$. This is always possible due to the following lemma.

Lemma 7. *The set* $\sigma(e^*) \cap S$ *is non-empty for all* $e^* \in \gamma$.

Proof. Let $e = (u, v)$ be the primal edge in G corresponding to e^*. Since e is a cut edge for the color separator S, it must be adjacent to some vertex in $V(S)$. Therefore, the color set $\sigma(e^*) = \sigma(e) = (\sigma(u) \ \cup \ \sigma(v))$ contains at least one color from S. $\square$

We can now easily extend the labeling function λ to vertices of γ as follows. Let the vertices on γ be arranged in clockwise order: $v_1^* \to v_2^* \to \ldots v_r^* \to v_1^*$. If $e_i^* = (v_{i-1}^*, v_i^*)$ is the edge preceding v_i^* in this order, We simply assign $\lambda(v_i^*) = \lambda(e_i^*)$. Since $\sigma(e_i^*) \subseteq \sigma(v_{i-1}^*) \cap \sigma(v_i^*)$, we have the following.

Observation 2. *If* v_{i-1}^* *and* v_i^* *are consecutive vertices on* γ*, then* $\lambda(v_i^*) \in \sigma(v_{i-1}^*)$.

This gives us a cyclic sequence of colors $\lambda(v_1^*) \to \lambda(v_2^*) \ \ldots \ \rightsquigarrow \ \ldots \ \lambda(v_r^*) \to \lambda(v_1^*)$ which we will refer to as *color-cycle* of γ and denote it by $\lambda(\gamma)$. Intuitively, the labeling function λ simply maps a separating cycle $\gamma \in \Gamma(S)$ to a color-cycle $\lambda(\gamma)$ by *selecting* one color belonging to S from every vertex on γ. However, we still need a little more structure on how the colors of S appear on the color-cycle $\lambda(\gamma)$. Towards that end, we establish the notion of a *well-behaved* separating cycle.

Definition 3 (Well-behaved Separating Cycles)**.** *We say that a color-cycle* Z *is* well-behaved *if all occurrences of any given color* c *in* Z *are consecutive. Then* γ *is a* well-behaved separating cycle *if there exists a labeling function* λ *such that the color-cycle* $\lambda(\gamma)$ *is well-behaved.*

For example, the color-cycle $Z_1 = c_1 \to c_2 \to c_2 \to c_1$ is well-behaved, but $Z_2 = c_1 \to c_2 \to c_3 \to c_2 \to c_1$ is not. The following lemma states that a well-behaved separating cycle always exists. For the sake of clarity, we defer its technical details to Section 5.

Lemma 8. *For any color separator* S*, there exists a separating cycle* γ *and its labeling* $\lambda : V^* \to S$ *such that the color-cycle* $\lambda(\gamma)$ *is well-behaved.*

We also note the following important property of separating cycles that follows from Jordan curve theorem.

Lemma 9. *Let* γ *be a simple closed curve in the plane and let* π_{st} *be a simple path between two points* s, t *in the plane. Then* γ separates s *and* t *if and only if the number of times* π_{st} *intersects* γ *is odd.*

Combining Lemma 8 and 9, we have the following.

Lemma 10. *Let* (G, σ) *be a color-connected planar graph and* (G^*, σ) *be its colored dual graph. Moreover, let* π_{st} *be an arbitrary* s–t *path in* G*. Then for any color separator* S *of* (G, σ)*, there exists a* well-behaved *cycle* γ *and its labeling* λ *in* G^* *such that* γ *crosses* π_{st} *an odd number of times.*

4 An $O(1)$-Approximation Algorithm

In the previous sections, we introduced the notion of color separators, established some useful properties, and designed a linear program Hitting-LP and claimed that it can be solved in polynomial time (Lemma 4). Recall that a fractional solution of Hitting-LP corresponds to a solution vector $\hat{x} = \langle x_1, x_2, \ldots, x_m \rangle$ such that $\sum_i x_i \le OPT$ where OPT is the number of colors used by a min-color path. In this section, our goal is to round $\hat{x}$ to compute an integer solution vector $\hat{y}$ such that $\sum_i y_i \le a_0 \cdot OPT$ for some constant a_0. Given a colored graph (G, σ), let $\mathcal{G} = (\mathcal{C}, \mathcal{E})$ be its *color-intersection graph* defined as follows. For every color $i \in [m]$, we add a vertex $c_i \in \mathcal{C}$. Moreover, we add an edge $(c_i, c_j) \in \mathcal{E}$ if the colors i and j *intersect* at some dual vertex $v^* \in G^*$. That is, $\{i, j\} \subseteq \sigma(v^*)$.

Our rounding scheme is based on the well-known *small diameter graph decomposition* (see e.g [LR99] and [KPR93]) method. We will need a node-weighted version of this decomposition where distance values are on nodes and distance between two nodes is defined as the sum of the distance values of the nodes on the shortest path between them, and we will be applying the decomposition to the color intersection graph $\mathcal{G} = (\mathcal{C}, \mathcal{E})$. To state the decomposition result that we need we first need a few definitions.

Suppose G is a graph with a distance function d that assigns a non-negative distance $d(v)$ to every vertex v. We extend the distance function d to the edges of G as follows: the edge uv gets the distance $d(uv) = (d(u) + d(v))/2$. The distance function can now be further extended to pairs of vertices, so that $d(u, v)$ is the shortest path distance between u and v in the edge-weighted graph with edge weights defined by the function d. The diameter of a vertex set $X \subseteq V$ is $\sup_{u,v \in X} d(u, v)$. Notice that when we look at the diameter of a vertex set X we are looking at shortest path distances between vertices of X in the entire graph G, not the shortest path distances in $G[X]$. We will prove the following lemma.

Lemma 11. *If* $\mathcal{G} = (\mathcal{C}, \mathcal{E})$ *is the color-intersection graph of* (G, σ)*, and each vertex* $c_i \in \mathcal{C}$ *has a weight* $w(c_i)$ *and distance* $d(c_i)$*, then for every* $\Delta > 0$ *there exists a set of vertices* $X \subseteq \mathcal{C}$ *such that the diameter of each component of* $\mathcal{G} - X$ *is at most* Δ *and* $\sum_{c_i \in X} w(c_i) = O(1/\Delta) \cdot \sum_{c_i \in \mathcal{C}} d(c_i) w(c_i)$*. Furthermore, such a set* X *can be computed from* $\mathcal{G}$ *and* d *in polynomial time.*

Lemma 11, but with the weaker bound $\sum_{c_i \in X} w(c_i) = O(\log m/\Delta) \cdot \sum_{c_i \in \mathcal{C}} d(c_i) \cdot w(c_i)$ follows quite directly from the work of Leighton and Rao [LR99]. Here m is the number of colors, and this would lead to an $O(\log m)$ approximation for min-color path. To prove Lemma 11 as stated we shall use that $\mathcal{G}$ is a *region intersection graph* over a planar graph, and use a decomposition theorem for such graphs of Lee [Lee17]. We say that a graph $G = (V, E)$ is a *region intersection* graph over the *base* graph $G_0 = (V_0, E_0)$ if for every $u \in V$ there exists a connected set of vertices $R_u \subseteq V_0$ such that$(u, v) \in E$ if and only if $R_u \cap R_v \neq \emptyset$.

Observation 3. $\mathcal{G}$ *is a region intersection graph over a planar graph* G_0*.*

Proof. Let G_0 be the (planar) graph constructed from G by adding a vertex for every face of G and connecting this vertex to every vertex of G incident to the face. We show that $\mathcal{G}$ is a region intersection graph over the base graph G_0. For every vertex $c_i \in \mathcal{C}$ make the region R_i which contains all vertices v of G_0 that are vertices of G and $i \in \sigma(v)$, and all vertices v^* of G_0 that are faces of G and $i \in \sigma(v^*)$. Note that R_i is connected in G_0 for every i, and that $c_i c_j \in \mathcal{E}$ if and only if $R_i \cap R_j \neq \emptyset$. ☐

For any vertex $c \in V$, define a ball centered at c of radius $R \geq 0$ as

$$\mathcal{B}(c, R) = \left\{ v \in V : d(c, v) < R - \frac{w(v)}{2} \right\}$$

A distribution over subsets $S \subseteq V$ is an (α, Δ)-*random separator* if (i) for all $v \in V$ and $R \geq 0$, the probability $\mathbf{P}[\ \mathcal{B}(v, R)$ does not intersect $S\ ] \ \geq \ 1 - \alpha \frac{R}{\Delta}$, and (ii) the diameter of every component of $G - S$ (in the metric d) is at most Δ. We now state the main result of Lee [Lee17] relevant to us.

Proposition 1 (Corollary 4.3 in [Lee17])**.** *If G is a region intersection graph over G_0 such that G_0 excludes K_h as a minor, then G admits an (α, Δ)-random separator with $\alpha = O(h^2)$.*

Putting the above together we obtain the following lemma.

