J Comb Optim (2013) 26:266–283
DOI 10.1007/s10878-012-9458-y

On some geometric problems of color-spanning sets

Wenqi Ju · Chenglin Fan · Jun Luo · Binhai Zhu · Ovidiu Daescu

Published online: 18 February 2012

Abstract In this paper we study several geometric problems of color-spanning sets: given n points with m colors in the plane, selecting m points with m distinct colors such that some geometric properties of the m selected points are minimized or maximized. The geometric properties studied in this paper are the maximum diameter, the largest closest pair, the planar smallest minimum spanning tree, the planar largest minimum spanning tree and the planar smallest perimeter convex hull. We propose an $O(n^{1+\varepsilon})$ time algorithm for the maximum diameter color-spanning set problem where ε could be an arbitrarily small positive constant. Then, we present hardness proofs for the other problems and propose two efficient constant factor approximation algorithms for the planar smallest perimeter color-spanning convex hull problem.

W. Ju
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
e-mail: wq.ju@siat.ac.cn

W. Ju
Graduate University of the Chinese Academy of Sciences, Beijing, China

W. Ju · C. Fan · J. Luo (✉)
Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China
e-mail: jun.luo@siat.ac.cn

C. Fan
e-mail: cl.fan@siat.ac.cn

B. Zhu
Department of Computer Science, Montana State University, Bozeman, MT, USA
e-mail: bhz@cs.montana.edu

O. Daescu
Department of Computer Science, University of Texas at Dallas, Dallas, TX, USA
e-mail: daescu@utdallas.edu

Keywords Computational geometry · Color-spanning set · NP complete

1 Introduction

Most of the classic algorithms in computational geometry are based on the assumption that the locations of input points are known exactly. In practice, that is not always the case. Data stored in computers are imprecise in most cases. The causes of imprecise data are various, for example, the uncertain properties of a moving object (Cheng et al. 2004), measurement error, sampling error, network latency (Sistla et al. 1997; Pfoser and Jensen 1999), location privacy protection (Beresford and Stajano 2003; Cheng et al. 2006; Gedik and Liu 2005), etc. Any or a combination of these could be the causes leading to imprecision of the data.

A continuous region model such as a disc model, a square model and so on can be used to describe an imprecise point in general. There are many studies based on such a model. Löffler et al. studied the largest and the smallest convex hull problems based on the disc model, the square model, and the parallel line segment model (Kreveld and Löffler 2007). They proved the NP-hardness for some problems and proposed polynomial time algorithms for some other problems, ranging in running time from $O(n \log n)$ to $O(n^{13})$. Other papers dealing with the problems based on the continuous region models include (Khanban and Edalat 2003; Nagai and Tokura 2000; Boissonnat and Lazard 1996) and so on.

In addition to this, a discrete point set model can be used to describe imprecise points. In the discrete point set model, a set of discrete points is used for the possible positions where an imprecise point may appear. In computational geometry, the discrete point set model is also called the color-spanning set model because the set of positions for an imprecise point is regarded as the positions with the same color. The input of color-spanning set problems are usually n points with m colors.

There has also been a lot of research studying the color-spanning set problems. Zhang et al. proposed a brute force algorithm to solve the minimum diameter color-spanning set (MDCS) problem and the running time of their algorithm is $O(n^m)$ (Zhang et al. 2009). Fleischer and Xu showed that the MDCS problem can be solved in polynomial time for the L_1 and L_∞ metrics, while it is NP-hard for L_p ($p = 2, 3, 4, \ldots$) metrics (Fleischer 2010). They also proposed an efficient constant factor approximation algorithm for the problem. Abellanas et al. showed that the Farthest Color Voronoi Diagram (FCVD) is of complexity $\Theta(nm)$ if $m \leq n/2$ (Abellanas et al. 2001a, 2001b). They proposed an efficient algorithm to construct FCVD and then they presented efficient algorithms to construct the smallest color-spanning circle, the smallest color-spanning rectangle and the narrowest color-spanning strip of arbitrary orientation with the help of FCVD. In Das et al. (2009) proposed an algorithm for identifying the smallest color-spanning corridor in $O(n^2 \log n)$ time and $O(n)$ space and an algorithm for identifying the smallest color-spanning rectangle of arbitrary orientation with an $O(n^3 \log m)$ running time and $O(n)$ space.

In the database community, a similar framework under a different name "uncertain data" has been used. An imprecise point is called an uncertain object and the different positions with the same color are regarded as the different possible instances of an uncertain object. Pei et al. have performed some research that pertains to geometric problems in this framework (Pei et al. 2007; Cheema et al. 2010; Yuen et al. 2010).

In this paper, we study five color-spanning set problems. We select m points with m colors such that

- The diameter of the m points is maximized. This problem is called the Maximum Diameter Color-spanning Set problem (MaxDCS).
- The distance between the closest pair of the m points is maximized. This problem is called the Largest Closest Pair Color-spanning Set problem (LCPCS).
- The length of the planar minimum spanning tree over the m points is minimized. This problem is called the Planar Smallest Minimum Spanning Tree Color-spanning Set problem (PSMSTCS).
- The length of the planar minimum spanning tree over the m points is maximized. This problem is called the Planar Largest Minimum Spanning Tree Color-spanning Set problem (PLMSTCS).
- The perimeter of the planar convex hull over the m points is minimized. This problem is called the Planar Smallest Perimeter Convex Hull Color-spanning Set problem (PSPCHCS).

We discuss these five problems in the following five sections and then give conclusions in the last section.

2 An efficient algorithm for MaxDCS

Computing the diameter of a point set has a long history. By a reduction to set disjointness, it can be shown that computing the diameter of n points with the same color in R^d requires $\Omega(n \log n)$ operations in the algebraic computation tree model (Preparata and Shamos 1985). However, to the best of our knowledge, the maximum diameter color-spanning set problem (see an example in Fig. 1) has not been studied yet.

Let S be a set of n points with m colors. The steps of our algorithm (MaxDCS1) to compute the maximum color-spanning diameter of S are as follows:

1. Compute the maximum distance between two points of S in the plane (ignoring colors) using the algorithm in Preparata and Shamos (1985). Let these two points be p_a and p_b. If p_a and p_b have different colors, then the distance d_0 between p_a and p_b is the maximum color-spanning diameter D of S, and we can exit the algorithm; else, we continue to the next step.

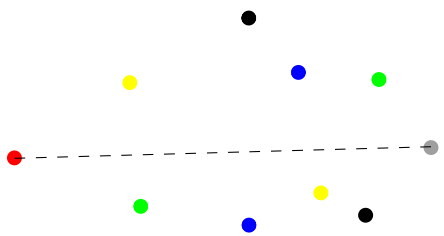


Fig. 1 Illustration of the maximum diameter of color-spanning sets

2. Let the subset of points with the same colors as that of points p_a and p_b be S_{ab} (S_{ab} also includes p_a and p_b). Let $S'_{ab} = S - S_{ab}$. We compute the distance between every pair of points, one in S_{ab} and the other in S'_{ab}, and let the resulting maximum distance be d_{ab}.
3. Let the set of points of S'_{ab} which are on or to the left of the line $p_a p_b$ be set S_l. Let the set of points of S'_{ab} which are on the right of the line $p_a p_b$ be S_r. For any point p in S_r such that no point in S_l has the same color of p, we put p into set S'_r. Symmetrically, for any point p in S_l such that no point in S_r has the same color of p, we put p into set S'_l. Let $S''_l = S_l \setminus S'_l$ and $S''_r = S_r \setminus S'_r$.
4. Let $S_1 = S_l \cup S'_r$ and compute the maximum distance d_l between two points in S_1 (ignoring colors) using the algorithm in Preparata and Shamos (1985). Similarly, let $S_2 = S_r \cup S'_l$ and compute the maximum distance d_r between two points in S_2.
5. Compute the diameter D' of point set $S''_l \cup S''_r$ (considering colors). The details of this step will be given in Algorithm MaxDCS2.
6. Let $D = \max(d_{ab}, d_l, d_r, D')$.

