Multi-colored Spanning Graphs

Hugo A. Akitaya[1], Maarten Löffler[2], and Csaba D. Tóth[1,3](✉)

[1] Tufts University, Medford, MA, USA
hugo.alves_akitaya@tufts.edu
[2] Utrecht University, Utrecht, The Netherlands
m.loffler@uu.nl
[3] California State University Northridge, Los Angeles, CA, USA
csaba.toth@csun.edu

Abstract. We study a problem proposed by Hurtado et al. [10] motivated by sparse set visualization. Given n points in the plane, each labeled with one or more primary colors, a *colored spanning graph* (CSG) is a graph such that for each primary color, the vertices of that color induce a connected subgraph. The MIN-CSG problem asks for the minimum sum of edge lengths in a colored spanning graph. We show that the problem is NP-hard for k primary colors when $k \geq 3$ and provide a $(2 - \frac{1}{3+2\varrho})$-approximation algorithm for $k = 3$ that runs in polynomial time, where ϱ is the Steiner ratio. Further, we give a $O(n)$ time algorithm in the special case that the input points are collinear and k is constant.

1 Introduction

Visualizing set systems is a basic problem in data visualization. Among the oldest and most popular set visualization tools are the Venn and Euler diagrams. However, other methods are preferred when the data involves a large number of sets with complex intersection relations [2]. In particular, a variety of tools have been proposed for set systems where the elements are associated with location data. Many of these methods use geometric graphs to represent set membership, motivated by reducing the amount of ink used in the representation, including LineSets [1], Kelp Diagrams [7] and KelpFusion [11].

Hurtado et al. [10] recently proposed a method for drawing sets using outlines that minimise the total visual clutter. The underlying combinatorial problem is to compute a *minimum colored spanning graph*; see Fig. 1. They studied the problem for n points in a plane and two sets (each point is a member of one or both sets). The output is a graph with the minimum sum of edge lengths such that the subgraph induced by each set is connected. They gave an algorithm that runs in $O(n^6)$-time,[1] and a $(\frac{1}{2}\varrho + 1)$-approximation in $O(n \log n)$ time, where ϱ is the Steiner ratio (the ratio between the length of a minimum spanning tree and the length of a minimum Steiner tree). Efficient algorithms are known in two special cases: One runs in $O(n)$ time for collinear points that are already sorted [10]; the other runs in $O(m^2 + n)$ time for cocircular points, where m is

[1] An earlier claim that the problem was NP-hard [9] turned out to be incorrect [10].

Y. Hu and M. Nöllenburg (Eds.): GD 2016, LNCS 9801, pp. 81–93, 2016.
DOI: 10.1007/978-3-319-50106-2_7

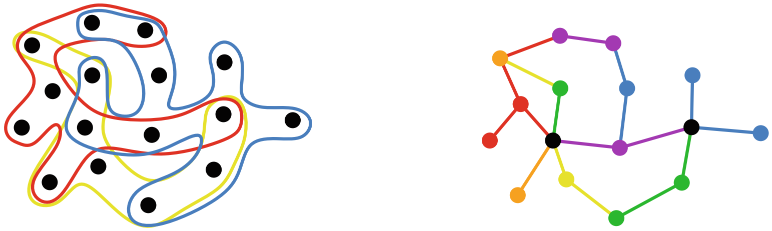


Fig. 1. Left: A set of points and three subsets, S_1, S_2, and S_3, drawn as outlines in different colors. Right: The corresponding (minimum) colored spanning graph. Refer to Sect. 2 for an explanation of color use. (Color figure online)

the number of points that are elements of both sets [5]. This problem also has applications for connecting different networks with minimum cost, provided that edges whose endpoints belong to both networks can be shared.

Results and Organization. We study the minimum colored spanning graph problem for n points in a plane and k sets, $k \geq 3$. The formal definition and some properties of the optimal solution are in Sect. 2. In Sect. 3, we show that Min-kCSG is NP-complete for all $k \geq 3$, and in Sect. 4 we provide an $(2 - \frac{2}{2+2\varrho})$-approximation algorithm for $k = 3$ that runs in $O(n \log n + m^6)$ time, where m is the number of multichromatic points. This improves the previous $(2 + \frac{\varrho}{2})$-approximation from [10]. Section 5 describes an algorithm for the special case of collinear points that runs in $2^{O(k^2 2^k)} \cdot n$ time. Due to space constraints, some proofs are omitted; they can be found in the full version of this paper.

2 Preliminaries

In this section, we define the problem and show a property of the optimal solution related to the minimum spanning trees, which is used in Sects. 3 and 4.

Definitions. Given a set of n points in the plane $S = \{p_1, \ldots, p_n\}$ and subsets $S_1, \ldots, S_k \subseteq S$, we represent set membership with a function $\alpha : S \rightarrow 2^{\{1,\ldots,k\}}$, where $p \in S_c$ iff $c \in \alpha(p)$ for every *primary color* $c \in \{1, \ldots, k\}$. We call $\alpha(p)$ the *color* of point p. A point p is *monochromatic* if it is a member of a single set S_i, that is, $|\alpha(p)| = 1$, and *multi-chromatic* if $|\alpha(p)| > 1$. For an edge $\{p_i, p_j\} \in E$ in a graph $G = (S, E)$, we use the shorthand notation $\alpha(\{p_i, p_j\}) = \alpha(p_i) \cap \alpha(p_j)$ for the shared primary colors of the two vertices. For every $c \in \{1, \ldots, k\}$, we let $G_c = (S_c, E_c)$ denote the subgraph of $G = (S, E)$ induced by S_c. All figures in this paper depict only three primary colors: $\mathtt{r}$, $\mathtt{b}$, and $\mathtt{y}$ for red, blue, and yellow respectively. Multi-chromatic points and edges are shown green, orange, purple, or black if their color is $\{\mathtt{b}, \mathtt{y}\}$, $\{\mathtt{r}, \mathtt{y}\}$ or $\{\mathtt{r}, \mathtt{b}\}$, or $\{\mathtt{r}, \mathtt{b}, \mathtt{y}\}$, respectively. See, for example, Fig. 1 (b).