Lemma 12. *Let $w(X) = \sum_{v \in X} w(v)$ denote the weight of vertex set X. If S is sampled from a (α, Δ)-random separator of G then each component of $G - S$ has diameter is at most Δ and the expectation of $w(S)$ is at most $O(\frac{\alpha}{\Delta}) \cdot w(V)$.*

Proof. Since S is an (α, Δ)-random separator, each component of $G - S$ has diameter is at most Δ. For the expectation of $w(S)$ we have that $\mathbf{P}[\ \mathcal{B}(v, R)$ intersects $S\ ] \leq \alpha \frac{R}{\Delta}$. Now suppose we set $R = w(v)$, then the ball $\mathcal{B}(v, R)$ only contains the vertex v. Therefore, $\mathbf{P}[\ \mathcal{B}(v, R)$ intersects $S\ ] = \mathbb{P}[v \in S\ ] \ \leq \ \alpha \frac{w(v)}{\Delta}$, and hence the expected weight of the separator S is $O(\frac{\alpha}{\Delta}) \cdot w(V)$. ☐

We are now ready to complete the proof of Lemma 11.

Proof of Lemma 11. Since $\mathcal{G}$ is a region intersection graph of a planar graph (by Observation 3) and planar graphs are K_5-minor-free, we may apply Proposition 1 on $\mathcal{G}$, and obtain an (α, Δ)-random separator with $\alpha = O(5^2) = O(1)$. By Lemma 12 if a set X is sampled according to this distribution then each component of $G - X$ has diameter is at most Δ, and the expected weight of X is $O(\frac{w(V)}{\Delta})$. This establishes the *existence* of the set X as claimed by Lemma 11.

For a polynomial time algorithm to compute X observe that the random separator of Proposition 1 (see Lee [Lee17]) is based on the well-known iterative chopping scheme of Klein, Plotkin and Rao [KPR93] , which can be turned into a polynomial time algorithm by, in each iteration of the algorithm, greedily choosing the "chop value" that would minimize the total weight of the vertices added to X. We omit the details as this is a standard adaptation of [Lee17] but would require a substantial fraction of the text to be re-produced here. ☐

We are now all set to describe our approximation algorithm (See Algorithm 1). In the algorithm, we will set $\epsilon = 0.1$ First we upper bound the size of the color set returned by the algorithm.

Lemma 13. *If OPT is the minimum number of colors of an s-t path, then the number of colors $|\mathcal{C}^*|$ returned by Algorithm 1 is at most $O(OPT)$.*

Proof. Rounding up variables with $x_j \geq \epsilon$ in Step 2 of the algorithm only increases the cost by a constant factor of $1/\epsilon$. Moreover in Step 4, we apply Lemma 11 with $\Delta = \frac{1}{2} - \epsilon = 0.4$, which gives $|\mathcal{C}^*| = O(1/\Delta) \cdot OPT \ = \ O(OPT)$. ☐

Algorithm 1 Approximate PLANAR-CONNECTED-MCP

Input: A colored graph (G, σ) that is planar graph and color-connected, $\epsilon = 0.1$
Output: Set of colors $\mathcal{C}^*$ that hit all color separators of G

1. Using Lemma 4, solve HITTING-LP in polynomial time. Let $\hat{x} = \langle x_1, x_2, \ldots, x_m \rangle$ be the fractional solution vector.
2. Include all colors j to the solution $\mathcal{C}^*$ such that $x_j \geq \epsilon$.
3. Build the *color intersection* graph $\mathcal{G} = (\mathcal{C}, \mathcal{E})$ over the colors not already included in $\mathcal{C}^*$. Assign weights to the nodes of $\mathcal{G}$ as $d(c_j) = x_j$, where $c_j \in \mathcal{C}$ is the vertex for color j.
4. Apply Lemma 11 on the node-weighted graph $\mathcal{G}$ with diameter $\Delta = \frac{1}{2} - \epsilon$. Let X be the set of cut vertices obtained from the lemma.
5. For each $c_j \in X$, add the corresponding color j to the solution $\mathcal{C}^*$. Return $\mathcal{C}^*$.

In the next two lemmas, we show that Algorithm 1 indeed computes a set of colors $\mathcal{C}^*$ that hits all color separators of G. For the ease of exposition, we will implicitly use a set of colors to also refer to the set of corresponding vertices of graph $\mathcal{G}$.

Lemma 14. *If S is an inclusion minimal color separator, then the colors in S are connected in $\mathcal{G}$.*

Proof. From Lemma 8, we know that for every color separator there exists a well-behaved separating cycle γ in G^* and its labeling $\lambda : V^* \to S$. Since S is inclusion minimal, this labeling is surjective. Consider any two consecutive vertices $v_{i-1}^*, v_i^* \in \gamma$ that get a different label. That is, $\lambda(v_{i-1}^*) = j$, $\lambda(v_i^*) = k$ and $j \neq k$. From Observation 2 it follows that $k \in \sigma(v_{i-1}^*)$ and therefore (c_j, c_k) must be an edge in $\mathcal{G}$. Since $\lambda(\gamma)$ is a well-behaved color-cycle, the colors in S form a cycle in $\mathcal{G}$. □

Lemma 15. *The set of colors $\mathcal{C}^*$ returned by Algorithm 1 hits all s-t color separators.*

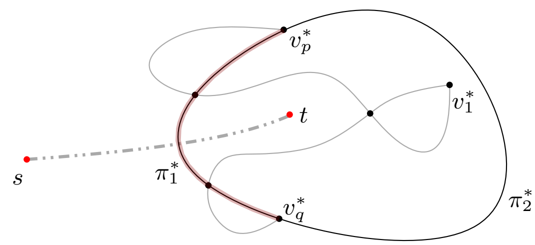
Proof. The proof is by contradiction. Suppose S is an (inclusion minimal) color separator that is not hit by $\mathcal{C}^*$. Then, S must be contained in a single component κ' of $\mathcal{G} - \mathcal{C}^*$. This holds because colors in S are connected $\mathcal{G}$ (Lemma 14) and can be split into different components only if $S \cap \mathcal{C}^* \neq \emptyset$. Let c_1 be a color in S and let κ be the set of colors in a ball in $\mathcal{G}$ of radius $1/2 - \epsilon$ around c_1. Observe that $\kappa' \subseteq \kappa$ by the choice of Δ.

We will now focus only on colors that lie in κ. First, we sort and rename the colors in κ as $\{c_1, c_2, \ldots, c_r\}$ by their distance from the central color c_1 of the ball κ. When renaming, fix a shortest path tree T of κ rooted at c_1 and break ties so that the predecessor c_j of every color c_i (with $i > 1$) in T satisfies $j < i$. Next, we define $C_\ell = \{c_1, c_2, \ldots, c_\ell\}$ to be the color set containing the first ℓ colors.

Observe that C_1 is not a color separator because $c_1 \notin \mathcal{C}^*$ and therefore in HITTING-LP we have that the corresponding variable x_1 is strictly less than 1. If $C_1 = \{c_1\}$ had been a color separator the constraints of HITTING-LP would have forced $x_1 = 1$. Further, observe that C_r contains S and therefore C_r is a color separator. Let $j > 1$ be the smallest index such that the set C_j is a color separator and C_{j-1} is not a color separator.

Since C_{j-1} is not a color separator, we can find an s–t path π_{st} in the *primal* graph G disjoint from $V(C_{j-1})$. That is, no vertex on π_{st} contains a color from C_{j-1}. From the way we assign colors to the dual graph, it follows that no edge on π_{st} contains a color from C_{j-1} and therefore every dual edge that crosses π_{st} does not contain a color from C_{j-1}. This proves the following claim.

Claim 1: Every dual edge that contains a color from C_{j-1} does not cross π_{st}.



(a) Separating cycle γ split into π_1^* and π_2^*

Figure 2: Modifying γ to obtain another separating cycle γ' of weight strictly less than 1. The primal path π_{st} is shown in dash-dotted. Only edges of π_1^* (shown highlighted) can cross π_{st}.

Now consider the separating cycle $\gamma \in G^*$ for the color separator C_j and its well behaved labeling λ that maps every edge of γ to a color in C_j. Let $\pi_1^* = v_p^* \rightsquigarrow v_q^*$ be the maximal subpath of γ such that $\lambda(e^*) = c_j$ for all $e^* \in \pi_1^*$. Let $\pi_2^* = \gamma - \pi_1^*$ be the remainder of γ. Since all occurrences of c_j are consecutive, we have that $\lambda(e^*) \in C_{j-1}$ for all $e^* \in \pi_2^*$. Therefore, from Claim 1, it follows that the path π_2^* does not cross π_{st}. Since γ crosses π_{st} an odd number of times, we obtain the following.

Claim 2: The path π_1^* crosses π_{st} an odd number of times.