Next, we prove the correctness and the time complexity of our algorithm.

Lemma 1 *For two points p_e and p_f in set S_l (S_r), at least one of the four segments $\overline{p_a p_e}$, $\overline{p_a p_f}$, $\overline{p_b p_e}$, $\overline{p_b p_f}$ has a length longer than that of $\overline{p_e p_f}$.*

Proof Since p_e and p_f are on the same side of line $p_a p_b$, there are only two cases for the positions of four points p_a, p_b, p_e, p_f:

1. Four points p_a, p_b, p_e, p_f form a convex quadrangle (see Fig. 2(a)). If the length of segment $\overline{p_e p_f}$ is longer than or equal to the lengths of the segments $\overline{p_a p_e}$, $\overline{p_a p_f}$, $\overline{p_b p_e}$, $\overline{p_b p_f}$, then the angles $\angle p_e p_f p_b$ and $\angle p_f p_e p_a$ are smaller than $\pi/2$. Because the length of the segment $\overline{p_a p_b}$ is longer than or equal to all the above segments, the angles $\angle p_a p_b p_f$ and $\angle p_b p_a p_e$ are smaller than $\pi/2$. This contradicts to the fact that the sum of four angles of a convex quadrangle is 2π.
2. The three points p_a, p_b, p_e form a triangle and p_f is in $\triangle p_a p_b p_e$, $\overline{p_e p_h}$ is the height of triangle; moreover, the point p_h must be on segment $\overline{p_a p_b}$ (see Fig. 2(b)). Otherwise, the length of $\overline{p_a p_e}$ or $\overline{p_b p_e}$ is longer than $\overline{p_a p_b}$. Since p_f is either in the right triangle $\triangle p_a p_h p_e$ or in the right triangle $\triangle p_b p_e p_h$, the length of $\overline{p_e p_f}$ is shorter than the longer of $\overline{p_e p_a}$ and $\overline{p_e p_b}$.

Hence the lemma is proven. □

Let p_x and p_y be the two points corresponding to the maximum color-spanning diameter of S. There are five cases:

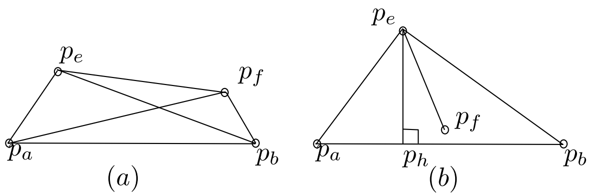


Fig. 2 The illustration for the proof of Lemma 1

1. p_x is p_a and p_y is p_b.
2. $p_x \in S_{ab}$ and $p_y \in S'_{ab}$.
3. p_x and p_y are in set S_l, or p_x and p_y are in set S_r.
4. p_x is in set S_l, while p_y is in set S'_r, or p_x in set S'_l and p_y in set S_r.
5. p_x is in set S''_l and p_y is in set S''_r.

Cases 1, 2 and 4 are computed at Steps 1, 2 and 4 of the algorithm MaxDCS1. We do not need to compute the diameter in Case 3 according to Lemma 1 and the fact that the color of p_a (p_b) is different from those points in S_l (S_r). In other words, the maximum diameter in Case 3 is less than d_{ab} in Case 2. Step 1, 3, 4, and 6 can be finished in $O(n \log n)$ time. Step 2 can be finished in $O(k'n)$ time where k' is the size of the set S_{ab}. However, if k' is larger than $O(\log n)$, in order to reduce the time complexity to $O(n \log n)$, we cannot use a brute force method at Step 2. Instead, we can compute the farthest-point Voronoi diagram of set S_{ab} in $O(k' \log k')$ time (Berg et al. 2008). The Voronoi cell where each point p in set S'_{ab} is located can be found in $O(\log k')$ time, and then we can compute the distance between p and corresponding site in an additional $O(1)$ time. Therefore, Step 2 can be finished in $O(n \log n)$ time.

How about the time complexity of Step 5 (or the algorithm MaxDCS2) (Algorithm 1)? The time consuming parts of MaxDCS2 are two FOR loops. Since those two loops are symmetric, we only analyze the first one. Notice that the difference between the consecutive steps inside the loop are point sets of two colors. In total, each point is inserted and deleted from S' once, that means only $O(n)$ insertions and deletions. It is shown that the diameter for n points without color after each insertion or deletion can be updated in $O(n^{\varepsilon})$ time where ε could be an arbitrarily small pos-

Algorithm 1 MaxDCS2

Require: Point set S''_l and S''_r with m' colors where $m' \le m - 1$;
Ensure: Maximum diameter D of color-spanning sets of $S''_l \cup S''_r$;

```
D = 0
Let S_l^k be the k-th color points in S_l'' and S_r^k be the k-th color points in S_r''
S' = S_l'' ∪ S_r^1 \ S_l^1
Compute the diameter D' of S' (ignoring colors)
D = Max(D, D')
for k = 2 to m' do
    S' = S' ∪ S_r^k ∪ S_l^{k-1} \ S_l^k \ S_r^{k-1}
    Compute the diameter D' of S' (ignoring colors)
    D = Max(D, D')
end for
S' = S_r'' ∪ S_l^1 \ S_r^1
Compute the diameter D' of S' (ignoring colors)
D = Max(D, D')
for k = 2 to m' do
    S' = S' ∪ S_l^k ∪ S_r^{k-1} \ S_r^k \ S_l^{k-1}
    Compute the diameter D' of S' (ignoring colors)
    D = Max(D, D')
end for
```

itive constant (Agarwal et al. 1992; Eppstein 1996). Therefore, the running time of MaxDCS2 is $O(n^{1+\varepsilon})$.

In Steps 4 and 5 of the algorithm MaxDCS1, the two points realizing d_l (d_r or D') could be on the same side of $p_a p_b$ and could be of the same color. However, according to Lemma 1, those d_l, d_r, D' are all shorter than d_{ab} and we let $D = max(d_{ab}, d_l, d_r, D')$. Therefore, those d_l, d_r, D' cannot be the real diameter D. Our algorithm considers the distances of all points with its farthest point of different color. Hence we have the following theorem.

Theorem 1 *Let S be a set of n points of m colors. The maximum color-spanning diameter of S can be computed in $O(n^{1+\varepsilon})$ time, where ε could be an arbitrarily small positive constant.*

3 Hardness of LCPCS

In this section, we show that LCPCS is NP-Complete even in one dimension. To facilitate the reading, we first present the proofs in the plane, under different metrics.

Theorem 2 *LCPCS is NP-hard under the L_p metric, for $2 \leq p < \infty$,*

Proof We prove the hardness of LCPCS by a reduction from 3-SAT. We give the proof for the L_2 metric in two dimensions and then show how to extend it to any L_p metric and to higher dimensions.

Let F be a Boolean formula in conjunctive normal form with n variables $x_1, \dots, x_n$ and m clauses $c_1, \dots, c_m$, each of size at most three. We take the following steps to construct an instance I of LCPCS.

For each Boolean variable $x_i, \neg x_i$ in F, let k_i be the maximum number of times that x_i and $\neg x_i$ appear in F. Then we draw a rectangle V_i vertically and separate it into $k_i - 1$ small rectangles horizontally. Every small rectangle has length b and height a. The diagonal length of each small rectangle is d ($d^2 = a^2 + b^2, d < 2a$). We place $2k_i$ points with different colors on the $2k_i$ vertices of small rectangles (see Fig. 3). The $2k_i$ points of different colors are placed on k_i rows and two columns.