A *colored spanning graph* for the pair (S, α), denoted CSG(S, α), is a graph $G = (S, E)$ such that (S_c, E_c) is connected for every primary color $c \in \{1, \ldots, k\}$. The *minimum colored spanning graph* problem (Min-CSG), for a given pair

(S, α), asks for the minimum cost $\sum_{e \in E} w(e)$ of a CSG(S, α), where $w(e)$ is the Euclidean length of e. When we wish to emphasize the number k of primary colors, we talk about the Min-kCSG problem.

Monochromatic Edges in a Minimum CSG. The following lemma shows that we can efficiently compute some of the monochromatic edges of a minimum CSG for an instance (S, α) using the *minimum spanning tree* (*MST*) of S_c for every primary color $c \in \{1, \ldots, k\}$.

Lemma 1. *Let (S, α) be an instance of* Min-CSG *and $c \in \{1, \ldots, k\}$. Let $E(MST(S_c))$ be the edge set of an MST of S_c, and let S'_c be the set of multi-chromatic points in S_c. Then there exists a minimum CSG that contains at least $|E(MST(S_c))| - |S'_c| + 1$ edges of $E(MST(S_c))$. The common edges of $E(MST(S_c))$ and of such a minimum CSG can be computed in $O(n \log n)$ time.*

Proof. Construct a monochromatic subset $E'_c \subset E(\mathrm{MST}(S_c))$ by successively removing a longest edge from the path in $\mathrm{MST}(S_c)$ between any two points in S'_c. An $\mathrm{MST}(S_c)$ can be computed in $O(n \log n)$ time, and E'_c can be obtained in $O(n)$ time. The graph (S_c, E'_c) has $|S'_c|$ components, each containing one element of S'_c, hence $|E'_c| = |E(\mathrm{MST}(S_c))| - |S'_c| + 1$.

Let (S, E^{OPT}) be a minimum CSG. While there is an edge $e \in E'_c \setminus E^{\mathrm{OPT}}$, we can find an edge $e^* \in E^{\mathrm{OPT}} \setminus E'_c$ such that exchanging e^* for e yields another minimum CSG. Indeed, since $(S_c, E_c^{\mathrm{OPT}})$ is connected, the insertion of the edge e creates a cycle C that contains e. Consider the longest (open or closed) path $P \subseteq C$ that is monochromatic and contains e. Note that at least one of the endpoints of e is monochromatic, therefore P contains at least two monochromatic edges. Since every component of (S_c, E'_c) is a tree and contains only one multi-chromatic point, there is a monochromatic edge $e^* \in E^{\mathrm{OPT}} \setminus E'_c$ in P. We have $w(e) \leq w(e^*)$, because there is a cut of the complete graph on S_c that contains both e and e^*, and $e \in E(\mathrm{MST}(S_c))$. Since $\alpha(e^*) = c$, the deletion of e^* can only influence the connectivity of the induced subgraph $(S_c, E_c^{\mathrm{OPT}})$. Consequently, $(S, E^{\mathrm{OPT}} \cup \{e\} \setminus \{e^*\})$ is a CSG with equal or lower cost than (S, E^{OPT}). By successively exchanging the edges in $E'_c \setminus E^{\mathrm{OPT}}$, we obtain a minimal CSG containing E'_c. □

Hurtado et al. [10] gave an $O(n^6)$-time algorithm for Min-2CSG, by a reduction to a matroid intersection problem on the set of all possible edges on S, which has $O(n^2)$ elements. Their algorithm for matroid intersection finds $O(n^2)$ single source shortest paths in a bipartite graph with $O(n^2)$ vertices and $O(n^4)$ edges, which leads to an overall running time of $O(n^6)$. We improve the runtime to $O(n \log n + m^6)$, where m is the number of multi-chromatic points.

Corollary 1. *An instance (S, α) of* Min-2CSG *can be solved in $O(n \log n + m^6)$ time, where m is the number of multi-chromatic points in S.*

Proof. By Lemma 1, we can compute two spanning forests on S_1 and S_2, respectively, each with m components, that are subgraphs of a minimum CSG in $O(n \log n)$ time. It remains to find edges of minimum total length that connect these components in each color, for which we can use the same matroid intersection algorithm as in [10], but with a ground set of size $O(m^2)$. □

3 General Case

We show that the decision version of Min-CSG is NP-complete. We define the decision version of Min-CSG as follows: given an instance (S, α) and $W > 0$, is there a CSG (S, E) such that $\sum_{e \in E} w(e) < W$?

Lemma 2. Min-kCSG *is in NP.*

Proof. Given a set of edges E, we can verify if (S, E) is a $CSG(S, \alpha)$ in $O(k|S|)$ time by testing connectivity in (S_c, E_c) for each primary color $c \in \{1, \dots, k\}$, and then check whether $\sum_{e \in E} w(e) \leq W$ in $O(|E|)$ time. □

We reduce Min-3CSG from Planar-Monotone-3SAT, which is known to be NP-complete [4]. For every instance A of Planar-Monotone-3SAT, we construct an instance $f(A)$ of Min-3CSG. An instance A consists of a plane bipartite graph between n variable and m clause vertices such that every clause has degree three or two, all variables lie on the x-axis and edges do not cross the x-axis. Clauses are called *positive* if they are in the upper half-plane or *negative* otherwise. The problem asks for an assignment from the variable set to $\{\texttt{true}, \texttt{false}\}$ such that each positive (negative) clause is adjacent to a `true` (`false`) variable.

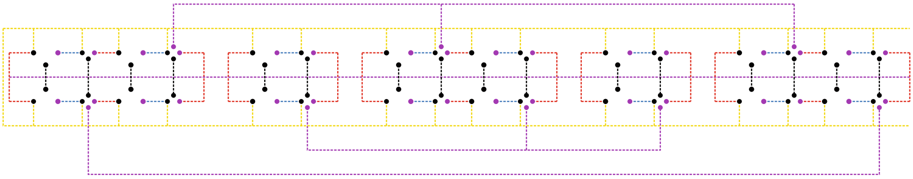
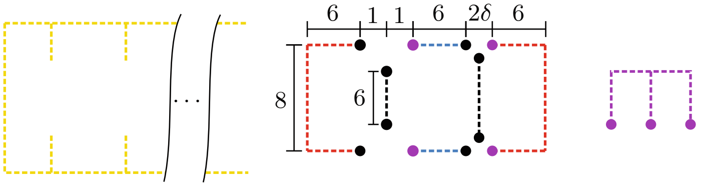