Observe that vertices v_p^* and v_q^* both contain at least one color in C_{j-1}. This holds because they are both adjacent to an edge $e^* \in \pi_2^*$ such that $\lambda(e^*) \in C_{j-1}$. Let c_p, c_q be two colors from C_{j-1} that lie on vertices v_p^* and v_q^* respectively. Moreover, let v_1^* be any vertex in the dual graph that contains color c_1.

Let $\hat{C}_p$ be the set of colors on the path from c_p to c_1 in T, and let $\pi_p^* = v_1^* \rightsquigarrow v_p^*$ be a path from v_1^* to v_p^* with a corresponding labeling λ' that assigns to every edge e^* in π_p^* a color from $\hat{C}_p \cap \sigma(e^*)$. Since the radius of κ is at most $1/2 - \epsilon$, the sum of distance values of colors in $\hat{C}_p$ is also at most $1/2 - \epsilon$. Moreover, $\hat{C}_p \subseteq C_{j-1}$ since the predecessor of every vertex in T is eariler in the ordering $c_1, \ldots, c_r$. Thus, every edge in π_p^* contains a color from C_{j-1}, and therefore it follows from Claim 1 that π_p^* does not cross π_{st}. By a symmetric argument, there also exists a similar path $\pi_q^* : v_1^* \rightsquigarrow v_q^*$. (See also Figure 2). Thus we have proved the following claim.

Claim 3: The paths π_p^* and π_q^* do not cross π_{st}.

Next, we combine these paths to obtain a closed walk : $W^* = v_1^* \xrightarrow{\pi_p^*} v_p^* \xrightarrow{\pi_1^*} v_q^* \xrightarrow{\pi_q^*} v_1^*$ in the dual graph. The walk simply picks colors given by the labeling λ' along the paths π_p^* and π_q^*, so the cost of those colors is $2 \times (1/2 - \epsilon)$. For the subpath $v_p^* \rightsquigarrow v_q^*$ of π_1^*, the walk picks the color c_j given by labeling λ. Therefore, the total cost of W^* is $1 - 2\epsilon + d(c_j) \leq 1 - 2\epsilon + \epsilon < 1$. Note that $d(c_j) < \epsilon$ because colors with $d(c_j) > \epsilon$ have already been included into $\mathcal{C}^*$ during Step 2 of the algorithm.

By Claims 2 and 3 the walk W^* crosses π_{st} an odd number of times. Among all closed sub-walks of W^* that cross π_{st} an odd number of times, pick an inclusion minimal one (using the same labeling). This is a simple cycle γ' that crosses π_{st} an odd number of times, so (by Lemma 9) the set of λ'-colors on γ' is a color separator. The cost of the colors on this separating cycle is upperbounded by the cost of W^*. This gives us a color separator S' such that $\sum_{j \in S'} x_j < 1$, which contradicts the constraint for S' in Hitting-LP. □

Lemmata 13, 15, 1, 12 and 4 imply that Algorithm 1 is a constant factor approximation for min color path on planar, color connected instances. More concretely Lemma 4 together with Lemma 12 prove that Algorithm 1 runs in polynomial time, Lemma 13 shows that the size of the set of colors returned by the algorithm is $O(OPT)$, while Lemmata 15 and 1 show that there is a path from s to t using only colors from the set $\mathcal{C}^*$ returned by the algorithm. This proves our main theorem.

Theorem 1. *There exists a polynomial time $O(1)$-approximation algorithm for* Min-Color Path *on color-connected planar graphs.*

The proof of Theorem 4 is nearly identical: the proof of Lemma 13 shows that the set of colors returned by the algorithm is $O(OPT_{LP})$, where OPT_{LP} is the value of the optimum solution to Hitting-LP. LP-duality yields that $OPT_{LP} = \upsilon$, while Lemmata 15 and 1 show that there is a path from s to t using only colors from the set $\mathcal{C}^*$ returned by the algorithm, completing the proof of Theorem 4.

The techniques used to prove Theorem 1 easily extend to the more general Min Color Steiner Forest setting. Here input is a graph (G, σ) and multiple source-destination pairs $(s_1, t_1), \ldots, (s_k, t_k)$. The goal is to find a set of paths $\Pi = \{\pi_1, \ldots, \pi_k\}$ such that π_i is a path connecting s_i to t_i and the total number of colors used by all paths $|\sigma(\Pi)|$ is minimized. Here, $\sigma(\Pi) = \bigcup_{i \in \{1,\ldots,k\}} \sigma(\pi_i)$.

In particular we extend the Hitting-LP by including in the family $\mathcal{F}$ the set of (s_i, t_i) separators for all i. Observe now that we get a separation oracle for this extended linear program by running the separation oracle of Lemma 3 for each s_i-t_i pair individually. If either one of them returns an unsatisfied constraint this is an unsatisfied constraint for the extended LP, because this corresponds to an $s_i - t_i$ color separator S such that $\sum_{i \in S} x_i < 1$.

We now run Algorithm 1, but with this extended LP instead of the original one. Observe that (the proof of) Lemma 13 bounds the number of output colors $|\mathcal{C}^*|$ in terms of the LP-optimum, not just the optimum for minimum color path. Thus the number of colors returned by Algorithm 1 is still $O(OPT)$ where OPT is interpreted as the minimum number of colors in a solution to the Min Color Steiner Forest instance.

Finally, Lemma 15 never uses that $x_1, \ldots, x_m$ is an *optimal* solution to (the original) Hitting-LP, only that it is a feasible solution. Thus, because a feasible solution to the extended linear program is a feasible solution to the original Hitting-LP simultaneously for all input (s_i, t_i) pairs, Lemma 15 yields that the output set $\mathcal{C}^*$ of colors hits all (s_i, t_i)-color separators for every i. Lemma 1 then yields that for every i there is a path connecting s_i and t_i using only colors from $\mathcal{C}^*$. Thus we obtain a constant factor approximation for Min-Color Steiner Forest.

Lemma 16. *There exists a polynomial time $O(1)$-approximation algorithm for* Min-Color Steiner Forest *on color-connected planar graphs.*

In fact, the algorithm of Lemma 16 can easily be generalized to a "Prize-Collecting" version of the problem. Here every input (s_i, t_i) pair comes with a cost w_i, and for every i we have to either connect the pair (s_i, t_i) or pay the cost w_i. The objective is to minimize total number of colors used in all of the paths plus the total cost of all the pairs that are left disconnected. We will call this version of the problem Prize Collecting Min-Color Steiner Forest.

To lift the algorithm of Lemma 16 to this variant we further extend the Hitting-LP by including a variable $0 \leq y_i \leq 1$ for every pair, minimizing $\sum_{i \leq m} x_i + \sum_{i \leq k} y_i \cdot w_i$ and adding y_i on the left hand side of each constraint for $s_i - t_i$ separators.

The rounding algorithm chooses not to connect the pairs (s_i, t_i) with $y_i \geq 1/2$. For the remaining pairs the variable assignment $x'_i = 2 \cdot x_i$ is a feasible assignment to the linear program for Min-Color

Steiner Forest, and we may round this solution with total cost $O(OPT)$ using the algorithm from Lemma 16. This brings us to the most general form of our algorithmic result.

Theorem 3. *There exists a polynomial time $O(1)$-approximation algorithm for* Prize Collecting Min-Color Steiner Forest *on color-connected planar graphs.*

5 Well-Behaved Color Separators

In this section, we present a proof of Lemma 8. That is, we show that for any color separator S, there exists a separating cycle γ and its labeling λ such that its color-cycle $\lambda(\gamma)$ is well-behaved (all occurrences of any given color are consecutive).

Let u^*, v^* be two non-consecutive vertices on γ such that $\lambda(u^*) = \lambda(v^*) = c$. Suppose we split γ at u^*, v^* into two disjoint subpaths $\pi_1 = u^* \rightsquigarrow v^*$ and $\pi_2 = v^* \rightsquigarrow u^*$. That is, $\gamma = \pi_1 \oplus \pi_2$ where $\oplus$ denotes the operation of concatenating two paths at their common endpoints. By color connectivity of G^* (Lemma 6), there must be a simple path π_c from u^* to v^* (called *shortcut-path*) such that all intermediate vertices on this path also contain color c. We now have the following lemma.

