NP-Completeness of Spreading Colored Points

Ovidiu Daescu[1], Wenqi Ju[2,3], and Jun Luo[3,⋆]

[1] Department of Computer Science, University of Texas at Dallas, USA
[2] Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China
[3] Institute of Computing Technology, Chinese Academy of Sciences, China
daescu@utdallas.edu, {wq.ju,jun.luo}@sub.siat.ac.cn

Abstract. There are n points in the plane and each point is painted by one of m colors where $m \leq n$. We want to select m different color points such that (1) the total edge length of resulting minimal spanning tree is as small as possible; or (2) the total edge length of resulting minimal spanning tree is as large as possible; or (3) the perimeter of the convex hull of m different color points is as small as possible. We prove NP-completeness for those three problems and give approximations algorithms for the third problem.

1 Introduction

Data from the real world are often imprecise. Measurement error, sampling error, network latency [5, 6], location privacy protection [1, 2] may lead to imprecise data. Imprecise data can be modeled by a continuous range such as square, line segment and circle [7, 8]. Imprecise data can also be modeled by discrete range such as point set. In discrete model, each point set is assigned one distinct color. Then the problem is converted to choosing one point from each colored point set such that the resulting geometric structure is optimal. The discrete model has applications in many areas such as Voronoi diagram [3], community system [9] and color-spanning object [3].

Related Work. Löffler and van Kreveld [7] address the problem of finding the convex hull of maximum/minimum area or maximum/minimum perimeter based on line segment or squares. The running time varies from $O(n \log n)$ to $O(n^{13})$.

Ju and Luo [8] propose an $O(n \log n)$ algorithm for the convex hull of the maximum area based on the imprecise data modeled by equal sized parallel line segment and an $O(n^4)$ algorithm for the convex hull of the maximum perimeter based on the imprecise data modeled by non-equal sized parallel line segment.

For discrete imprecise data model, Zhang et al. [13] use a brute force algorithm to solve the problem of the minimum diameter color-spanning set problem (MDCS). The running time of their algorithm is $O(n^k)$ if there are k colors and n points. Chen et al. [9] implement the algorithm in their geographical tagging system. Fleischer and Xu [10] show that MDCS can be solved in polynomial time

⋆ This work was supported by 2008-China Shenzhen Inovation Technology Program (SY200806300211A).

W. Wu and O. Daescu (Eds.): COCOA 2010, Part I, LNCS 6508, pp. 41–50, 2010.

for L_1 and L_∞ metrics, while it is NP-hard for all other L_p metrics even in two dimensions and gave a constant factor approximation algorithm.

There are also other works on colored point sets problems [3, 4, 11].

Problem Definition. The problems we discuss in this paper are as follows:

Problem 1. **Min-MST.** There are n points in the plane and each point is painted by one of m colors where $m \leq n$. We want to select m different color points such that the total edge length of resulting minimal spanning tree is as small as possible.

Problem 2. **Max-MST.**There are n points in the plane and each point is painted by one of m colors where $m \leq n$. We want to select m different color points such that the total edge length of resulting minimal spanning tree is as large as possible.

Problem 3. **Min-Per.**There are n points in the plane and each point is painted by one of m colors where $m \leq n$. We want to select m different color points such that the perimeter of the convex hull of m different color points is as small as possible.

We will prove those three problems are NP-complete by reduction from 3-SAT problem and give approximation algorithms for the third problem.

2 Min-MST Is NP-Complete

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 compute 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 reduction from 3-SAT problem. We need several gadgets to represent variables and clauses of 3-SAT formula. The general idea is for a 3-SAT formula, we put some colored points on the plane such that the given 3-SAT formula has a true assignment if and only if the length of minimum MST of colored points equals some given value.

First we draw a point O with distinct color at $(0, 0)$. For a 3-SAT formula ψ, suppose it has n variables $x_1, x_2, ..., x_n$ and m clauses. Let $x_{i,j,k}$ (or $\neg x_{i,j,k}$) be the variable x_i that appears at the j-th literal in ψ from left to right such that x_i (including $\neg x_i$) appears $k - 1$ times already in ψ before it. For each variable x_i $(1 \leq i \leq n)$, six points $p_i^{j'}$ $(1 \leq j' \leq 6)$ are created on $p_i^1 = (400i - 300, 0)$, $p_i^2 = (400i - 200, 0)$, $p_i^3 = (400i - 100, 0)$, $p_i^4 = (400i, 0)$, $p_i^5 = (300i - 100, -100)$, $p_i^6 = (300i, -100)$. For each literal $x_{i,j,k}$ (or $\neg x_{i,j,k}$), eight additional points are created on $p_{i,j,k}^7 = (400i - 200, -100)$, $p_{i,j,k}^8 = (400i, -100)$, $p_{i,j,k}^9 = (0, 400j - 300)$, $p_{i,j,k}^{10} = (0, 400j - 200)$, $p_{i,j,k}^{11} = (0, 400j - 100)$, $p_{i,j,k}^{12} = (0, 400j)$, $p_{i,j,k}^{13} = (-\frac{100}{3m}, 400j - 200)$, $p_{i,j,k}^{14} = (-\frac{100}{3m}, 400j)$ respectively. Among those fourteen points, p_i^5 and p_i^6 have the same color, $p_{i,j,k}^7$ and $p_{i,j,k}^{13}$ have the