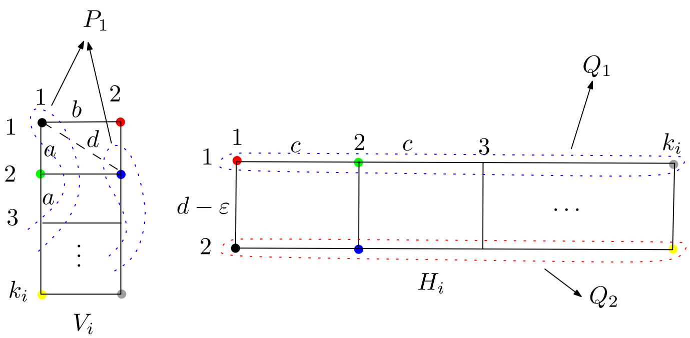


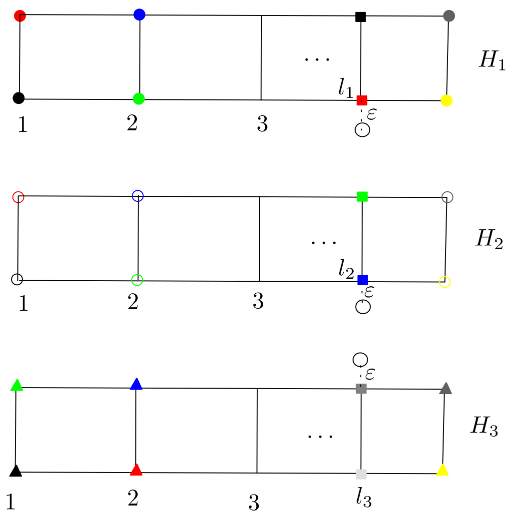
Fig. 3 Variable gadget

Let A_{xy} $(1 \le x \le k_i, 1 \le y \le 2)$ denote the point at row x and column y. Then we draw another rectangle H_i horizontally which is far away from V_i and separate it into $k_i - 1$ small rectangles vertically. Every small rectangle has length c $(c = d + \varepsilon)$ and height $d - \varepsilon$ (see Fig. 3). We place $2k_i$ points with different colors on the $2k_i$ vertices of those small rectangles. Let B_{xy} $(1 \le x \le 2, 1 \le y \le k_i)$ denote the point at row x and column y. When x is an odd number, only A_{x1} and B_{2x} have the same distinct color, and only A_{x2} and B_{1x} have the same distinct color.[1] When x is an even number, only A_{x2} and B_{2x} have the same distinct color, and only A_{x1} and B_{1x} have the same distinct color (see Fig. 3).

Let P_1 be the set of points A_{xy} where x is odd and $y = 1$ or x is even and $y = 2$. Let P_2 be the set of points A_{xy} where x is even and $y = 1$ or x is odd and $y = 2$. Let Q_1 be the set of points B_{xy} where $x = 1$ and Q_2 be the set of points B_{xy} where $x = 2$. The idea is that if we want to maximize the distance between the closest pair with this configuration, we either choose the point sets P_1 and Q_1, or point sets P_2 and Q_2. The fist case represents the value **1** for this variable, and the second case represents the value **0**. The rectangles corresponding to different variables lie far away enough to each other (at least $4d$). The points in different variables have totally different colors.

For each clause in F, we add three points and these three points have the same distinct color. We deal with the clauses from left to right. For example, let the i-th clause be $(x_1 \vee x_2 \vee \neg x_3)$, and suppose that x_1 appears $l_1 - 1$ times, x_2 appears $l_2 - 1$ times, $\neg x_3$ appears $l_3 - 1$ times in the previous $i - 1$ clauses. Then we put one point next to rectangle H_1 and it is right below the point B_{2l_1} with distance ε. The second point is next to rectangle H_2 and it is right below B_{2l_2} with distance ε. The third point is next to rectangle H_3 and it is right above B_{1l_3} with distance ε (see Fig. 4). One of these three points has to be selected for this color. For the distance between the closest

Fig. 4 Clause gadget for $(x_1 \vee x_2 \vee \neg x_3)$. Different shapes of points also mean different colors



[1] Throughout this paper, when we say that points in set S' have the same distinct color and points in set S'' have the same distinct color, it means that points in S' have a color c' and points in S'' have a color c'' with $c' \ne c''$.

pair to be maximum (i.e., equal to d), it is only possible when x_1 is assigned 1, or x_2 is assigned 1, or x_3 is assigned 0.

It is easy to see that F is satisfiable if and only if the largest distance between the closest pair is equal to d.

In order to extend the proof to the L_p metric, the only requirement is $d > a$ and $d > b$. When $1 \le p < \infty$, we have $d > a$ and $d > b$. Only when $p = \infty$, we have $d = a$ or $d = b$ (then the above construction fails). Also, the hardness for two dimensions implies the hardness for higher dimensions. Therefore LCPCS is NP-hard for the L_p metric, for $1 \le p < \infty$, in two or higher dimensions. □

We next show that LCPCS is NP-hard even in one dimension.

Theorem 3 *LCPCS is NP-Complete in one dimension.*

Proof We prove LCPCS is NP-Complete by a reduction from the 3-SAT problem. Of course, in this case all points lie on a line l. Given a 3-SAT formula F, we make the following construction.

For each Boolean variable x_i in F, let k_i be the maximum number of times that x_i and $\neg x_i$ appear in F (see Fig. 5) . We first put $2k_i$ points $A_1, \dots, A_{2k_i}$ on the segment l_{i1} which is a part of the line l and the distance between two adjacent points is $d/2$. Then we put anther $2k_i$ points $B_1, \dots, B_{2k_i}$ on the segment l_{i2} which is another part of the line l. The distance between two points B_j and B_{j+1} is $d - \varepsilon$ when j is odd, or $2d$ when j is even. Furthermore, we require that B_{j+1} and A_j have the same distinct color for $1 \le j \le 2k_i - 1$, and B_1 and A_{2k_i} have the same distinct color.

Let P_1 be the set of points A_j when j is odd, and P_2 be the set of points A_j when j is even. Let Q_1 be the set of points B_j when j is odd, and Q_2 be the set of points B_j when j is even. If we want to maximize the distance between the closest pair with this configuration, we must select either the point sets P_1 and Q_1 or the point sets P_2 and Q_2. If we select P_1 and Q_1, x_i is assigned 0 and if we select P_2 and Q_2, x_i is assigned 1. The distance between every two variable gadgets is great enough (not less than $4d$). The colors of points in different variable gadgets are totally different.

For every clause in F, we need three additional points P_a, P_b and P_c. Certainly, only P_a, P_b and P_c for the same clause have the same distinct color. We deal with the clauses from left to right. For example, let the i-th clause be $(x_u \vee x_v \vee \neg x_w)$, and suppose that x_u appears $h_1 - 1$ times, x_v appears $h_2 - 1$ times, $\neg x_w$ appears $h_3 - 1$ times in the previous $i - 1$ clauses. Then we put P_a on the segment l_{u2} and it is to the left of the point B_{2h_1-1} with distance ε, P_b on l_{v2} and it is to the left of the pointB_{2h_2-1} with distance ε, and P_c on the segment l_{w2} and it is to the right of the

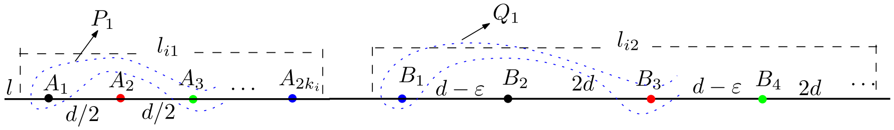


Fig. 5 Variable gadget for variable x_i for one dimension

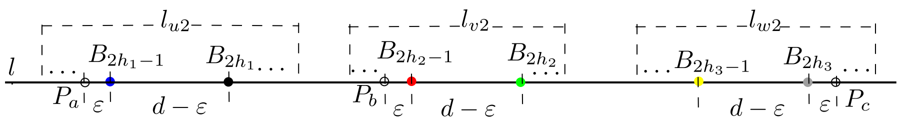


Fig. 6 Clause gadget for $(x_u \vee x_v \vee \neg x_w)$ for one dimension

point B_{2h_3} (see Fig. 6). The distance between the closest pair become maximum (i.e., equal to d) if and only if at least one of x_u or x_v or $\neg x_w$ is assigned 1.