Lemma 17. *If π_c is* internally disjoint *from γ, then exactly one of $\gamma_1 = \pi_1 \oplus \pi_c$ and $\gamma_2 = \pi_2 \oplus \pi_c$ is also a separating cycle.*

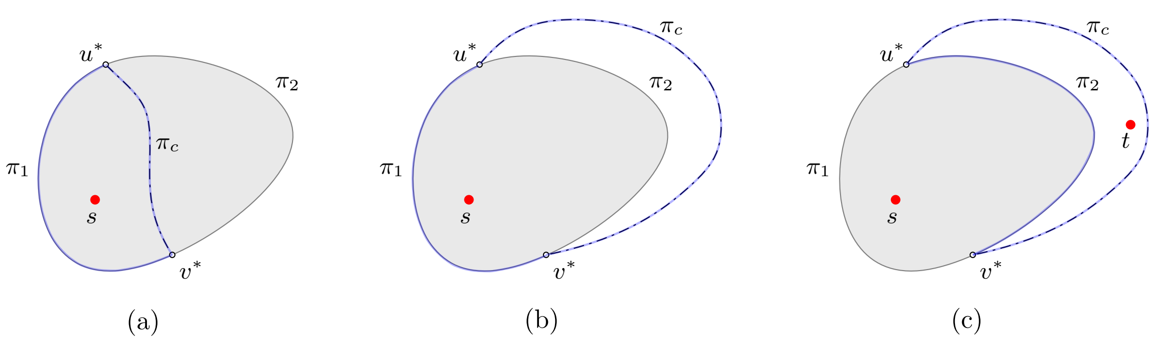


Figure 3: Cases from proof of Lemma 17. Separating cycle γ bounds the region R (shown in gray) and is split into paths π_1 and π_2 by the shortcut path π_c. The new separating cycles are shown in bold.

Proof. Since π_c is internally disjoint from γ, both γ_1 and γ_2 are simple cycles. Let R be the region enclosed by γ, we have the following two cases. We assume that source s lies inside R and destination t lies outside R.

1. *Path π_c lies inside R:* In this case, π_c splits the region R into two disjoint sub-regions defined by γ_1 and γ_2. We set γ' to whichever of γ_1 or γ_2 contains the source s. (See also Figure 3a.)

2. *Path π_c lies outside R:* In this case, one of the two cycles γ_1 and γ_2 completely encloses the region R and the other is disjoint from R. Without loss of generality assume that γ_1 encloses R and γ_2 is disjoint from R. If t lies outside γ_1, we set γ' to γ_1 as our separating cycle as shown in Figure 3b. If t lies inside γ_1, we set γ' to γ_2 as shown in Figure 3c.

In both these cases, we were able to find another separating cycle γ', which is basically π_c concatenated with either π_1 or π_2. ☐

For any color-cycle $Z = \lambda(\gamma)$, we can define the measure $\mu(\gamma, \lambda)$ to be the number of times Z switches colors. That is, $\mu(\gamma, \lambda)$ is the number of indices i where $Z_i \neq Z_{i+1}$. Moreover, let $u^*, v^* \in \gamma$

be two non-consecutive vertices (called a *violating pair*) such that $\lambda(u^*) = \lambda(v^*) = c$ and at least one intermediate vertex on both subpaths π_1, π_2 of γ gets a different label. We then have the following lemma.

Lemma 18. *Let γ be a separating cycle and λ be its labeling. If u^*, v^* is a violating pair of vertices of γ and π_c is a shortcut-path connecting them, then there exists another cycle γ' and its labeling λ' such that $\mu(\lambda', \gamma')$ is strictly less than $\mu(\lambda, \gamma)$.*

Proof. The proof is by induction on the number of vertices in $\gamma \cap \pi_c$. The base case is when $|\gamma \cap \pi_c| = 2$. That is, when π_c and γ are internally disjoint. Then using Lemma 17 either $\gamma' = \pi_1 \oplus \pi_c$ or $\gamma' = \pi_2 \oplus \pi_c$. In both cases, we can label $\lambda'(w^*) = c$ for all vertices $w^* \in \pi_c$. It is easy to verify that $\mu(\lambda', \gamma') < \mu(\lambda, \gamma)$.

For the inductive step, let $u_1^* \in \gamma \cap \pi_c$ be the vertex closest to u^* on the path π_c. We have the following cases.

1. *u^* and u_1^* are consecutive on γ.* In this case, we can simply modify $\lambda(u_1^*) = c$ which does not increases $\mu(\gamma, \lambda)$. Now we apply induction with $u^* = u_1^*$ and the shortcut π_c to be the subpath of π_c from u_1^* to v^*. Clearly, the number of vertices in $\gamma \cap \pi_c$ is one less and by induction there must exist another cycle γ' and its labeling λ' such that $\mu(\gamma', \lambda') < \mu(\gamma, \lambda)$.

2. Otherwise, u^* and u_1^* are non-consecutive on γ and the subpath $\pi_c' : u^* \rightsquigarrow u_1^*$ of π_c is internally disjoint from γ. The vertices u^* and u_1^* split γ into two paths: let π_1' be the path containing vertex v^* and π_2' be the other path. (See also Figure 4a.) Next, we apply Lemma 17 again with the shortcut-path π_c' connecting vertices u^*, u_1^*. This gives that either $\gamma_1 = \pi_1' \oplus \pi_c'$ or $\gamma_2 = \pi_2' \oplus \pi_c'$ as a separating cycle. In both cases, we modify the label $\lambda(w^*) = c$ for all vertices $w^* \in \pi_c'$. That is, the π_c' part of the separating cycle does not switch colors. We now analyze the number of color switches in both cases as follows.

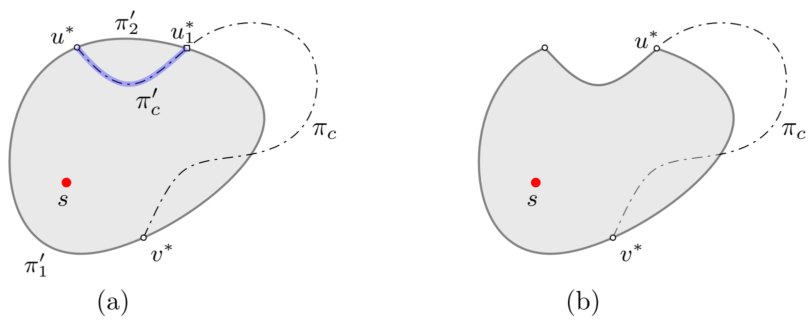


Figure 4: (a) The cycle γ with $|\gamma \cap \pi_c| = 4$. Here, $\gamma_1 = \pi_1' \oplus \pi_c'$ is also a separating cycle. (b) The cycle γ_1 with $|\gamma_1 \cap \pi_c| = 3$. The shortcut-path π_c is shown dash-dotted.

   (a) *$\gamma_1 = \pi_1' \oplus \pi_c'$ is a separating cycle.* Observe that γ_1 is obtained by replacing the part π_2' of γ with path π_c'. It is easy to verify that that $\mu(\gamma_1, \lambda) \leq \mu(\gamma, \lambda)$. Next we apply induction with $u^* = u_1^*$ and the shortcut-path π_c to be the subpath of π_c from u_1^* to v^*. The number of vertices in $\gamma_1 \cap \pi_c$ is one less and by induction there must exist another cycle γ' and its labeling λ' such that $\mu(\gamma', \lambda') < \mu(\gamma_1, \lambda) \leq \mu(\gamma, \lambda)$. (See also Figure 4b.)

   (b) *$\gamma_2 = \pi_2' \oplus \pi_c'$ is a separating cycle.* In this case, observe that γ_2 is obtained by replacing the part π_1' of γ (that contains v^*) with the path π_c'. Indeed, replacing π_1' gets rid of at least one color switch due to vertex v^* and adds no new color switches. Therefore, $\mu(\gamma_1, \lambda) < \mu(\gamma, \lambda)$.

□

Lemma 19. *Every color separator S has a separating cycle that is well-behaved.*

Proof. Let $\gamma \in \Gamma(S)$ be any separating cycle and λ be an arbitrary labeling function. (From Lemma 7, such a labeling always exists). Now, if the color-cycle $\lambda(\gamma)$ is well-behaved, we are done. If not, there must be a violating pair of vertices $u^*, v^* \in \gamma$ such that $\lambda(u^*) = \lambda(v^*) = c$. Therefore we can apply Lemma 18, and obtain a separating cycle γ' with at least one less color switch. Since the number of color switches $\mu(\gamma, \lambda) > 0$, exhaustively applying Lemma 18 gives a separating cycle with no violating pair of vertices. □

6 Computing a Min-Color Separator

In this section, we will discuss a polynomial time algorithm to compute the minimum weight color separator. Specifically, we are given the colored graph (G, σ) such that each color $j \in [m]$ has a weight w_j associated with it and the goal is to compute a color separator S that minimizes $w(S) = \sum_{j \in S} w_j$.

The key to an algorithm for min-color separator is Lemma 10, which states that for every color separator S, there exists a well-behaved cycle in the colored dual graph (G^*, σ) that crosses any arbitrary s-t path π_{st} an odd number of times. Recall that the notion of well-behaved means that all occurrences of a given color on the cycle are consecutive. This lets us formulate the problem of finding a min-color separator as a shortest path problem in an auxiliary *layered graph* H.