Fig. 2. Construction for an instance A equivalent to the boolean formula $(x_1 \vee x_3 \vee x_5) \wedge (\neg x_1 \vee \neg x_5) \wedge (\neg x_2 \vee \neg x_3 \vee \neg x_4)$. (Color figure online)

Given an instance A of Planar-Monotone-3SAT, we construct $f(A)$ as shown in Fig. 2 (refer to the full paper for a figure of a single variable gadget). The points marked with small disks are called *active* and they are the only multi-chromatic points in the construction. The dashed lines in a primary color represent a chain of equidistant monochromatic points, where the gap between consecutive points is ε. A purple (resp., black) dashed line represents a red and a blue (resp., a red, a blue, and a yellow) dashed line that run ε close to each other. Informally, the value of ε is set small enough such that every point in the interior of a dashed line is adjacent to its neighbors in any minimum CSG. The boolean assignment of A is encoded in the edges connecting active points. We break the construction down to gadgets and explain their behavior individually.

The long horizontal purple dashed line is called *spine* and the set of yellow dashed lines (shown in Fig. 3(a)) is called *cage*. The rest of the construction consists of variable and clause gadgets (shown in Figs. 3(b) and (c)). The width



(a) (b) (c)

Fig. 3. (a) Cage. (b) Variable gadget. (c) Clause gadget. (Color figure online)

of a variable gadget depends on the degree of the corresponding variable in the bipartite graph given by the instance A. For every edge incident to the variable, we repeat the middle part of the gadget as shown in Fig. 3(b) (see Fig. 2 for variables of degree 1 and 2). The vertical black dashed lines are called *ribs* and the set of three or four active points close to an endpoint of a rib is called *switch*. The variable gadget contains switches of two different sizes alternately from left to right. A *2-switch* (resp., *2δ-switch*) is a switch in which active points are at most 2 (resp., 2δ) apart. The clause gadgets are positioned as the embedding of clauses in A; refer to Fig. 2. Each active point of a positive (negative) clause is assigned to a 2δ-switch and positioned vertically above (below) the active point of the rib, at distance 2δ from it.

Let E' be the set of all monochromatic edges of a minimum CSG computable by Lemma 1. Let r be the number of edges in the bipartite graph of A. The instance $f(A)$ contains $13r$ active points, so (S, E') contains $13r$ connected components. By construction, the number of ε-edges in a solution of $f(A)$ between components of (S, E') is upper bounded by $39r$ (one edge per color per component). Finally, we set $W = (\sum_{e \in E'} w(e)) + 39r\varepsilon + r(2 + 2\sqrt{2}) + r\delta(2 + 2\sqrt{2}) + m\delta(2\sqrt{2} - 2)$ and we choose $\varepsilon = \frac{1}{500r^2}$ and $\delta = \frac{1}{10r}$. This particular choice of ε and δ is justified by the proofs of Corollaries 2 and 3. By construction, $f(A)$ has the following property:

(I) For every partition of the components of (S_c, E'_c) into two sets C_1, C_2, where c is a primary color, let $\{p_1, p_2\}$ be the shortest edge between C_1 and C_2. Then either $w(\{p_1, p_2\}) = \varepsilon$ or p_1 and p_2 are active points in the same switch.

Definition 1. *A* standard solution *of* Min-3CSG *is a solution that contains E' and in which every edge longer than ε is between two active points of the same switch.*

Lemma 3. *Let A be a positive instance of* Planar-Monotone-3SAT. *Then $f(A)$ is a positive instance of* Min-3CSG.

To prove the lemma, we construct a standard solution for $f(A)$ based on the solution for A. This proof, and subsequent proofs, argues about all possible ways to connect the vertices in a switch of $f(A)$. The most efficient ones are shown in Fig. 4; these may appear in an optimal solution. Refer to the full paper for a complete list.

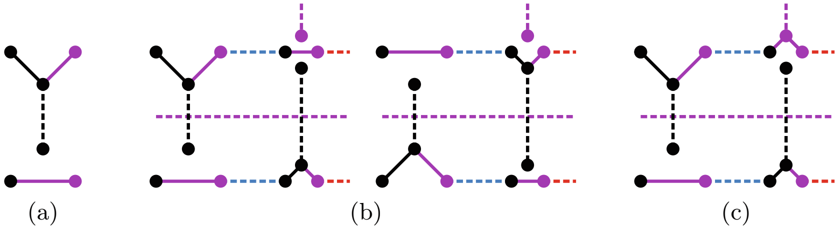


Fig. 4. Possible ways to connect the vertices in a switch of $f(A)$. (a) One of the two states of a 2-switch, encoding the truth value of the variable. (b) The two possible states of a 2δ-switch if the incident clause is not satisfied through this variable. (c) The only possible state of a 2δ-switch if the incident clause is satisfied through this variable. (Color figure online)

Lemma 4. *If $f(A)$ is a positive instance of* MIN-3CSG*, there exists a standard solution for this instance.*

Before proving the other direction of the reduction, we show some properties of a standard solution. The active points in a switch impose some local constraints. The black and purple points attached to horizontal dashed lines determine the *switch constraint*: since these points have more colors than their incident dashed lines, they each are incident to at least one edge in the switch. Each rib determines a *rib constraint* to a pair of switches that contain its endpoints: at least one of these switches must contain an edge between its black active points or else there is no yellow path between this rib and the cage. The following lemmas establish some bounds on the length of the edges used to satisfy local constraints of a pair of switches adjacent to a rib. We refer to this pair as a 2-pair or 2δ-pair according to the type of the switch.

Lemma 5. *In a standard solution, the minimum length required to satisfy the local constraints of a 2-pair (resp., 2δ-pair) is $2(1+\sqrt{2})$ (resp., $2\delta(1+\sqrt{2})$).*

Corollary 2. *In a standard solution, every 2-pair is connected minimally.*

Lemma 6. *In a standard solution, for each clause gadget, there exists a 2δ-pair with local cost at least $4\delta\sqrt{2}$.*

Corollary 3. *In a standard solution, for each clause gadget, there exists a 2δ-pair connected as Fig. 4(c). All other 2δ-pairs are connected minimally as shown in Fig. 4(b).*

Lemma 7. *Let $f(A)$ be a positive instance of* MIN-3CSG*. Then A is a positive instance of* PLANAR-MONOTONE-3SAT*.*

The following theorem is a direct consequence of Lemmata 2, 3, and 7.