Therefore, F is satisfiable if and only if the largest possible distance between the closest pair is equal to d. □

Actually we can prove that LCPCS is $(\frac{1}{2}+\varepsilon)$-APX-hard, which means that it is NP-hard to find any approximation algorithm with an approximation ratio better than $\frac{1}{2}$.

Theorem 4 *LCPCS is* $(\frac{1}{2}+\delta)$*-APX-hard* $(0<\delta\le\frac{1}{2})$ *in one dimension.*

Proof Consider Theorem 3 and Figs. 5 and 6, and let $\varepsilon = d/2$. Let O^* be the optimal solution value of LCPCS. We need to consider two cases:

(1) $O^* = d$ if and only if the 3-SAT formula F is satisfiable.
(2) $O^* \le d/2$ if and only if the 3-SAT formula F is not satisfiable.

Hence, to find an approximation algorithm whose approximation ratio is better than $\frac{1}{2}$ is equivalent to finding an algorithm for $O^* = d$, which is NP-hard. The theorem is proved. □

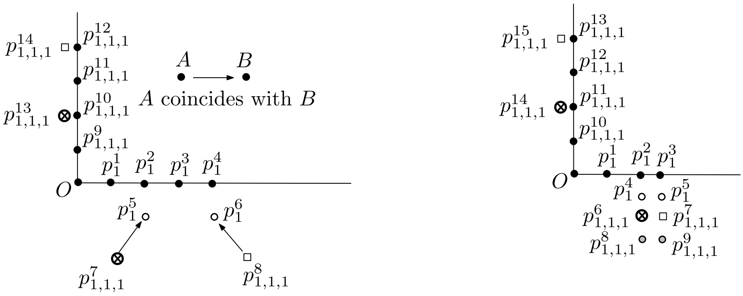
4 Hardness of PSMSTCS

In this section, we prove the NP-completeness of the Planar Smallest Minimum Spanning Tree Color-Spanning Set (PSMSTCS) problem. First we show that this problem belongs to NP. Given an instance of the problem, we use as a certificate the m different color points chosen from n points. The verification algorithm computes the MST of those m points and check whether the length is at most L. This process can certainly be done in polynomial time.

We then prove that PSMSTCS is NP-hard by a reduction from the 3-SAT problem. The general idea is that for a 3-SAT formula, we put some colored points on the plane such that the given 3-SAT formula is satisfiable if and only if the length of the smallest color-spanning minimum spanning tree equals some given value.

First we put the point O with a distinct color at $(0,0)$. Given a 3-SAT formula ψ, suppose that it has n_1 variables $x_1, x_2, \dots, x_{n_1}$ and m_1 clauses. For each variable x_i $(1 \le i \le n_1)$, six points p_i^1, p_i^2, p_i^3, p_i^4, p_i^5 and p_i^6 are put at $(400i-300,0)$, $(400i-200,0)$, $(400i-100,0)$, $(400i,0)$, $(400i-200,-100)$ and $(400i,-100)$ respectively.

If the j-th literal, which is in the k-th clause in ψ, is x_i $(\neg x_i)$, we denote the literal by $x_{i,j,k}$ $(\neg x_{i,j,k})$. For every literal $x_{i,j,k}$ $(\neg x_{i,j,k})$, where $1 \le i \le n_1$,



(a) (b)

Fig. 7 (**a**) The gadget for the variable x_1, which is also the first literal and in the first clause, in PSMSTCS. (**b**) The gadget for the variable x_1, which is also the first literal and in the first clause, in PLMSTCS. Different symbols means different colors. Every *solid circle* has a distinct color

$j = 1, 2, 3, \dots$ and $1 \le k \le m_1$, eight additional points are constructed. $p^7_{i,j,k}$ is put at $(400i - 200, -100)$, $p^8_{i,j,k}$ at $(400i, -100)$, $p^9_{i,j,k}$ at $(0, 400j - 300)$, $p^{10}_{i,j,k}$ at $(0, 400j - 200)$, $p^{11}_{i,j,k}$ at $(0, 400j - 100)$, $p^{12}_{i,j,k}$ at $(0, 400j)$, $p^{13}_{i,j,k}$ at $(-\frac{1}{3m_1}, 400j - 200)$ and $p^{14}_{i,j,k}$ at $(-\frac{1}{3m_1}, 400j)$. Among those fourteen points, only p^5_i and p^6_i have the same distinct color, only $p^7_{i,j,k}$ and $p^{13}_{i,j,k}$ have the same distinct color, only $p^8_{i,j,k}$ and $p^{14}_{i,j,k}$ have the same distinct color, and every one of the other points has a distinct color. Figure 7(a) shows the gadget for the variable x_1 which is also the first literal and in the first clause.

Then, we have a set P of $6n_1 + 24m_1 + 1$ points for the formula ψ. $4n_1$ points (except O) are put on the x-axis, $2n_1 + 2 \times 3m_1 = 2n_1 + 6m_1$ points are put on the line $y = -100$, $4 \times 3m_1 = 12m_1$ (except O) points are put on the y-axis, $2 \times 3m_1 = 6m_1$ points are put on the line $x = -\frac{1}{3m_1}$, one point is on the point O. Let T_P be the planar smallest color-spanning minimum spanning tree over P.

Since only p^5_i and p^6_i have the same distinct color, only $p^7_{i,j,k}$ and $p^{13}_{i,j,k}$ have the same distinct color and only $p^8_{i,j,k}$ and $p^{14}_{i,j,k}$ have the same distinct color, we have to select either $\{p^5_i, p^7_{i,j,k}, p^{14}_{i,j,k}\}$ or $\{p^6_i, p^8_{i,j,k}, p^{13}_{i,j,k}\}$ to get the planar smallest color-spanning minimum spanning tree T_P. Whichever set of points is selected, the length of T_P is $500n_1 + 400 \times 3m_1 + 3m_1 \times \frac{1}{3m_1} = 500n_1 + 1200m_1 + 1$. If $\{p^5_i, p^7_{i,j,k}, p^{14}_{i,j,k}\}$ are selected, x_i is assigned 0. If $\{p^6_i, p^8_{i,j,k}, p^{13}_{i,j,k}\}$ are selected, x_i is assigned 1 (see Fig. 8).