6.1 Constructing the Auxiliary Graph H

The input to our construction is the colored dual graph (G^*, σ) and an arbitrary s-t path π_{st}. Roughly speaking, the auxiliary graph H consists of two layers: L_a and L_b, with an identical set of vertices and two types of edges: *intra-layer* edges that go within the layer and *inter-layer* edges that go between layers. The graph H will be *edge weighted*. We first add vertices and edges to H and later assign weights to its edges.

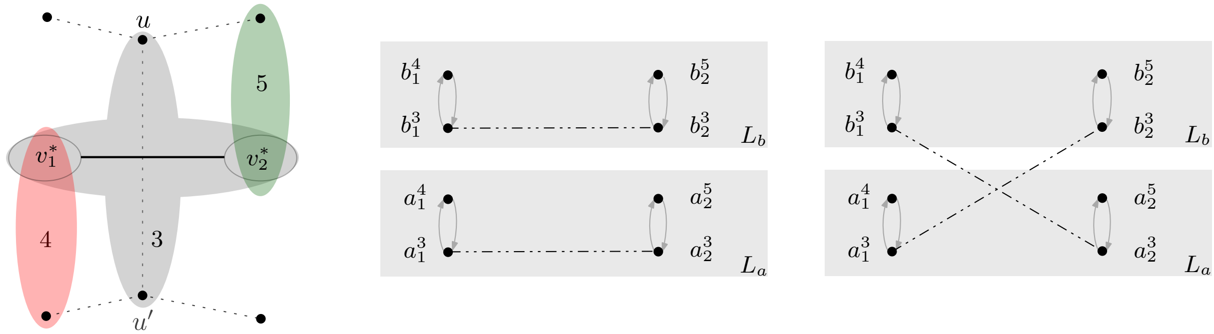
Adding vertices to H For every dual vertex $v_i^* \in V^*$, we create $r = |\sigma(v_i^*)|$ copies in L_a and L_b, one for each color in $\sigma(v^*)$. More precisely, for every pair (i, j) such that $v_i^* \in V^*$ and $j \in \sigma(v_i^*)$, we add the vertex a_i^j in layer L_a and b_i^j in layer L_b.

Since G^* is a planar graph, we can think of following visualization of H in three dimensions. Consider L_a be the bottom layer, L_b to be the top layer, and stack all copies a_i^j, b_i^j of v_i^* one above another, such that all a_i^j copies come first followed by b_i^j. (See also Figure 5)

Adding edges to H We add two groups of edges to H. The first group will be called *clique edges* and are added as follows. Let A_i be the set of all copies of vertex v_i^* in layer L_a. Add edges to H such that the vertex set A_i is a clique. Similarly, let B_i be the set of all copies of vertex v_i^* in layer L_b, add edges to H so that B_i is a clique. Repeat for all v_i^* Note that clique edges are intra-layer edges.

The second group of edges will be called *free edges* and are added as follows. For each edge $e^* = (v_x^*, v_y^*)$ of the dual graph G^*, we add a set of edges E_{xy} as follows depending on whether or not e^* *crosses* path π_{st}. We have the following two cases.

1. *Edge e^* does not cross π_{st}*: For every $j \in \sigma(e^*)$, add edges (a_x^j, a_y^j) and (b_x^j, b_y^j). Note that all edges added in this case are also *intra-layer* edges. (See Figure 5(b))



(a) $e^* = (v_1^*, v_2^*)$ (b) e^* does not crosses π_{st} (c) e^* crosses π_{st}

Figure 5: An example of layered graph construction with $w_j = 1$ for all colors $j \in [m]$. An edge $e^* = (v_1^*, v_2^*)$ and two cases for adding corresponding edges to H are shown. In Figure (b, c), free edges *of H are dash-dotted and have weight zero. The* clique edges *of H are shown in gray have weight 1.*

2. *Edge e^* crosses π_{st}*: For every $j \in \sigma(e^*)$, add edges (a_x^j, b_y^j) and (a_y^j, b_x^j). Note that all edges added in this case are *inter-layer* edges. (See Figure 5(c))

Assigning weights All *free edges* are assigned a weight of zero. Next, we assign weight to *clique edges*. Recall that *clique edge* are of the form (p_i^k, p_i^ℓ) where $p \in \{a, b\}$ and $k, \ell \in \sigma(v_i^*)$. First, we make these edges directed by adding two directed edges $(p_i^k \to p_i^\ell)$ and $(p_i^k \leftarrow p_i^\ell)$. We assign the weights as $w(p_i^k \to p_i^\ell) = w_\ell$ and $w(p_i^k \leftarrow p_i^\ell) = w_k$. (See also Figure 5.) Intuitively, we can think of this assignment of weights as follows: we only pay for a color when entering its vertex – all consecutive usage is free.

6.2 Min-Color Separator as Shortest Path on *H*

In order to cast min-color separator as a shortest path problem on H, we need to make one final change: for each vertex $v_i^* \in G^*$ we add a pair of *source-sink* nodes and connect them to existing nodes of H. Specifically, for each $v_i^* \in G^*$, we add two special vertices u_i^a (source) and u_i^b (sink) to H. Next, we add the source edges $(u_i^a \to a_i^k)$ for all $k \in \sigma(v_i^*)$ and assign its weight to be w_k. Similarly, we add the sink edges $(b_i^k \to u_i^b)$ and assign their weight to be zero.

Lemma 20. *Let S be the min-color separator and π be the shortest of all source-sink paths in H. Then $w(S) = w(\pi)$.*

Proof. Given a color separator S, we incrementally build a path $\pi : u_x^a \rightsquigarrow u_x^b$ in H as follows. Using Lemma 8, we first obtain a separating cycle γ and its well-behaved labeling λ. Fix any vertex v_x^* on γ and let $k = \lambda(v_x^*)$. Then we add the *source* edge $u_x^a \to a_x^k$ to π. Now, let $e^* = (v_x^*, v_y^*)$ be the edge adjacent to v_x^* on cycle γ in clockwise order. We have two cases.

1. $\lambda(v_x^*) = \lambda(v_y^*) = k$. If e^* crosses the reference path π_{st}, we add the inter-layer edge $a_x^k \to b_y^k$ to π. Otherwise, we add intra-layer edge $a_x^k \to a_y^k$ to π.

2. $\lambda(v_x^*) = k$ and $\lambda(v_y^*) = \ell$. We know from Observation 2 that $\ell \in \sigma(v_x^*)$. Therefore, we can add the clique edge $a_x^k \to a_x^\ell$ followed by the free edge $a_x^\ell \to a_y^\ell$ (or $a_x^\ell \to b_y^\ell$) to our path π.

We repeat this for all edges of γ in clockwise order and finally add the *sink* edge $b_x^k \to u_x^b$ to the path π. Since γ crosses π_{st} and odd number of times, we can verify that π will be connected. Moreover, since γ is well-behaved, we pay for each color exactly once and therefore $w(\pi) = w(\gamma) \leq w(S)$.

For the other direction, given a source-sink path $\pi : u_i^a \rightsquigarrow u_i^b$ in H, we obtain a separating cycle γ and its labeling λ as follows. First, we simply replace each vertex of the form a_x^y (or b_x^y) by its corresponding vertex $v_x^* \in G^*$. This gives us a closed walk W^*. We claim that all occurrences of any given vertex v_x^* on W^* are consecutive. This holds because π is a shortest path, so it does not contain a subpath of the form $a_x^k \rightsquigarrow a_y^\ell$ (or b_y^ℓ) $\rightsquigarrow a_x^\ell$. For the same reason, π will also not contain a subpath of the form $a_x^k \rightsquigarrow b_x^\ell$ for some $x \neq i$ (else the source-sink pair $u_x^a \rightsquigarrow u_x^b$ would be the shortest). Therefore, we can compress consecutive occurrences of the same vertex v_x^* on W^* into one to obtain the cycle γ. We choose the label $\lambda(v_x^*)$ to be any color k such that a_x^k (or b_x^k) lies on π. This gives $w(\gamma) \le w(\pi)$. Finally, since u_i^a and u_i^b lie in different layers, π must have taken an odd number of inter-layer edges, and therefore it must cross π_{st} an odd number of times. Since γ is a separating cycle, the set of colors S in $\lambda(\gamma)$ form a color separator with $w(S) \le w(\pi)$. ☐

From Lemma 20, we obtain the following theorem.

Theorem 5. *A min-color separator on color-connected planar graphs can be computed in polynomial time.*

It turns out that color-connectivity is crucial for a polynomial time algorithm for computing a min-color separator. Without color connectivity, the problem can easily be shown to be NP-hard even on planar graphs of treewidth two, by a simple reduction from the Hitting Set problem.

7 Hardness of Approximation

In this section, we give improved lower bounds for approximating min-color paths when G is not a color-connected planar graph. In particular, we show that under plausible complexity theoretic assumptions, without planarity and color-connectivity, the Min-Color Path problem admits no polynomial time $O(m^{1-\epsilon})$-approximation algorithm, and no polynomial time $O(n^{1/4-\epsilon})$-approximation algorithm, where m is the number of colors and n is the number of vertices in the graph. In fact, these bounds hold even when G has constant treewidth.