Theorem 1. MIN-kCSG *is NP-complete for $k \geq 3$.*

4 Approximation

Hurtado et al. [10] gave an approximation algorithm for Min-kCSG that runs in $O(n \log n)$ time and achieves a ratio of $\lceil k/2 \rceil + \lfloor k/2 \rfloor \varrho/2$, where ϱ is the Steiner ratio. The value of ϱ is not known and the current best upper bound is $\varrho \le 1.21$ by Chung and Graham [6] (Gilbert and Pollack [8] conjectured $\varrho = \frac{2}{\sqrt{3}} \approx 1.15$). For the special case $k = 3$, the previous best approximation ratio is $2+\varrho/2 \le 2.6$. We improve the approximation ratio to 2, and then further to 1.81. Our first algorithm immediately generalises to $k \ge 3$, and yields an $\lceil k/2 \rceil$-approximation, improving on the general result by Hurtado et al.; our second algorithm also generalizes to $k > 3$, however, we do not know whether it achieves a good ratio.

Suppose we are given an instance of Min-3CSG defined by (S, α) where $|S| = n$ and the set of primary colors is $\{\mathtt{r},\mathtt{b},\mathtt{y}\}$. We define $\alpha_{\mathtt{rb}} : S_{\mathtt{r}} \cup S_{\mathtt{b}} \to 2^{\{r,b\}} \setminus \{\emptyset\}$ where $\alpha_{\mathtt{rb}}(p) = \alpha(p) \setminus \{\mathtt{y}\}$. Let G^* be an optimal solution for Min-3CSG, and put $\mathrm{OPT} = \|G^*\|$. Algorithm A1 computes a minimum red-blue-purple graph $G_{\mathtt{rb}} = CSG(S_{\mathtt{r}} \cup S_{\mathtt{b}}, \alpha_{\mathtt{rb}})$ in $O(n \log n + m^6)$ time, where $m = |S_{\mathtt{r}} \cap S_{\mathtt{b}}|$ by Corollary 1; then computes a minimum spanning tree $G_{\mathtt{y}}$ of $S_{\mathtt{y}}$, and returns the union $G_{\mathtt{rb}} \cup G_{\mathtt{y}}$. Since G^* contains a red, a blue, and a yellow spanning tree, we have $\|G_{\mathtt{rb}}\| \le \mathrm{OPT}$ and $\|G_{\mathtt{y}}\| \le \mathrm{OPT}$; that is, Algorithm A1 returns a solution to Min-3CSG whose length is at most 2OPT.

Theorem 2. *Algorithm A1 returns a 2-approximation for* Min-3CSG*; it runs in $O(n \log n + m^6)$ time on n points, m of which are multi-chromatic.*

Algorithm A1 can be extended to k colors by partitioning the primary colors into $\lceil \frac{k}{2} \rceil$ groups of at most two and computing the minimum CSG for each group. The union of these graphs is a $\lceil \frac{k}{2} \rceil$-approximation that can be computed in $O(kn^6)$ time.

Algorithm A2 computes six solutions for a given instance of Min-3CSG, $G_1, \ldots, G_6$, and returns one with minimum weight. Graph G_1 is the union of $G_{\mathtt{rb}}$ and $G_{\mathtt{y}}$ defined above. Graphs G_2 and G_3 are defined analogously: $G_2 = G_{\mathtt{ry}} \cup G_{\mathtt{b}}$ and $G_3 = G_{\mathtt{by}} \cup G_{\mathtt{r}}$, each of which can be computed in $O(n^6)$ time by [10]. Let $S_{\mathtt{rby}} \subseteq S$ be the set of "black" points that have all three colors, and let H be an MST of $S_{\mathtt{rby}}$, which can be computed in $O(n \log n)$ time. We augment H into a solution of Min-3CSG in three different ways as follows. First, let $G_{\mathtt{rb}:H}$ be the minimum forest such that $H \cup G_{\mathtt{rb}:H}$ is a minimum red-blue-purple spanning graph on $S_{\mathtt{r}} \cup S_{\mathtt{b}}$. $G_{\mathtt{rb}:H}$ can be computed in $O(n \log n + m^6)$ time by the same matroid intersection algorithm as in Corollary 1, by setting the weight of any edge between components containing black points to zero. Similarly, let $G_{\mathtt{y}:H}$ be the minimum forest such that $H \cup G_{\mathtt{y}:H}$ is a spanning tree on $S_{\mathtt{y}}$, which can be computed in $O(n \log n)$ time by Prim's algorithm. Now we let $G_4 = H \cup G_{\mathtt{rb}:H} \cup G_{\mathtt{y}:H}$. Similarly, let $G_5 = H \cup G_{\mathtt{ry}:H} \cup G_{\mathtt{b}:H}$ and $G_6 = H \cup G_{\mathtt{by}:H} \cup G_{\mathtt{r}:H}$.

Theorem 3. *Algorithm A2 returns a $(2-\frac{1}{3+2\varrho})$-approximation for* Min-3CSG*; it runs in $O(n \log n + m^6)$ time on an input of n points, m of which are multi-chromatic.*

Proof. Consider an instance (S, α) of MIN-3CSG, and let $G^* = (S, E^*)$ be an optimal solution with $\|E^*\| = \mathrm{OPT}$. Partition E^* into 7 subsets: for every color $\gamma \in 2^{\{\mathtt{r},\mathtt{b},\mathtt{y}\}} \setminus \emptyset$, let $E^*_\gamma = \{e \in E^* : \alpha(e) = \gamma\}$, that is E^*_γ is the set of edges of color γ in G^*. Put $\beta = \|E^*_{\mathtt{rby}}\|/\mathrm{OPT}$. Then we have $2(1-\beta)\mathrm{OPT} = (2\|E^*_{\mathtt{r}}\| + \|E^*_{\mathtt{rb}}\| + \|E^*_{\mathtt{ry}}\|) + (2\|E^*_{\mathtt{b}}\| + \|E^*_{\mathtt{rb}}\| + \|E^*_{\mathtt{by}}\|) + (2\|E^*_{\mathtt{y}}\| + \|E^*_{\mathtt{ry}}\| + \|E^*_{\mathtt{by}}\|)$. Without loss of generality, we may assume $2\|E^*_{\mathtt{y}}\| + \|E^*_{\mathtt{ry}}\| + \|E^*_{\mathtt{by}}\| \le \frac{2}{3}(1-\beta)\mathrm{OPT}$.