Now, we need three additional points for every clause in the 3-SAT formula, where only the three points in the same clause have the same distinct color. If there is a literal $x_{i1,j1,k1}$ which is the $j1$-th literal and in the $k1$-th clause, we put a point $p^{or}_{i1,\lceil\frac{j1}{3}\rceil}$ at the position of $p^{13}_{i1,j1,k1}$. If there is a literal $\neg x_{i1,j1,k1}$, we put a point $p^{or}_{i1,\lceil\frac{j1}{3}\rceil}$ at the position of $p^{14}_{i1,j1,k1}$. Note that the points representing the literals in the same clause should have the same distinct color. For example, we assume that one clause is $\neg x_{i1,j1,k1} \vee x_{i2,j1+1,k2} \vee x_{i3,j1+2,k3}$. We put three points $p^{or}_{i1,\lceil\frac{j1}{3}\rceil}$, $p^{or}_{i2,\lceil\frac{j1+1}{3}\rceil}$

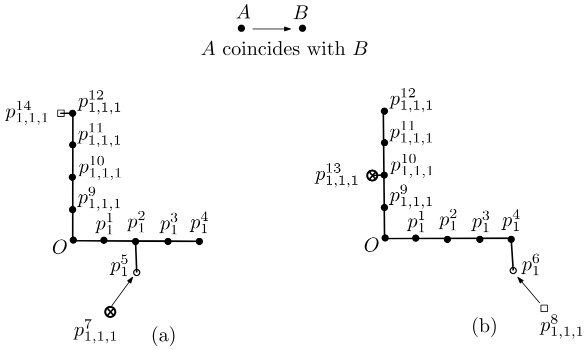


Fig. 8 Suppose x_1 is the first literal in the first clause of 3-SAT formula. (**a**) The planar smallest color-spanning minimum spanning tree when x_1 is assigned 0. (**b**) The planar smallest color-spanning minimum spanning tree when x_1 is assigned 1

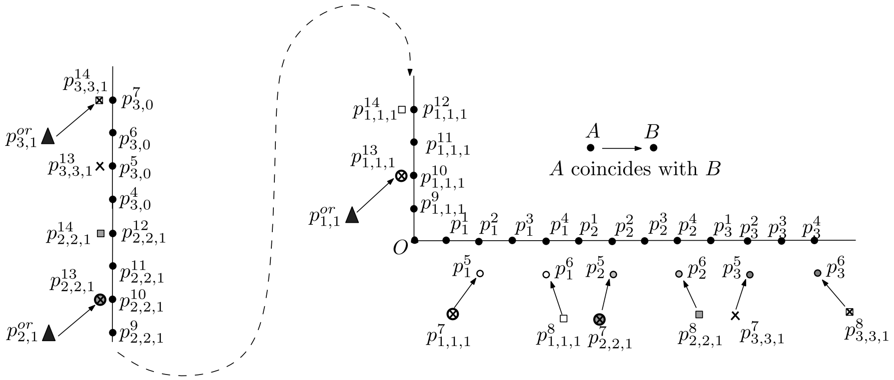


Fig. 9 Gadget for the clause $(x_1 \vee x_2 \vee \neg x_3)$, assuming it is the first clause

and $p^{or}_{i3,\lceil\frac{j1+2}{3}\rceil}$ at $p^{14}_{i1,j1,k1}$, $p^{13}_{i2,j1+1,k2}$ and $p^{13}_{i3,j1+2,k3}$ respectively. Note that only $p^{or}_{i1,\lceil\frac{j1}{3}\rceil}$, $p^{or}_{i2,\lceil\frac{j1+1}{3}\rceil}$ and $p^{or}_{i3,\lceil\frac{j1+2}{3}\rceil}$ have the same distinct color.

Figure 9 shows the variable gadget for the first clause $(x_1 \vee x_2 \vee \neg x_3)$ and Fig. 10 shows the gadgets for $\psi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_3 \vee x_4)$. Let T_{opt} be the planar smallest color-spanning minimum spanning tree over P plus the $3m_1$ points for every clause. For literal $x_{i,j,k}$ $(\neg x_{i,j,k})$, if a variable x_i is assigned 1 (0), $\{p^6_i, p^8_{i,j,k}, p^{13}_{i,j,k}\}$ $(\{p^5_i, p^7_{i,j,k}, p^{14}_{i,j,k}\})$ are selected. That means the point $p^{or}_{i,\lceil\frac{j}{3}\rceil}$ can be selected to construct T_{opt} without adding any extra length comparing with T_P. Therefore, if at least one literal is assigned 1 in every clause, the length of T_{opt} equals that of T_P. If all three literals in at least one clause are assigned 0, all the three points for the clause cannot be covered by points in T_P and the length of T_{opt} is greater than that of T_P. Therefore, we obtain that a 3-SAT formula ψ with n_1 variables $x_1, x_2, \ldots, x_{n_1}$ and m_1 clauses is satisfiable if and only if the length of the corresponding T_{opt} is equal to $500n_1 + 1200m_1 + 1$. Because the 3-SAT problem is NP-Complete, we obtain the following theorem:

Theorem 5 *PSMSTCS is NP-Complete.*

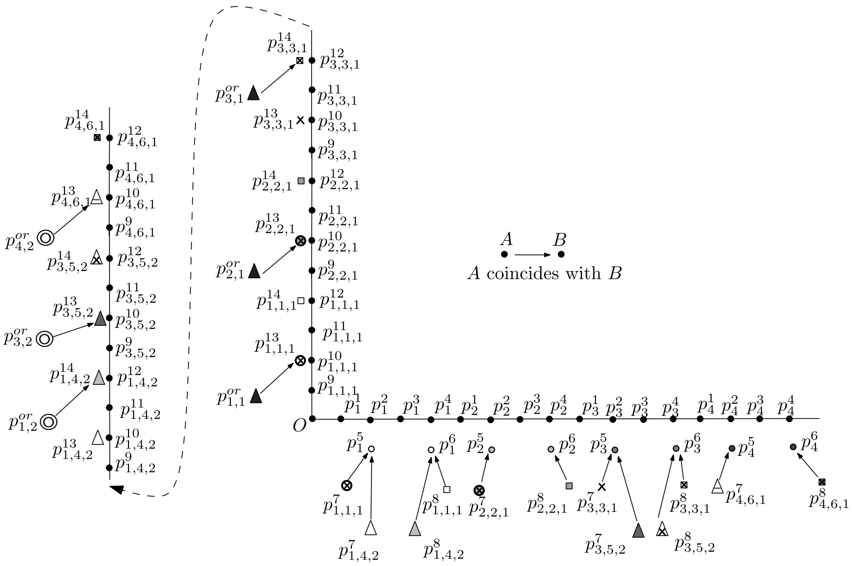


Fig. 10 Gadgets for $\psi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_3 \vee x_4)$

5 Hardness of PLMSTCS

In this section, we prove the NP-completeness of the Planar Largest Minimum Spanning Tree Color-Spanning Set (PSMSTCS) problem. Again, we first show that this problem belongs to NP. Given an instance of the problem, we use as a certificate the m different color points chosen from n points. The verification algorithm computes the MST of those m points and check whether the length is at most L. This process can certainly be done in polynomial time.

We prove this problem is NP-hard by a reduction from the 3-SAT problem. First we put a point O with a distinct color at $(0,0)$. For a 3-SAT formula ψ, suppose that it has n_1 variables $x_1, x_2, \ldots, x_{n_1}$ and m_1 clauses. For each variable x_i $(1 \le i \le n_1)$, five points p_i^1, p_i^2, p_i^3, p_i^4 and p_i^5 are put at $(201i-101, 0)$, $(201i-1, 0)$, $(201i, 0)$, $(201i-1, -10)$, $(201i, -10)$ respectively.

If the j-th literal, which is in the k-th clause in ψ, is x_i $(\neg x_i)$, we also denote the literal by $x_{i,j,k}$ $(\neg x_{i,j,k})$. For every literal $x_{i,j,k}$ (or $\neg x_{i,j,k}$), ten additional points are put down. $p^6_{i,j,k}$ is put at $(201i-1, -9-2k)$, $p^7_{i,j,k}$ at $(201i, -9-2k)$, $p^8_{i,j,k}$ at $(201i-1, -10-2k)$, $p^9_{i,j,k}$ at $(201i, -10-2k)$, $p^{10}_{i,j,k}$ at $(0, 400j-300)$, $p^{11}_{i,j,k}$ at $(0, 400j-200)$, $p^{12}_{i,j,k}$ at $(0, 400j-100)$, $p^{13}_{i,j,k}$ at $(0, 400j)$, $p^{14}_{i,j,k}$ at $(-0.5, 400j-200)$ and $p^{15}_{i,j,k}$ at $(-0.5, 400j)$. Among those fifteen points, only p_i^4 and p_i^5 have the same distinct color, only $p^6_{i,j,k}$ and $p^{14}_{i,j,k}$ have the same distinct color, only $p^7_{i,j,k}$ and $p^{15}_{i,j,k}$ have the same distinct color, only $p^8_{i,j,k}$ and $p^9_{i,j,k}$ have the same distinct color, and every one of the other points has a distinct color. Figure 7(b) shows the gadget for the variable x_1 which is the first literal and in the first clause.