We begin by discussing Dense vs Random, the complexity-theoretic assumption that our lower bounds are based on. A hypergraph $\mathcal{G} = (X, H)$ over a set of vertices X is *r-uniform* if every hyperedge of H has cardinality r. For positive integers n and r and real $0 \le p \le 1$, the $\mathcal{G}(n, p, r)$ *model* is the probability distribution on r-regular n-vertex hypergraphs where every subset of the vertex set of size r is included in the set of hyperedges independently with probability p.

For a hypergraph $\mathcal{G} = (X, H)$ and vertex set $X' \subseteq X$ the sub-hypergraph of $\mathcal{G}$ *induced by* X' is denoted by $\mathcal{G}[X']$ and defined as $(X', \{E \in H : E \subseteq X'\})$. Our hardness results are based on the assumed hardness of the densest-k-subhypergraph problem. Here input is a hypergraph $\mathcal{G}$ and integer k, and the task is to find the subset X' of X that maximizes the number of hyperedges in $\mathcal{G}[X']$. This problem is conjectured to be very hard to approximate. Indeed, it is conjectured that there is no polynomial time algorithm that can distinguish between hypergraphs with a dense subhypergraph on k vertices, and hypergraphs drawn from $\mathcal{G}(n, p, r)$ where the parameters are set in such a way that with high probability each set on k vertices induces a subhypergraph with very few edges. We now formally define this conjecture.

Conjecture 1 (Dense vs Random [CDM17]). *For all constant r and $0 < \alpha, \beta < r - 1$, sufficiently small $\epsilon > 0$, and function $k : \mathbb{N} \to \mathbb{N}$ so that $k(n)$ grows polynomially with n, $(k(n))^{1+\beta} \le n^{(1+\alpha)/2}$, there does not exist an algorithm* ALG *that takes as input an r-regular n-vertex hypergraph $\mathcal{G}$, runs in polynomial time, and outputs either* dense *or* sparse*, such that:*

- *For every subhypergraph $\mathcal{G}$ that contains an induced sub-hypergraph on $k = k(n)$ vertices and $k^{1+\beta}$ hyperedges,* $\mathsf{ALG}(\mathcal{G})$ *outputs* dense *with high probability.*
- *If $\mathcal{G}$ is drawn from $\mathcal{G}(n, p, r)$ with $p = n^{\alpha-(r-1)}$ then* $\mathsf{ALG}(\mathcal{G})$ *outputs* sparse *with high probability.*

In Conjecture 1 and the remainder of this section, *with high probability* (w.h.p) means with probability at least $1 - O(n^{-c})$ for some constant $c > 0$. In the dense case the probability is taken only over the random bits drawn by the algorithm ALG if ALG is a randomized algorithm. In the sparse case the probability taken over both the draw of $\mathcal{G}$ from $\mathcal{G}(n, p, r)$ and the random bits of the algorithm.

We remark that Conjecture 1 is not stated in exactly this way by [CDM17], indeed their statement of the conjecture leaves some details to interpretation. The statement of the conjecture here is (in our opinion) the "weakest reasonable formalization" of the conjecture of [CDM17], in the sense that every reasonable way to disambiguate their conjecture leads to a statement that implies ours.

In order to obtain hardness guarantees for our problem using Conjecture 1, we will describe a reduction Red that given a hypergraph $\mathcal{G}$ produces an instance (G, σ) of Min-Color Path. We shall then argue that, the images of dense instances under this transformation will have (with high probability) optimum *at most* x^*_d, while the images of random instances under this transformation will have optimum at least x^*_r, where x^*_r is much bigger than x^*d. We shall call $d = x^*_r/x^*_d$the *distinguishing ratio* of the reduction. An approximation algorithm for Min-Color Path with ratio smaller than d can now (with high probability) distinguish between the images of dense and random instances, thereby refuting Conjecture 1. This immediately yields the following lemma.

Lemma 21. *Suppose there exists a construction with distinguishing ratio d, then, assuming Conjecture 1, there is no polynomial time approximation algorithm for* Min-Color Path *with approximation ratio less than d.*

In light of Lemma 21 we will now provide two constructions, one with distinguishing ratio $m^{1-\epsilon}$ where m is the number of colors in the instance (G, σ), and the other with distinguishing ratio $|V(G)|^{1-\epsilon}$, implying the previously claimed hardness of approximation results. The constructions for the two cases are identical, however the choices of parameters α, β, k and r are different for the two results. For this reason we will give a more general construction (for a range of the parameter values) and prove some of its properties, and then obtain our results by instantiating the parameters.

We now prove the following bound on the number of vertices in a subhypergraph of $\mathcal{G}(n, p, r)$ which will be used later.

Lemma 22. *Let $\mathcal{G}$ be drawn from $\mathcal{G}(n, p, r)$. Then, with high probability, every subhypergraph of $\mathcal{G}$ with $q = n^{\Omega(1)}$ hyperedges contains $\tilde{\Omega}(\min\{q, (q/p)^{1/r}\})$ vertices. Here $\tilde{\Omega}$ ignores logarithmic factors.*

Proof. Define $z = \min\left\{\frac{q \ln\ln n}{3\ln n},\ \left(\frac{q}{ep\ln n}\right)^{1/r}\right\}$, where ln denotes natural logarithm. We will show that for every fixed set of z vertices, the probability that the sub-hypergraph induced by these vertices contains at least q edges is small. Let H be any subhypergraph of $\mathcal{G}$ with z vertices. The probability that H has q edges is the same as probability of q successes in $N = \binom{z}{r}$ Bernoulli trials where each trial has success probability p. Therefore, we have:

$$\begin{aligned}
\Pr\left[H \text{ contains } q \text{ edges}\right] &= \binom{N}{q} \cdot p^q \cdot (1-p)^{(N-q)} \\
&\leq \left(\frac{eN}{q}\right)^q \cdot p^q \qquad\qquad \text{since } (1-p)^{(N-q)} \leq 1
\end{aligned}$$

$$
\begin{aligned}
&\le \left(\frac{ez^r}{q}\right)^q \cdot p^q && \text{since } N < z^r \\
&\le \left(\frac{ep}{q} \cdot z^r\right)^q \\
&\le \left(\frac{1}{\ln n}\right)^q && \text{since } z \le \left(\frac{q}{ep\ln n}\right)^{1/r} \\
&\le \left(e^{-\ln\ln n}\right)^q \\
&\le \left(e^{-\ln\ln n}\right)^{3z\ln n/\ln\ln n} && \text{since } z \le \frac{q\ln\ln n}{3\ln n} \\
&\le e^{\ln n^{-3z}} \quad \le \quad \frac{1}{n^{3z}}
\end{aligned}
$$

Applying the union bound over all possible subsets on z vertices we get that the probability that there exists a sub-hypergraph of $\mathcal{G}$ on z vertices with at least q hyperedges is at most $n^z \cdot n^{-3z} = 1/n^{2z}$. $\square$

Construction. For every fixed r, α and β and $k : \mathbb{N} \to \mathbb{N}$ satisfying the conditions of Conjecture 1 we describe a reduction that, given a r-uniform hypergraph $\mathcal{G} = (X, H)$ on n vertices constructs an instance of Min-Color Path, namely a graph $G = (V, E)$ and a coloring function $\sigma : V \to 2^X$. Note that the vertex-set X of the hypergraph forms the color set for (G, σ). We set the following parameters: $q = k^{1+\beta}$, and $\ell = \frac{q}{(r+1)\ln n}$. Observe that because k grows polynomially with n, so does q. Thus q satisfies the conditions of Lemma 22. The construction proceeds as follows.

- Add $\ell + 1$ vertices $v_1, v_2, \ldots, v_{\ell+1}$ to G and arrange them sequentially in the plane from left to right. (See also Figure 6.)
- Uniformly partition hyperedges H into ℓ groups as $H_1, H_2, \ldots, H_\ell$. That is, every hyperedge is assigned a group with a probability $1/\ell$ independent of other hyperedges.
- For each hyperedge $e \in H_i$, add a vertex v_e and connect it to vertices v_i and v_{i+1}. Assign $\sigma(v_e) = adj(j)$, where $adj(e)$ to denotes vertices adjacent to hyperedge $e \in H$.
- Let $s = v_1$ be the source $t = v_{\ell+1}$ be the destination.