First, consider $G_1 = G_{\mathtt{rb}} \cup G_{\mathtt{y}}$. The edges of G^* whose colors include red or blue (resp., yellow) form a connected graph on $S_{\mathtt{r}} \cup S_{\mathtt{b}}$ (resp., $S_{\mathtt{y}}$). Consequently,

$$\|G_{\mathtt{rb}}\| \le \|E^*_{\mathtt{r}}\| + \|E^*_{\mathtt{b}}\| + \|E^*_{\mathtt{rb}}\| + \|E^*_{\mathtt{ry}}\| + \|E^*_{\mathtt{by}}\| + \|E^*_{\mathtt{rby}}\|. \tag{1}$$

$$\|G_{\mathtt{y}}\| \le \|E^*_{\mathtt{y}}\| + \|E^*_{\mathtt{ry}}\| + \|E^*_{\mathtt{by}}\| + \|E^*_{\mathtt{rby}}\|. \tag{2}$$

The combination of (1) and (2) yields

$$\begin{aligned}\|G_1\| &\le \|G_{\mathtt{rb}}\| + \|G_{\mathtt{y}}\| \le \mathrm{OPT} + \|E^*_{\mathtt{ry}}\| + \|E^*_{\mathtt{by}}\| + \|E^*_{\mathtt{rby}}\| \\ &\le \mathrm{OPT} + \frac{2}{3}(1-\beta)\cdot\mathrm{OPT} + \beta\cdot\mathrm{OPT} = \frac{5+\beta}{3}\mathrm{OPT}.\end{aligned} \tag{3}$$

Next, consider $G_4 = H \cup G_{\mathtt{rb}:H} \cup G_{\mathtt{y}:H}$. The edges of G^* whose colors include yellow contain a spanning tree on $S_{\mathtt{y}}$, hence a Steiner tree on the black points $S_{\mathtt{rby}}$. Specifically, the black edges in $E^*_{\mathtt{rby}}$ form a black spanning forest, which is completed to a Steiner tree by some of the edges of $E^*_{\mathtt{y}} \cup E^*_{\mathtt{by}} \cup E^*_{\mathtt{ry}}$. This implies

$$\begin{aligned}\|H\| &\le \|E^*_{\mathtt{rby}}\| + \varrho\cdot(\|E^*_{\mathtt{y}}\| + \|E^*_{\mathtt{by}}\| + \|E^*_{\mathtt{ry}}\|) \\ &\le \beta\cdot\mathrm{OPT} + \varrho\frac{2}{3}(1-\beta)\cdot\mathrm{OPT} = \left(\beta + \frac{2}{3}\varrho - \frac{2}{3}\beta\varrho\right)\mathrm{OPT}.\end{aligned}$$

Since H is a spanning tree on the black vertices S_{rby}, (1) and (2) reduce to

$$\|G_{\mathtt{rb}:H}\| \le \|E^*_{\mathtt{r}}\| + \|E^*_{\mathtt{b}}\| + \|E^*_{\mathtt{rb}}\| + \|E^*_{\mathtt{ry}}\| + \|E^*_{\mathtt{by}}\|, \tag{4}$$

$$\|G_{\mathtt{y}:H}\| \le \|E^*_{\mathtt{y}}\| + \|E^*_{\mathtt{ry}}\| + \|E^*_{\mathtt{by}}\|. \tag{5}$$

The combination of (4) and (5) yields

$$\begin{aligned}\|G_{\mathtt{rb}:H}\| + \|G_{\mathtt{y}:H}\| &\le (\mathrm{OPT} - \|E^*_{\mathtt{rby}}\|) + \|E^*_{\mathtt{ry}}\| + \|E^*_{\mathtt{by}}\| \\ &\le (1-\beta)\cdot\mathrm{OPT} + \frac{2}{3}(1-\beta)\cdot\mathrm{OPT} = \frac{5}{3}(1-\beta)\cdot\mathrm{OPT}.\end{aligned}$$

Therefore,

$$\|G_4\| = \|H\| + \|G_{\mathtt{rb}:H}\| + \|G_{\mathtt{y}:H}\| \le \left(\frac{5}{3} + \frac{2}{3}(\varrho - \beta - \beta\varrho)\right)\mathrm{OPT}. \tag{6}$$

If we set $\beta = \frac{2\varrho}{3+2\varrho}$, then both (3) and (6) give the same upper bound

$$\frac{\min(\|G_1\|, \|G_4\|)}{\mathrm{OPT}} \le \frac{5+\beta}{3} = 2 - \frac{1}{3+2\varrho} \le 1.816,$$

where we used the current best upper bound for the Steiner ratio $\varrho \le 1.21$ from [6]. □

5 Collinear Points

In this section we consider instances of Min-kCSG, (S, α), where $k \geq 3$ and S consists of collinear points. An example is shown in Fig. 5. Without loss of generality, $S = \{p_1, \ldots, p_n\}$ and the points p_i, $1 \leq i \leq n$, lie on the x-axis sorted by x-coordinates. We present a dynamic programming algorithm that solves Min-kCSG in $2^{O(k^2 2^k)} \cdot n$ time.

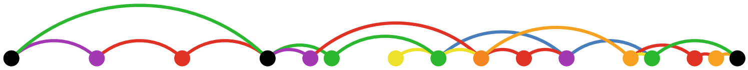


Fig. 5. An example with optimal solution for collinear points. (Color figure online)

Our first observation is that if the points in S are collinear, we may assume that every edge satisfies the following property.