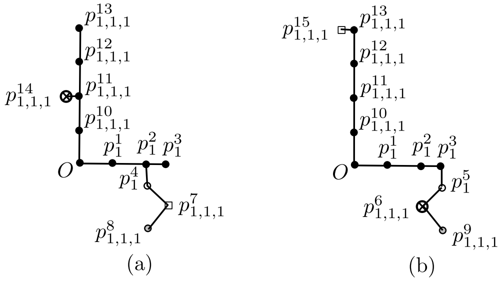


Fig. 11 (**a**) The planar largest color-spanning minimum spanning tree when x_1 is assigned 0. (**b**) The planar largest color-spanning minimum spanning tree when x_1 is assigned 1

Then, we obtain a set P of $5n_1 + 30m_1 + 1$ points in the plane. Let T_P be the planar largest color-spanning minimum spanning tree over P. $3n_1 + 12m_1 + 1$ points are on the x-axis and the y-axis, and every one of them has a distinct color.

Since only p_i^4 and p_i^5 have the same distinct color, only $p^6_{i,j,k}$ and $p^{14}_{i,j,k}$ have the same distinct color, only $p^7_{i,j,k}$ and $p^{15}_{i,j,k}$ have the same distinct color, only $p^8_{i,j,k}$ and $p^9_{i,j,k}$ have the same distinct color, we have to select either $\{p_i^4, p^7_{i,j,k}, p^8_{i,j,k}, p^{14}_{i,j,k}\}$ or $\{p_i^5, p^6_{i,j,k}, p^9_{i,j,k}, p^{15}_{i,j,k}\}$ to obtain the planar largest color-spanning minimum spanning tree T_P whose length is $(201 + 10)n_1 + 3m_1(400 + 0.5 + 2\sqrt{2}) = 211n_1 + 1201.5m_1 + 6\sqrt{2}m_1$. If $\{p_i^4, p^7_{i,j,k}, p^8_{i,j,k}, p^{14}_{i,j,k}\}$ are selected, x_i is assigned 0. If $\{p_i^5, p^6_{i,j,k}, p^9_{i,j,k}, p^{15}_{i,j,k}\}$ are selected, x_i is assigned 1 (see Fig. 11).

Now, we need three additional points for every clause in the 3-SAT formula, where only the three points in the same clause have the same distinct color. For every literal $x_{i,j,k}$ (or $\neg x_{i,j,k}$), we put one point $p^{or}_{i,\lceil \frac{j}{3} \rceil}$ at the position of $p^{14}_{i,j,k}$ (or $p^{15}_{i,j,k}$). Figure 12 shows the gadgets for $\psi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_3 \vee x_4)$.

Let T_{opt} be the planar largest color-spanning minimum spanning tree over P plus the $3m_1$ points for every clause. For literal $x_{i,j,k}$ ($\neg x_{i,j,k}$), if a variable x_i is assigned 1 (0), $\{p_i^5, p^6_{i,j,k}, p^9_{i,j,k}, p^{15}_{i,j,k}\}$ ($\{p_i^4, p^7_{i,j,k}, p^8_{i,j,k}, p^{14}_{i,j,k}\}$) are selected. This means that the point $p^{or}_{i,\lceil \frac{j}{3} \rceil}$ can be selected to construct T_{opt}. Therefore, if at least one literal is assigned 1 in every clause, the length of T_{opt} equals that of T_P plus $m_1 * 0.5$. If all three literals in at least one clause are assigned 0, the length of T_{opt} is less than that of T_P plus $3m_1 * 0.5$. Consequently, we obtain that a 3-SAT formula ψ with n_1 variables $x_1, x_2, \ldots, x_{n_1}$ and m_1 clauses is satisfiable if and only if the length of the corresponding T_{opt} is equal to $211n_1 + 1202m_1 + 6\sqrt{2}m_1$. Because the 3-SAT problem is NP-Complete, we obtain the following theorem:

Theorem 6 *PLMSTCS is NP-Complete.*

6 Hardness of PSPCHCS

Finally in Sect. 6, we prove the NP-completeness of the Planar Smallest Perimeter Convex Hull Color-Spanning Set (PSPCHCS) problem, followed with two simple approximation algorithms.

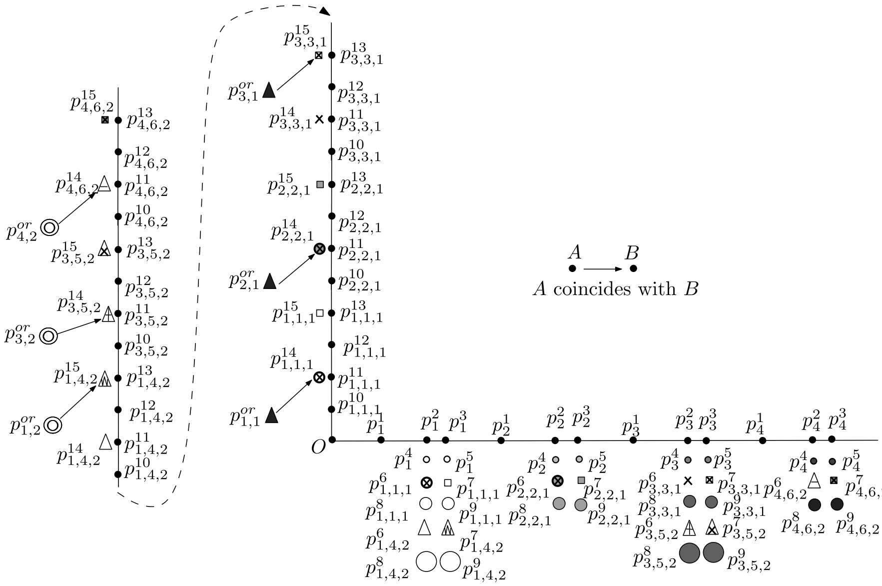


Fig. 12 Gadgets for $\psi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_3 \vee x_4)$

6.1 NP-completeness for PSPCHCS

First we show that this problem belongs to NP. Given an instance of the problem, we use as a certificate the m different color points chosen from n points. The verification algorithm computes the perimeter of the convex hull of those m points and check whether the perimeter is at most p. This process can certainly be done in polynomial time.

Again, we prove this problem is NP-hard by a reduction from the 3-SAT problem. For a given 3-SAT formula, suppose that it has n_1 variables $x_1, x_2, \ldots, x_{n_1}$. First we draw a circle C. For two points a and b on C, let $\widetilde{ab}$ be the arc of C from a to b and $\overline{ab}$ be the line segment between a and b. Let the length of $\overline{ab}$ be $|\overline{ab}|$.