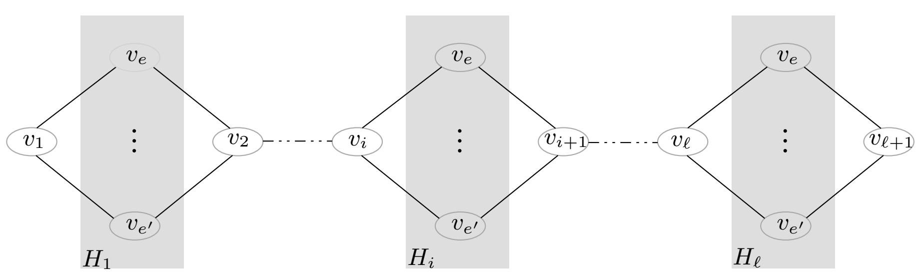


Figure 6: An example of the construction. The groups H_1, H_i and H_ℓ are shown shaded in gray.

The following lemma follows immediately from a standard "balls and bins" kind of argument, using that we set $\ell = \frac{q}{(r+1)\ln n}$.

Lemma 23. *With high probability, for every subset $H^* \subseteq H$ of q hyperedges of $\mathcal{G}$, every group $H_i \in \{H_1, H_2, \ldots, H_\ell\}$ contains at least one edge from H^*. Here n is the number of vertices of $\mathcal{G}$.*

Proof. Fix a subset H^* on q hyperedges. We say that group H_i is H^*-*empty* if $H_i \cap H^* = \emptyset$ (in other words H_i contains no edge from H^*). The probability H_i is H^*-empty is upper by $(1-\frac{1}{\ell})^q \leq \frac{1}{n^{r+1}}$. A union bound over all $\ell \leq n^r$ groups H_i proves the statement of the lemma. □

Lemma 24. *Let (G, σ) be the result of applying the transformation above to a hypergraph $\mathcal{G}$.*

1. *If $\mathcal{G}$ contains a sub-hypergraph on k vertices and q hyperedges, then with high probability G contains an $s-t$ path that uses at most k colors.*

2. *If $\mathcal{G}$ is drawn from $\mathcal{G}(n,p,r)$ with $p = n^{\alpha-(r-1)}$ then with high probability every $s-t$ path in G uses at least $\tilde{\Omega}(\min\{q, (q/p)^{1/r}\})$ colors.*

Proof. For the first point, $\mathcal{G}$ contains a subhypergraph with k vertices and at least $q = k^{\beta+1}$ hyperedges. Let H^* be the set of these hyperedges. Applying Lemma 23, we get that each of the ℓ groups H_i contain at least one edge from H^*. Therefore, the colored graph G contains a path that only uses colors corresponding to vertices in hyperedges in the edge set H^*. Since the number of such vertices is at most k, there exists a path in G with at most k colors.

For the second point, the set of colors of any $s-t$ path P in G corresponds to a subhypergraph of $\mathcal{G}$ with ℓ edges. Applying Lemma 22 on this subhypergraph and observing that $\ell = \tilde{\Omega}(q)$, yields that P uses at least $\tilde{\Omega}(\min\{q, (q/p)^{1/r}\})$ colors. □

From Lemma 24, we obtain the following lower bound on the distinguishing ratio:

$$d \quad \geq \quad \frac{\min\{k^{\beta+1}, \quad (k^{\beta+1}/n^{1+\alpha-r})^{1/r}\}}{k} = \min\left\{k^{\beta}, \quad \left(\frac{k^{\beta+1-r}}{n^{1+\alpha-r}}\right)^{1/r}\right\} \tag{3}$$

We are now ready to prove the first of our hardness of approximation bounds.

Lemma 25. *Assuming* Dense vs Random, Min-Color Path *on a graph $G = (V, E)$ with a coloring function $\sigma : V \to 2^{[m]}$ cannot be approximated within a factor of $m^{1-\epsilon}$, for any $\epsilon > 0$.*

Proof. We choose the parameters $\alpha = \sqrt{r} - 1$, $\beta = \alpha - \epsilon'$ and $k = n^{\frac{1}{\sqrt{r}+1}}$ for some sufficiently small $\epsilon' > 0$. It is easy to verify that these parameter values satisfy the requirements for Conjecture 1.That is, we have $0 \leq \alpha, \beta < r - 1$ and $k^{\beta+1} = n^{\frac{\sqrt{r}-\epsilon}{\sqrt{r}+1}} \leq n^{1+\alpha}$.

Substituting the values in Equation 3, we obtain the following bound for the distinguishing ratio.

$$k^{\beta} \quad = \quad n^{\frac{\sqrt{r}-1-\epsilon}{\sqrt{r}+1}} \quad = \quad n^{1-\frac{2+\epsilon}{\sqrt{r}+1}} \quad = \quad n^{1-\epsilon'} \qquad \text{where } \epsilon' = \frac{2+\epsilon}{\sqrt{r}+1}$$

$$\text{Similarly, we have } \left(\frac{k^{\beta+1-r}}{n^{1+\alpha-r}}\right)^{1/r} \quad = \quad n^{\frac{\sqrt{r}-1-\epsilon/r}{\sqrt{r}+1}} \quad = \quad n^{1-\epsilon''} \qquad \text{where } \epsilon'' = \frac{2+\epsilon/r}{\sqrt{r}+1}$$

Setting ϵ' sufficiently small and r sufficiently large (but independent of n) yields that the distinguishing ratio d is at least $n^{1-\epsilon}$. Here n is the number of vertices in $\mathcal{G}$, which is equals to the number m of colors in (G, σ). The statement now follows from the lower bound on the distinguishing ratio together with Lemma 21. □

The above lemma shows that, it is quite unlikely to find a good approximation in terms of the number of colors. We will now obtain a bound in terms of number of vertices of G. Observe that the number of vertices in G is $h + \ell = \Theta(h)$ where h is the number of hyperedges of $\mathcal{G}$. We can rewrite Equation 3 by expressing the probability $p = n^{1+\alpha-r}$ in terms of the expected number

$h = \theta(n^{1+\alpha})$ of edges in a hypergraph drawn from $\mathcal{G}(n, p, r)$. Note that the actual number of edges of a hypergraph $\mathcal{G}$ drawn from $\mathcal{G}(n, p, r)$ is $\theta(n^{1+\alpha})$ with high probability. Thus, we can rewrite the lower bound on the distinguishing ratio from Equation 3 in terms of h as follows, using that $p = h^{(1+\alpha-r)/(1+\alpha)}$.

$$\mathrm{d} \quad \geq \quad \min \left\{ k^{\beta}, \quad \left(\frac{k^{\beta+1-r}}{h^{(1+\alpha-r)/(1+\alpha)}} \right)^{1/r} \right\} \tag{4}$$

Moreover, since the number of vertices in the colored graph G is $\Theta(h)$, we obtain the following theorem.

Lemma 26. *Assuming* Dense vs Random, Min-Color Path *on a graph $G = (V, E)$ with a coloring function $\sigma : V \to 2^{[m]}$ cannot be $O(|V|^{1/4-\epsilon})$-approximated, for any $\epsilon > 0$.*

Proof. We set the parameters $\alpha = \frac{r-1}{r+1}$, $\beta = \alpha - \epsilon$ and $k = h^{\frac{r+1}{4r}}$, and choose $\epsilon' > 0$ sufficiently small. It is easy to verify that these parameter values satisfy Conjecture 1. More precisely, we have $0 < \alpha, \beta < (r-1)$ and $k^{\beta+1} \;<\; k^{\alpha+1} \;=\; h^{\frac{r+1}{4r}\cdot\frac{2r}{r+1}} \;=\; \sqrt{h} \;=\; n^{\frac{1+\alpha}{2}}$. Substituting these values into Equation 4, we obtain:

$$k^{\beta} \quad = \quad h^{\frac{r+1}{4r}\cdot(\frac{r-1}{r+1}-\epsilon)} \quad = \quad h^{\frac{1}{4}-\frac{1+\epsilon(r+1)}{4r}} \quad = \quad h^{1/4-\epsilon'} \qquad \text{where } \epsilon' = \frac{1+\epsilon(r+1)}{4r}$$

Similarly, we have

$$\left(\frac{k^{\beta+1-r}}{h^{(1+\alpha-r)/(1+\alpha)}} \right)^{1/r} \quad = \quad h^{\frac{1}{4}-\frac{1+\epsilon(r+1)/r}{4r}} \quad = \quad h^{1/4-\epsilon''} \qquad \text{where } \epsilon'' = \frac{1+\epsilon(r+1)/r}{4r}$$

Setting ϵ' sufficiently small and r sufficiently large (but independent of n) yields that the distinguishing ratio d is at least $h^{\frac{1}{4}-\epsilon}$. Finally, since the number of vertices in G is $\Theta(h)$ (with high probability), Lemma 21 completes the proof. □

Observe that the Min-Color Path instances G constructed in Lemmata 26 and 25 are planar and have treewidth two. One can make color-connected (but non-planar) equivalent instances by adding a single vertex v^* with color set $\sigma(v^*) = [m]$ and connecting it to all other vertices. This only increases treewidth of G by 1. We summarize our results in the following theorem.