> If $\{p_a, p_b\}$, $a < b$, is an edge, then there is no r, $a < r < b$, such that $\alpha(\{p_a, p_b\}) \subseteq \alpha(p_r)$. $(\star)$

Lemma 8. *For every graph $G = (S, E)$, there exists a graph $G' = (S, E')$ of the same cost that satisfies $(\star)$ and for each color $c \in \{1, \ldots, k\}$, every component of (S_c, E_c) is contained in some component of (S_c, E'_c). In particular,* Min-kCSG *has a solution with property $(\star)$.*

Proof. Let $G = (S, E)$ be a graph, and let X_G denote the set of triples $(i, j; r)$ such that $1 \leq i < r < j \leq n$, $\{p_i, p_j\} \in E$, and $\alpha(\{p_i, p_j\}) \subseteq \alpha(p_r)$. If $X_G = \emptyset$, then G satisfies $(\star)$. Suppose $X_G \neq \emptyset$. For every triple $(i, j; r) \in X_G$, successively, replace the edge $\{p_i, p_j\}$ by two edges $\{p_i, p_r\}$ and $\{p_r, p_j\}$ (i.e., subdivide edge $\{p_i, p_j\}$ at p_r). Note that $\alpha(\{p_h, p_i\}), \alpha(\{p_i, p_j\}) \subseteq \alpha(p_i)$, consequently p_i and p_j remain in the same component for each primary color $c \in \alpha(\{p_i, p_j\})$. Each step maintains the total edge length of the graph and strictly decreases X_G. After $|X_G|$ subdivision steps, we obtain a graph $G' = (S, E')$ as required. □

In the remainder of this section we assume that every edge has property $(\star)$. Furthermore, all graphs considered in this section are defined on an interval of consecutive vertices of S.

Corollary 4. *Let $G = (S, E)$ be a graph and let $i \in \{1, \ldots, n\}$.*

1. *If $e \in E$ is an edge between $\{p_1, \ldots, p_i\}$ and $\{p_{i+1}, \ldots, p_n\}$ and $\alpha(e) = \gamma$, then the endpoints of e are uniquely determined. Specifically, if $e = \{p_a, p_b\}$ with $1 \leq a \leq i < b \leq n$, then $a \in \{1, \ldots, i\}$ is the largest index such that $\gamma \subset \alpha(p_a)$, and $b \in \{i+1, \ldots, n\}$ is the smallest index such that $\gamma \subset \alpha(p_b)$.*
2. *If two edges $e_1, e_2 \in E$ overlap, then $\alpha(e_1) \neq \alpha(e_2)$.*

Proof.

(1) Suppose, to the contrary, that there is index j, $a < j \leq i$, such that $\gamma \subset \alpha(p_j)$. Then edge $\{p_a, p_b\}$ and point p_r violate $(\star)$. The case that there is some j, $i+1 \leq j < b$, leads to the same contradiction.

(2) Without loss of generality $e_1 = \{p_a, p_b\}$ and $e_2 = \{p_i, p_j\}$ with $a \leq i < b \leq j$. Then both edges e_1 and e_2 are between $\{p_1, \ldots, p_i\}$ and $\{p_{i+1}, \ldots, p_n\}$, contradicting part (1). □

The basis for our dynamic programming algorithm is that Min-kCSG has the *optimal substructure* and *overlapping substructures* properties when the points in S are collinear. We introduce some notation for defining the subproblems. For indices $1 \leq a \leq b \leq n$, let $S[a,b] = \{p_a, \ldots, p_b\}$. For every graph $G = (S,E)$ and index $i \in \{1, \ldots, n\}$, we partition the edge set E into three subsets as follows: let E_i^- be the set of edges induced by $S[1,i]$, E_i^+ the set of edges induced by $S[i+1,n]$, and E_i^0 the set of edges between $S[1,i]$ and $S[i+1,n]$. With this notation, Min-kCSG has the following optimal substructure property.

Lemma 9. *Let $G = (S,E)$ be a minimum CSG, $i \in \{1, \ldots, n\}$, and $\mathcal{X}$ be the family of edge sets X_i^- on $S[1,i]$ such that $(S, X_i^- \cup E_i^0 \cup E_i^+)$ is a CSG. Then $(S, X_i^- \cup E_i^0 \cup E_i^+)$ is a minimum CSG iff $X_i^- \in \mathcal{X}$ has minimum cost.*

Proof. If $(S, X_i^- \cup E_i^0 \cup E_i^+)$ is a minimum CSG, but some $Y_i^- \in \mathcal{X}$ costs less than E_i^-, then $(S, Y_i^- \cup E_i^0 \cup E_i^+)$ would be a CSG that costs less than $G = (S,E)$, contradicting the minimality of $(S, X_i^- \cup E_i^0 \cup E_i^+)$. If $X_i^- \in \mathcal{X}$ has minimum cost, but $G = (S,E)$ costs less than $(S, X_i^- \cup E_i^0 \cup E_i^+)$, then $E_i^- \in \mathcal{X}$ would costs less than X_i^-, contradicting the minimality of $X_i^- \in \mathcal{X}$. □

Lemma 9 immediately suggests a naïve algorithm for Min-kCSG: Guess the edge set $E_i^0 \cup E_i^+$ of a minimum CSG $G = (S,E)$, and compute a minimum-cost set X_i^- on $S[1,i]$ such that $(S, X_i^- \cup E_i^0 \cup E_i^+)$ is a CSG. However, all possible edge sets $E_i^0 \cup E_i^+$ could generate $2^{\Theta(n)}$ subproblems. We reduce the number of subproblems using the overlapping subproblem property. Instead of guessing $E_i^0 \cup E_i^+$, it is enough to guess the information relevant for finding the minimal cost X_i^- on $S[1,i]$. First, the edges in E_i^0 can be uniquely determined by the set of their colors (using Corollary 4 (1)). Second, the only useful information from E_i^+ is to tell which points in $S[1,i]$ are adjacent to the same component of $(S[i+1,n]_c, (E_i^+)_c)$, for each primary color $c \in \{1, \ldots, k\}$. This information can be summarized by k equivalence relations on the sets $(E_i^0)_1, \ldots, (E_i^0)_k$. We continue with the details.

We can encode E_i^0 by the set of its colors $\Gamma_i = \{\alpha(e) : e \in E_i^0\}$. For $i \in \{1, \ldots, n\}$, a set of edges X_i^0 between $S[1,i]$ and $S[i+1,n]$ is *valid* if there exists a CSG $G = (S,E)$ such that $X_i^0 = E_i^0$.