For each variable x_j $(j = 1, 2, \ldots, n_1)$, we put 10 points $x_j^1, x_j^2, \ldots, x_j^{10}$ on the circle C in clockwise order (see Fig. 13). Note that in Fig. 13 these points are shown just for the clarity purpose, we can in fact screeze these points so that all the $10n_1$ points can be put on the circle C. Only x_j^2 and x_j^3 have the same distinct color and only p has a distinct color for every one point p of the other points. $|\overline{x_j^1 x_j^2}| = |\overline{x_j^3 x_j^4}|$. Then we add two additional points x_j^{11} and x_j^{12}, where x_j^{11} is on the line segment $\overline{x_j^1 x_j^2}$ and x_j^{12} is on the line segment $\overline{x_j^3 x_j^4}$. Only x_j^{11} and x_j^6 have the same distinct color, and only x_j^{12} and x_j^9 have the same distinct color. Moreover, we have the following equations:

$$|\overline{x_j^1 x_j^2}| + |\overline{x_j^2 x_j^4}| = |\overline{x_j^1 x_j^3}| + |\overline{x_j^3 x_j^4}| = Z_1;$$

$$|\overline{x_j^5 x_j^7}| = |\overline{x_j^8 x_j^{10}}| = Z_2;$$

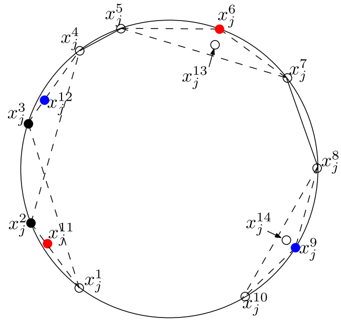


Fig. 13 The gadget for x_j. The colors of all the empty circles are different from each other and from *solid colored points*. *Solid line segments* must appear on CH_{opt}. *Dashed line segments* are candidates for edges on CH_{opt}

$$|\overline{x_j^5x_j^6}| = |\overline{x_j^6x_j^7}| = |\overline{x_j^8x_j^9}| = |\overline{x_j^9x_j^{10}}| = Z_3;$$

$$|\overline{x_j^1x_j^{11}}| + |\overline{x_j^{11}x_j^3}| - |\overline{x_j^1x_j^3}| = |\overline{x_j^2x_j^{12}}| + |\overline{x_j^{12}x_j^4}| - |\overline{x_j^2x_j^4}| = \Delta Z_1;$$

$$2Z_3 - Z_2 = \Delta Z_2; \Delta Z_1 \gg \Delta Z_2.$$

$\Delta p_1 \gg \Delta p_2$ ensures that either $\overline{x_j^1x_j^2} \cup \overline{x_j^2x_j^4}$ or $\overline{x_j^1x_j^3} \cup \overline{x_j^3x_j^4}$ is the part of CH_{opt}. Suppose we select x_j^1, x_j^2 and x_j^4 to be the vertices of CH_{opt}. Then we must select x_j^{11}, x_j^8, x_j^9, x_j^{10}, x_j^5 and x_j^7 in order to obtain the minimum length.

If we select $\overline{x_j^1x_j^2} \cup \overline{x_j^2x_j^4} \cup \overline{x_j^4x_j^5} \cup \overline{x_j^5x_j^7} \cup \overline{x_j^7x_j^8} \cup \overline{x_j^8x_j^9} \cup \overline{x_j^9x_j^{10}}$ as a part of CH_{opt}, the Boolean variable x_j is assigned 1. If we select $\overline{x_j^1x_j^3} \cup \overline{x_j^3x_j^4} \cup \overline{x_j^4x_j^5} \cup \overline{x_j^5x_j^6} \cup \overline{x_j^6x_j^7} \cup \overline{x_j^7x_j^8} \cup \overline{x_j^8x_j^{10}}$ as a part of CH_{opt}, x_j is assigned 0.

Therefore, if we connect all the n_1 gadgets for the n_1 variables together to construct a convex hull CH, the perimeter of CH is $n_1 \times (Z_1 + 2Z_2 + \Delta Z_2) + Z_4$ where $Z_4 = 2\sum_{j=1}^{n_1}(|\overline{x_j^4x_j^5}| + |\overline{x_j^7x_j^8}|) + \sum_{j=1}^{n_1-1}|\overline{x_j^{10}x_{j+1}^1}| + |\overline{x_{n_1}^{10}x_1^1}|$.

We need three additional points for every clause and only the three points for the same clause have the same distinct color. For every positive literal x_j in one clause, we put a new point at the position of x_j^{13} which is inside the triangle $\triangle x_j^5x_j^6x_j^7$ such that $\Delta Z_3 = |\overline{x_j^5x_j^{13}}| + |\overline{x_j^{13}x_j^7}| - |\overline{x_j^5x_j^7}| \ll \Delta Z_2$. For every negative literal $\neg x_j$ in one clause, we put a new point at the position of x_j^{14} which is inside the triangle $\triangle x_j^8x_j^9x_j^{10}$ such that $\Delta Z_3 = |\overline{x_j^8x_j^{14}}| + |\overline{x_j^{14}x_j^{10}}| - |\overline{x_j^8x_j^{10}}| \ll \Delta Z_2$. For example, if there is a clause $(x_1 \vee x_2 \vee \neg x_3)$, we put three points at the position of x_1^{13}, x_2^{13} and x_3^{14} respectively. x_1^{13}, x_2^{13} and x_3^{14} lie in the triangle $\triangle x_1^5x_1^6x_1^7$, $\triangle x_2^5x_2^6x_2^7$ and $\triangle x_3^8x_3^9x_3^{10}$ respectively.

We call x_j^6 and x_j^9 the *apex-points*, and call x_j^{13} and x_j^{14} the *or-points*. If all the three literals in one clause are assigned 0, then the corresponding *apex-points* will not be selected as the vertices of CH_{opt}. One of the three *or-points* has to be selected as the vertex of CH_{opt} because only they have the same distinct color. Then the perimeter of CH_{opt} at least equals that of CH plus ΔZ_3. If a given 3-SAT formula with n_1 variables is satisfiable, the perimeter of CH_{opt} equals $n_1 \times (Z_1 + 2Z_2 + \Delta Z_2) + Z_4$. Similarly, at least one literal of every clause is assigned 1 and the given 3-SAT formula is satisfiable if the perimeter of CH_{opt} over the gadgets equals $n_1 \times (Z_1 +$

$2Z_2 + \Delta Z_2) + Z_4$. Otherwise, the perimeter of the convex hull $CH_{opt} \geq n_1 \times (Z_1 + 2Z_2 + \Delta Z_2) + Z_4 + \Delta Z_3$. Because the 3-SAT problem is NP-Complete, we obtain the following theorem:

Theorem 7 *PSPCHCS is NP-Complete.*

6.2 Approximation algorithms for PSPCHCS

Here we present two simple approximation algorithms for the PSPCHCS problem.

6.2.1 The π-approximation algorithm

For every point p in the plane, select $m-1$ points $p_1, \ldots, p_{m-1}$. The colors of every two points of the m selected points are different. Moreover, p_i $(1 \leq i \leq m-1)$ is the closest point from p among all the points which have the same color with p_i. Use the m points to construct a convex hull CH_p and compute the perimeter of CH_p. Thus, we can obtain n perimeters and the smallest one is what we would return.

The running time for constructing a convex hull is $O(m \log m)$ according to Graham (1972) and the running time for finding out the other $m-1$ points is $O(n)$ when p is selected. Therefore, the total running time of this algorithm is $O(n(n+m\log m))$.

Assume that the convex hull we obtain is CH_{app}, the length of CH_{app} is per_{app}, the optimal convex hull is CH_{opt} and perimeter of CH_{opt} is per_{opt}. Now we prove that $per_{app} \leq \pi * per_{opt}$.

Suppose p_a and p_b are the vertices of CH_{opt} and $\overline{p_a p_b}$ is the diameter of CH_{opt} (see Fig. 14). Let $r = |\overline{p_a p_b}|$. We draw a circle C with center p_a and radius r. When we select p_a as the above p and select the $m-1$ points $p_1, \ldots, p_{m-1}$ to construct CH_{p_a}, CH_{p_a} must be inside C because p_i $(1 \leq i \leq m-1)$ is the closest point from p among all the points which have the same color with p_i and CH_{opt} is inside C. Then the perimeter of CH_{p_a}, denoted by per_{p_a}, satisfies $per_{p_a} \leq 2\pi r$. Because $per_{app} \leq per_{p_a}$ and $per_{opt} \geq 2r$, $per_{app} \leq \pi * per_{opt}$.