Theorem 2. *Assuming* Dense vs Random *conjecture* [CDM17]*, one cannot approximate* Min-Color Path *within ratio $O(m^{1-\epsilon})$ or $O(n^{1/4-\epsilon})$ in polynomial time, for any $\epsilon > 0$, where m is the number of colors and n is the number of vertices in G. The bounds hold even on the following two restricted classes of instances.*

1. *G is a diamond path.*
2. *G has a vertex v so that $G - v$ is a diamond path and (G, σ) is color-connected.*

8 Conclusion

In this paper we gave the first constant factor approximation algorithm for the Min-Color Path problem on color-connected planar graphs, answering an open question posed in [BKSV18, CK14]. This algorithm immediately yields a constant factor approximation algorithm for the Barrier Resilience and Minimum Constraint Removal problems. In fact we obtained a constant factor approximation for a substantially more general Minimum Color Prize Collecting Steiner Forest version of the problem, which generalizes classic Steiner Forest and Prize-Collecting Steiner Forest problems on planar graphs. We complemented our algorithmic findings by showing that neither the assumption on planarity nor the connectivity of colors can be dropped from our results; either of these would lead to strong inapproximability results.

We believe that Theorem 4 together with the methods for rounding algorithm for Hitting-LP will find further applications for obstacle removal problems. It is interesting both to pin down the "right" constant factor approximation ratio for Min-Color Path for planar color connected graphs, as well as evaluate the performance of heuristic algorithms for Min-Color Path based on rounding Hitting-LP on practical instances.

References

[BCE+11] MohammadHossein Bateni, Chandra Chekuri, Alina Ene, Mohammad Taghi Hajiaghayi, Nitish Korula, and Dániel Marx. Prize-collecting steiner problems on planar graphs. In *Proc. of 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1028–1049, 2011. 3

[BDHM16] MohammadHossein Bateni, Erik D. Demaine, MohammadTaghi Hajiaghayi, and Dániel Marx. A PTAS for planar group steiner tree via spanner bootstrapping and prize collecting. In *Proc. of 48th Annual Symposium on Theory of Computing*, pages 570–583. ACM, 2016. 3

[BHM11] MohammadHossein Bateni, Mohammad Taghi Hajiaghayi, and Dániel Marx. Approximation schemes for steiner forest on planar graphs and graphs of bounded treewidth. *J. ACM*, 58(5):21:1–21:37, 2011. 3

[BK09] Sergey Bereg and David G. Kirkpatrick. Approximating barrier resilience in wireless sensor networks. In *Proc. of ALGOSENSORS*, volume 5804, pages 29–40, 2009. 1

[BKK07] Glencora Borradaile, Claire Kenyon-Mathieu, and Philip N. Klein. A polynomial-time approximation scheme for steiner tree in planar graphs. In *Proc. of 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1285–1294, 2007. 3

[BKM09] Glencora Borradaile, Philip N. Klein, and Claire Mathieu. An $O(n \log n)$ approximation scheme for steiner tree in planar graphs. *ACM Trans. Algorithms*, 5(3):1–31, 2009. 3

[BKSV18] Sayan Bandyapadhyay, Neeraj Kumar, Subhash Suri, and Kasturi R. Varadarajan. Improved approximation bounds for the minimum constraint removal problem. In *Proc. of APPROX/RANDOM*, pages 2:1–2:19, 2018. 1, 2, 22

[CDKM00] Robert D. Carr, Srinivas Doddi, Goran Konjevod, and Madhav V. Marathe. On the red-blue set cover problem. In *Proc. of 11th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 345–353, 2000. 1

[CDM17] Eden Chlamtác, Michael Dinitz, and Yury Makarychev. Minimizing the union: Tight approximations for small set bipartite vertex expansion. In *Proc. of 28th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 881–899, 2017. 1, 2, 5, 18, 19, 22

[CG16] Sergio Cabello and Panos Giannopoulos. The complexity of separating points in the plane. *Algorithmica*, 74(2):643–663, 2016. 4, 7

[CK12] David Yu Cheng Chan and David G. Kirkpatrick. Approximating barrier resilience for arrangements of non-identical disk sensors. In *8th International Symposium on Algorithms for Sensor Systems*, pages 42–53, 2012. 1, 2

[CK14] David Yu Cheng Chan and David G. Kirkpatrick. Multi-path algorithms for minimum-colour path problems with applications to approximating barrier resilience. *Theor. Comput. Sci.*, 553:74–90, 2014. 1, 2, 22

[DHK14] Erik D. Demaine, Mohammad Taghi Hajiaghayi, and Philip N. Klein. Node-weighted steiner tree and group steiner tree in planar graphs. *ACM Trans. Algorithms*, 10(3):13:1–13:20, 2014. 3

[EGKY18] Eduard Eiben, Jonathan Gemmell, Iyad A. Kanj, and Andrew Youngdahl. Improved results for minimum constraint removal. In *Proc. of 32nd Conference on Artificial Intelligence (AAAI)*, pages 6477–6484. AAAI Press, 2018. 1

[EK20] Eduard Eiben and Iyad A. Kanj. A colored path problem and its applications. *ACM Trans. Algorithms*, 16(4):47:1–47:48, 2020. 1, 2

[EKM12] David Eisenstat, Philip N. Klein, and Claire Mathieu. An efficient polynomial-time approximation scheme for steiner forest in planar graphs. In *Proc. of 23rd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 626–638, 2012. 3

[EL13] Lawrence H. Erickson and Steven M. LaValle. A simple, but NP-Hard, motion planning problem. In *Proc. of 27th Conference on Artificial Intelligence (AAAI)*, 2013. 1

[EL20] Eduard Eiben and Daniel Lokshtanov. Removing connected obstacles in the plane is FPT. In *36th International Symposium on Computational Geometry (SoCG)*, volume 164, pages 39:1–39:14, 2020. 1

[Hau14] Kris K. Hauser. The minimum constraint removal problem with three robotics applications. *I. J. Robotics Res.*, 33(1):5–17, 2014. 1, 2

[HKKN12] Mohammad Taghi Hajiaghayi, Rohit Khandekar, Guy Kortsarz, and Zeev Nutov. Prize-collecting steiner network problems. *ACM Trans. Algorithms*, 9(1):2:1–2:13, 2012. 3

[HMS07] Refael Hassin, Jérôme Monnot, and Danny Segev. Approximation algorithms and hardness results for labeled connectivity problems. *J. Comb. Optim.*, 14(4):437–453, 2007. 1

[IS79] Alon Itai and Yossi Shiloach. Maximum flow in planar networks. *SIAM J. Comput.*, 8(2):135–150, 1979. 7

[JKK$^+$99] R Jacob, G Koniedov, S Krumke, M Marathe, R Ravi, and H Wirth. The minimum label path problem. *Unpublished manuscript, Los Alamos National Laboratory*, 1999. 1

[KB17] Athanasios Krontiris and Kostas E. Bekris. Trade-off in the computation of minimum constraint removal paths for manipulation planning. *Adv. Robotics*, 31:1313–1324, 2017. 1

[KLA07] Santosh Kumar, Ten-Hwang Lai, and Anish Arora. Barrier coverage with wireless sensors. *Wirel. Networks*, 13(6):817–834, 2007. 1

[KLSS18] Matias Korman, Maarten Löffler, Rodrigo I. Silveira, and Darren Strash. On the complexity of barrier resilience for fat regions and bounded ply. *Comput. Geom.*, 72:34–51, 2018. 1

[KMS18] Subhash Khot, Dor Minzer, and Muli Safra. Pseudorandom sets in grassmann graph have near-perfect expansion. In *59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 592–601, 2018. 2

[KPR93] Philip N. Klein, Serge A. Plotkin, and Satish Rao. Excluded minors, network decomposition, and multicommodity flow. In *Proc. of 25th Annual ACM Symposium on Theory of Computing*, pages 682–690, 1993. 5, 9, 10

[KR08] Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within $2 - \epsilon$. *J. Comput. Syst. Sci.*, 74(3):335–349, 2008. 2

[Kum19] Neeraj Kumar. Computing a minimum color path in edge-colored graphs. In *Int. Symposium on Experimental Algorithms (SEA)*, Lecture Notes in Computer Science, pages 35–50, 2019. 1

[Lee17] James R. Lee. Separators in region intersection graphs. In *8th Innovations in Theoretical Computer Science Conference (ITCS)*, volume 67, pages 1–8, 2017. 4, 5, 9, 10

[LR99] Frank Thomson Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM*, 46(6):787–832, 1999. 4, 5, 9

[PS82] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Prentice-Hall, 1982. 4

[TK11] Kuan-Chieh Robert Tseng and David Kirkpatrick. On barrier resilience of sensor networks. In *International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics*, pages 130–144. Springer, 2011. 1

[XM16] Bo Xu and Huaqing Min. Solving minimum constraint removal (MCR) problem using a social-force-model-based ant colony algorithm. *Appl. Soft Comput.*, 43:553–560, 2016. 1