Lemma 10. *For $i \in \{1, \ldots, n\}$, an edge set X_i^0 between $S[1,i]$ and $S[i+1,n]$ is* valid *iff for every primary color $c \in \{1, \ldots, k\}$, there is an edge $e \in X_i^0$ such that $c \in \alpha(e)$ whenever both $S[1,i]_c$ and $S[i+1,n]_c$ are nonempty.*

We encode the relevant information from E_i^+ using k equivalence relations as follows. For every $c \in \{1, \ldots, k\}$, the components of $(S[i+1,n]_c, (E_i^+)_c)$ define an equivalence relation on $(E_i^0)_c$, which we denote by π_i^c: two edges in $(E_i^0)_c$ are related iff they are incident to the same component of $(S[i+1,n]_c, (E_i^+)_c)$. Let $\Pi_i = (\pi_i^1, \ldots, \pi_i^k)$. The equivalence relation π_i^c, in turn, determines a graph $(S[1,i]_c, E(\pi_i^c))$: two distinct vertices in $S[1,i]_c$ are adjacent iff they are incident to equivalent edges in $(E_i^0)_c$ (that is, two distinct vertices in $S[1,i]_c$ are adjacent iff they both are adjacent to the same component of $(S[i+1,n]_c, (E_i^+)_c)$). See Fig. 6 for examples of E_i^0 and Π_i. The condition that $(S, X_i^- \cup E_i^0 \cup E_i^+)$ is a CSG can now be formulated in terms of E_i^0 and Π_i (without using E_i^+ directly).

Lemma 11. *Let $G = (S, E)$ be a CSG, $i \in \{1, \ldots, n\}$, and X_i^- an edge set on $S[1,i]$. The graph $(S, X_i^- \cup E_i^0 \cup E_i^+)$ is a CSG iff the graph $(S[1,i]_c, (X_i^-)_c \cup E(\pi_i^c))$ is connected for every $c \in \{1 \ldots, k\}$.*

We can now define subproblems for Min-kCSG. For an index $i \in \{1, \ldots, n\}$, a valid set E_i^0, and equivalence relations $\Pi_i = (\pi_i^1, \ldots, \pi_i^k)$, let $\mathcal{X}(E_i^0, \Pi_i)$ be the family of edge sets X_i^- on $S[1,i]$ such that for every $c \in \{1 \ldots, k\}$, the graph $(S[1,i]_c, (X_i^-)_c \cup E(\pi_i^c))$ is connected. The subproblem $\mathbf{A}[i, E_i^0, \Pi_i]$ is to find the minimum cost of an edge set $X_i^- \in \mathcal{X}(E_i^0, \Pi_i)$.

Note that for $i = n$, $\mathbf{A}[n, \emptyset, (\emptyset, \ldots, \emptyset)]$ is the minimum cost of a CSG for an instance (S, α) of Min-kCSG. Next, we establish a recurrence relation for $\mathbf{A}[i, E_i^0, \Pi_i]$, which will allow computing $\mathbf{A}[n, \emptyset, (\emptyset, \ldots, \emptyset)]$ by dynamic programming. For $i = 1$, we have $\mathbf{A}[1, E_1^0, \Pi_1] = 0$ for any valid E_1^0 and Π_1. For all i, $1 < i \leq n$, we wish to express $\mathbf{A}[i, E_i^0, \Pi_i]$ in terms of $\mathbf{A}[i-1, E_{i-1}^0, \Pi_{i-1}]$'s for suitable E_{i-1}^0 and Π_{i-1}.

We say that two valid edge sets E_{i-1}^0 and E_i^0 are *compatible* if there exists an $X_i^- \in \mathcal{X}(E_i^0, \Pi_i)$ for some Π_i such that $E_{i-1}^0 = (X_i^- \cup E_i^0)_{i-1}^0$. We can characterize compatible edge sets as follows.

Lemma 12. *Two valid edge sets E_{i-1}^0 and E_i^0 are compatible iff every edge e in the symmetric difference of E_{i-1}^0 and E_i^0 is incident to p_i.*

For two valid compatible edge sets, E_{i-1} and E_i, and a sequence of equivalence relations Π_i, we define equivalence relations $\widehat{\Pi}_{i-1} = (\hat{\pi}_{i-1}^1, \ldots, \hat{\pi}_{i-1}^k)$ as follows. For every primary color $c \in \{1, \ldots, k\}$, let the equivalence relation $\hat{\pi}_{i-1}^c$ on $(E_{i-1}^0)_c$ be the transitive closure of the union of four equivalence relations: two edges in $(E_{i-1}^0)_c$ are related if (1) they both incident to p_i; (2) they both are in $(E_i^0)_c$ and π_i^c-equivalent; (3) they are both in $(E_i^0)_c$ and each are equivalent to some edge in $(E_i^0)_c$ that are π_i^c-equivalent; (4) one is incident to p_i and the other is in $(E_i^0)_c$ and π_i^c-equivalent to some edge in $(E_i^0)_c$ incident to π_i^c.

Lemma 13. *Let E_{i-1}^0 and E_i^0 be two valid compatible edge sets, and $\Pi_i = (\pi_i^1, \ldots, \pi_i^c)$. Let E_{i-1}^- be a set of edges on $S[1, i-1]$, and put $E = E_{i-1}^- \cup E_{i-1}^0 \cup E_i^0$. Then, $\widehat{\Pi}_{i-1}$ has the following property: $E_i^- \in \mathcal{X}(E_i^0, \Pi_i)$ if and only if*

(d1) *$E_{i-1}^- \in \mathcal{X}(E_{i-1}^0, \widehat{\Pi}_{i-1})$; and*
(d2) *if $c \in \alpha(p_i)$ and $S[1,i]_c \neq \{p_i\}$, then p_i is incident to an edge in $(E_{i-1}^0)_c$ or an edge in $(E_i^0)_c$ that is π_i^c-equivalent to some edge incident to $S[1, i-1]_c$.*