Theorem 8 *There is a π-approximation algorithm for PSPCHCS with a running time $O(n^2 + nm \log m)$.*

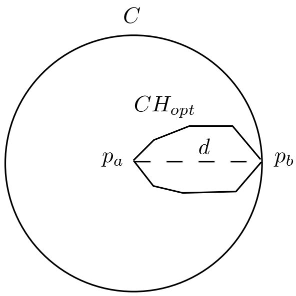


Fig. 14 Illustration of the π-approximation algorithm

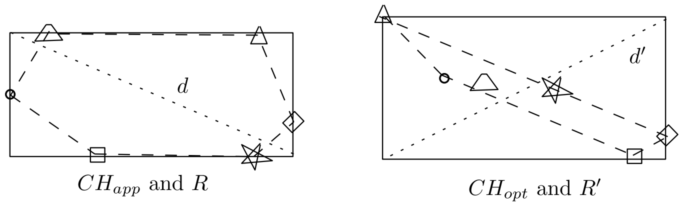


Fig. 15 Illustration of the $\sqrt{2}$-approximation algorithm

6.2.2 The $\sqrt{2}$-approximation algorithm

In Abellanas et al. (2001b) proposed an $O(\min\{n(n-m)^2, nm(n-m)\})$ time algorithm for computing the smallest perimeter axis-parallel rectangle enclosing at least one point of each color. We use the above algorithm to obtain the smallest perimeter axis-parallel rectangle R. After R is obtained, we construct a convex hull CH_{app} over the points inside R. CH_{app} is the convex hull we need.

Let the perimeter of R be per_R and let the diagonal of R be l_{dia} and the length of l_{dia} be d. CH_{app} contains at least one point of each color and CH_{app} is totally inside R (see Fig. 15). Therefore $per_{app} \leq per_R$. Let R' be the smallest perimeter axis-parallel rectangle enclosing CH_{opt}, $per_{R'}$ be the perimeter of R', l'_{dia} be the diagonal of R' and d' be the length of l'_{dia}. Because R' is the smallest perimeter axis-parallel rectangle, at least one point with a distinct color lies on every edge of R'. Then $per_{R'} \geq per_{opt} \geq 2d'$. We know $per_{R'} \leq 2\sqrt{2}d'$. Therefore, $per_{R'} \geq per_{opt} \geq \frac{\sqrt{2}}{2} per_{R'}$. Hence we have $per_{opt} \leq per_{app} \leq \sqrt{2} per_{opt}$, due to that $per_{R'} \geq per_R \geq per_{app}$.

Theorem 9 *There is a $\sqrt{2}$-approximation algorithm for PSPCHCS with a running time $O(\min\{n(n-m)^2, nm(n-m)\})$.*

7 Conclusions

In this paper we study several geometric problems of color-spanning sets. We propose an $O(n^{1+\varepsilon})$ time algorithm for MaxDCS and show that LCPCS is NP-Complete for the L_p $(1 \leq p < \infty)$ metric. Moreover, we prove that LCPCS is $(\frac{1}{2}+\varepsilon)$-APX-hard in one dimension, which means that finding an approximation algorithm whose approximation ratio is better than $\frac{1}{2}$ is NP-hard. Then we prove that PSMSTCS, PLMSTCS and PSPCHCS are NP-Complete and propose two efficient constant factor approximation algorithms for PSPCHCS. For the future work, it will be interesting to investigate whether there exists an $\frac{1}{2}$-approximation for LCPCS in one dimension.

References

Abellanas M, Hurtado F, Icking C, Klein R, Langetepe E, Ma L, Palop B, Sacristan V (2001a) The farthest color Voronoi diagram and related problems. In: Proceedings of the 17th European workshop on computational geometry (EWCG'01), pp 113–116

Abellanas M, Hurtado F, Icking C, Klein R, Langetepe E, Ma L, Palop B, Sacristan V (2001b) Smallest color-spanning objects. In: Proc 9th annu European sympos algorithms, pp 278–289
Agarwal PK, Eppstein D, Matousek J (1992) Dynamic half-space reporting, geometric optimization, and minimum spanning trees. In: Proceedings of the 33rd annual symposium on foundations of computer science, pp 80–89. ISBN:0-8186-2900-2
Beresford AR, Stajano F (2003) Location privacy in pervasive computing. IEEE Pervasive Comput 2(1):46–55
Berg M, Cheong O, Kreveld M, Overmars M (2008) Computational geometry: algorithms and applications, 3rd edn. Springer, Berlin
Boissonnat JD, Lazard S (1996) Convex hulls of bounded curvature. In: Proc 8th Canad conf on comput geom, pp 14–19
Cheema MA, Lin X, Wang W, Zhang W, Pei J (2010) Probabilistic reverse nearest neighbor queries on uncertain data. IEEE Trans Knowl Data Eng 22(4):550–564
Cheng R, Kalashnikov DV, Prabhakar S (2004) Querying imprecise data in moving object environments, knowledge and data engineering. IEEE Trans Knowl Data Eng 16(9):1112–1127
Cheng R, Zhang Y, Bertino E, Prabhakar S (2006) Preserving user location privacy in mobile data management infrastructures. In: Proceedings of the 6th intl workshop on privacy enhancing technologies (PET'06). LNCS, vol 4258. Springer, Berlin, pp 393–412
Das S, Goswani PP, Nandy SC (2009) Smallest color-spanning object revised. Int J Comput Geom Appl 19(5):457–478
Eppstein D (1996) Average case analysis of dynamic geometric optimization. Comput Geom Theory Appl 6(1):45–68
Fleischer R, Xu X (2010) Computing minimum diameter color-spanning sets. In: Proceedings of the 4th international workshop on frontiers in algorithmics (FAW'10). LNCS, vol 6213. Springer, Berlin, pp 285–292
Gedik B, Liu L (2005) A customizable k-anonymity model for protecting location privacy. In: Proceedings of the 25th international conference on distributed computing systems (ICDCS'05), pp 620–629
Graham RL (1972) An efficient algorithm for determining the convex hull of a finite planar set. Inf Process Lett 26:132–133
Khanban AA, Edalat A (2003) Computing Delaunay triangulation with imprecise input data. In: Proc 15th Canad conf on comput geom, pp 94–97
Kreveld MV, Löffler M (2007) Largest bounding box, smallest diameter, and related problems on imprecise points. In: Proc 10th workshop on algorithms and data structures. LNCS, vol 4619. Springer, Berlin, pp 447–458
Nagai T, Tokura N (2000) Tight error bounds of geometric problems on convex objects with imprecise coordinates. In: Jap conf on discrete and comput geom. LNCS, vol 2098. Springer, Berlin, pp 252–263
Pei J, Jiang B, Lin X, Yuan Y (2007) Probabilistic skylines on uncertain data. In: VLDB'2007, pp 15–26
Pfoser D, Jensen C (1999) Capturing the uncertainty of moving-objects representations. In: Proceedings of the 6th intl symp on advances in spatial databases (SSD'99). LNCS, vol 1651. Springer, Berlin, pp 111–131
Preparata FP, Shamos MI (1985) Computational geometry: an introduction. Springer, New York
Sistla PA, Wolfson O, Chamberlain S, Dao S (1997) Querying the uncertain position of moving objects. In: Temporal databases: research and practice. LNCS, vol 1399. Springer, Berlin, pp 310–337
Yuen SM, Tao Y, Xiao X, Pei J, Zhang D (2010) Superseding nearest neighbor search on uncertain spatial databases. IEEE Trans Knowl Data Eng 22(7):1041–1055
Zhang D, Chee YM, Mondal A, Tung AKH, Kitsuregawa M (2009) Keyword search in spatial databases: towards searching by document. In: Proceedings of the 25th IEEE international conference on data engineering (ICDE'09), pp 688–699