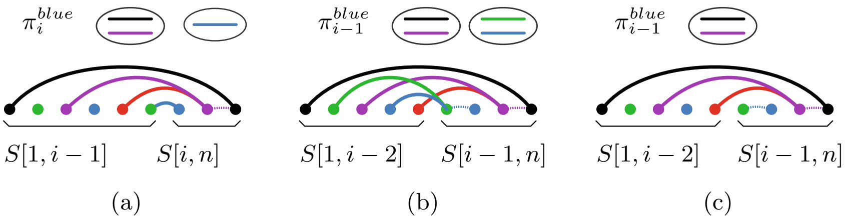


Fig. 6. (a) E_i^0 and π_i^{blue}. (b) E_{i+1}^0 and π_{i+1}^{blue}, where E_{i+1}^0 and E_i^0 are compatible. (c) E_{i+1}^0 and π_{i+1}^{blue} violate condition (**d2**). (Color figure online)

Lemma 14. *For all $i \in \{2, \ldots, n\}$, we have the following recurrence:*

$$\boldsymbol{A}[i, E_i^0, \Pi_i] = \sum_{\{p_h,p_i\}\in E_i^0} w(\{p_h, p_i\}) + \min_{E_{i-1}^0\, compatible} \boldsymbol{A}[i-1, E_{i-1}^0, \widehat{\Pi}_{i-1}]. \quad (7)$$

Theorem 4. *For every constant $k \geq 1$,* Min-kCSG *can be solved in $O(n)$ time when the input points are collinear.*

Proof. We determine the number of subproblems. By Corollary 4, every valid E_i^0 contains at most $|2^{\{1,\ldots,k\}} \setminus \{\emptyset\}| = 2^k - 1$ edges. We have $|(E_i^0)_c| \leq 2^{k-1}$, since 2^{k-1} different colors contain any primary color $c \in \{1, \ldots, k\}$. The number of equivalence relations of a set of size t is known as the t-th *Bell number*, denoted $B(t)$. It is known [3] that $B(t) \leq (0.792t/\ln(t+1))^t < 2^{O(t \log t)}$. Consequently, the number of possible Π_i is at most $(B(2^{k-1}))^k$. The total number of subproblems is $O(n2^k(B(2^{k-1}))^k)$, which is $O(n)$ for any constant k. We solve the subproblems $\mathbf{A}[i, E_i^0, \Pi_i]$, $1 < i \leq n$, by dynamic programming, using the recursive formula (7). The time required to evaluate (7) is $O(2^k)$ for the sum of edge weights and $O(2^k(B(2^{k-1}))^k)$ to compare all compatible subproblems $\mathbf{A}[i-1, E_{i-1}^0, \widehat{\Pi}_{i-1}]$, that is, $O(1)$ time when k is a constant. Therefore, the dynamic programming can be implemented in $O(n)$ time. □

6 Conclusions

We have shown that Min-3CSG is NP-complete in general and given a $O(n)$ time algorithm for Min-kCSG in the special case that all points are collinear and k is a constant. We also improved the approximation factor of a polynomial time algorithm from $(2 + \frac{1}{2}\varrho)$ [10] to $(2 - \frac{2}{2+2\varrho})$ when $k = 3$. It remains open whether there exists a PTAS for Min-kCSG, $k \geq 3$. Several other special cases are open for Min-3CSG, such as when the points in S are on a circle or in convex position. We can generalize Min-kCSG so that the edge weights need not be Euclidean distances. Given an arbitrary graph (V, E) and a coloring $\alpha : V \to \mathcal{P}(\{1, \ldots, k\})$, what is the minimum set $E' \subseteq E$ such that (V, E') is a colored spanning graph? Since the 2-approximation algorithm presented here did not rely on the geometry of the problem, it extends to the generalization; however, this problem may be harder to approximate than its Euclidean counterpart.

Acknowledgements. Research on this paper was supported in part by the NSF awards CCF-1422311 and CCF-1423615. Akitaya was supported by the Science Without Borders program. Löffler was partially supported by the Netherlands Organisation for Scientific Research (NWO) projects 639.021.123 and 614.001.504.

References

1. Alper, B., Riche, N., Ramos, G., Czerwinski, M.: Design study of linesets, a novel set visualization technique. IEEE Trans. Vis. Comput. Graph. **17**(12), 2259–2267 (2011)
2. Alsallakh, B., Micallef, L., Aigner, W., Hauser, H., Miksch, S., Rodgers, P.: Visualizing sets and set-typed data: state-of-the-art and future challenges. In: Proceedings of Eurographics Conference on Visualization (EuroVis), pp. 1–21 (2014)
3. Berend, D., Tassa, T.: Improved bounds on Bell numbers and on moments of sums of random variables. Prob. Math. Stat. **30**(2), 185–205 (2010)
4. de Berg, M., Khosravi, A.: Optimal binary space partitions for segments in the plane. Int. J. Comput. Geom. Appl. **22**(3), 187–206 (2012)
5. Biniaz, A., Bose, P., van Duijn, I., Maheshwari, A., Smid, M.: A faster algorithm for the minimum red-blue-purple spanning graph problem for points on a circle. In: Proceedings of 28th Canadian Conference on Computational Geometry, Vancouver, BC, pp. 140–146 (2016)
6. Chung, F., Graham, R.: A new bound for Euclidean Steiner minimum trees. Ann. N.Y. Acad. Sci. **440**, 328–346 (1986)
7. Dinkla, K., van Kreveld, M.J., Speckmann, B., Westenberg, M.A.: Kelp diagrams: point set membership visualization. Comput. Graph. Forum **31**(3), 875–884 (2012)
8. Gilbert, E., Pollak, H.: Steiner minimal trees. SIAM J. Appl. Math. **16**, 1–29 (1968)
9. Hurtado, F., Korman, M., van Kreveld, M., Löffler, M., Sacristán, V., Silveira, R.I., Speckmann, B.: Colored spanning graphs for set visualization. In: Wismath, S., Wolff, A. (eds.) GD 2013. LNCS, vol. 8242, pp. 280–291. Springer, Heidelberg (2013). doi:10.1007/978-3-319-03841-4_25
10. Hurtado, F., Korman, M., van Kreveld, M., Löffler, M., Sacristan, V., Shioura, A., Silveira, R.I., Speckmann, B., Tokuyama, T.: Colored spanning graphs for set visualization (2016). Preprint arXiv:1603.00580
11. Meulemans, W., Riche, N.H., Speckmann, B., Alper, B., Dwyer, T.: KelpFusion: a hybrid set visualization technique. IEEE Trans. Vis. Comput. Graph. **19**(11), 1846–1858 (2013)