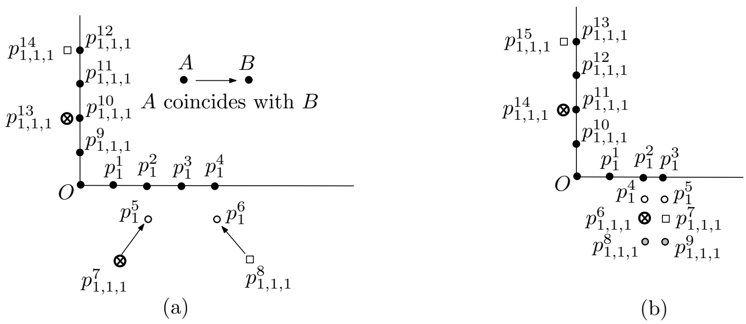

Fig. 1. Gadget for the first literal x_1 (a) for MIN-MST (b) for MAX-MST. Different symbols means different colors. All solid circles are with distinct colors.

same color, $p^8_{i,j,k}$ and $p^{14}_{i,j,k}$ have the same color, and all other points have different colors. Figure 1(a) shows gadget for the first literal x_1. According to above construction, we get a set P of $6n + 24m + 1$ points ($4n$ points are on x axis, $2n + 2 \times 3m = 2n + 6m$ points are on line $y = -100$, $4 \times 3m = 12m$ points are on y axis, $2 \times 3m = 6m$ points are on line $x = -\frac{100}{3m}$, one point is on point O) in the plane. Let T_P be the minimum MST over P. $4n + 12m + 1$ points $\subset P$ are on x and y axis and they are all with distinct colors. Therefore, all points on x and y axis and edges with length 100 connecting those points appear in T_P. Since p^5_i and p^6_i have the same color, $p^7_{i,j,k}$ and $p^{13}_{i,j,k}$ have the same color and $p^8_{i,j,k}$ and $p^{14}_{i,j,k}$ have the same color, we have to choose 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 minimum MST T_P with length $500n + 400 \times 3m + 3m \times \frac{100}{3m} = 500n + 1200m + 100$. Let choosing $\{p^5_i, p^7_{i,j,k}, p^{14}_{i,j,k}\}$ correspond x_i is *false* and choosing $\{p^6_i, p^8_{i,j,k}, p^{13}_{i,j,k}\}$ correspond x_i is *true* (see Figure 2).

Now, we illustrate how to represent the binary relation *or* in 3-SAT formula. We assume that the one clause is $x_{i1,j,k1} \vee x_{i2,j+1,k2} \vee x_{i3,j+2,k3}$. We create three *or* points $p^{or}_{i1,\lceil \frac{j}{3} \rceil}$, $p^{or}_{i2,\lceil \frac{j+1}{3} \rceil}$ and $p^{or}_{i3,\lceil \frac{j+2}{3} \rceil}$ with the same color (but different with all other colors) at $p^{13}_{i1,j,k1}$, $p^{13}_{i2,j+1,k2}$ and $p^{13}_{i3,j+2,k3}$ respectively. If one literal is in negation form $\neg x_{i,j,k}$, then we need to create the *or* point $p^{or}_{i,\lceil \frac{j}{3} \rceil}$ at $p^{14}_{i,j,k}$ instead of $p^{13}_{i,j,k}$. Figure 3 shows the gadget for the first clause $(x_1 \vee x_2 \vee \neg x_3)$ and Figure 4 shows the gadget for $\psi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_3 \vee x_4)$. Let T be the minimum MST over P plus $3m$ *or* points. For literal $x_{i,j,k}$ (or $\neg x_{i,j,k}$), if variable x_i is true (or false), then $\{p^6_i, p^8_{i,j,k}, p^{13}_{i,j,k}\}$ (or $\{p^5_i, p^7_{i,j,k}, p^{14}_{i,j,k}\}$) are chosen that means the *or* point $p^{or}_{i,\lceil \frac{j}{3} \rceil}$ can be chosen to construct T without adding any extra length comparing with T_P since $p^{or}_{i,\lceil \frac{j}{3} \rceil}$ is located at the same place as $p^{13}_{i,j,k}$ (or $p^{14}_{i,j,k}$). Therefore, if at least one literal is true in one clause, no extra length will be added to T_P for that clause. Otherwise, all three literals in one clause are false, then all three *or* points in that clause can not be covered by points in T_P. Since we have to choose one point from those three *or* points,

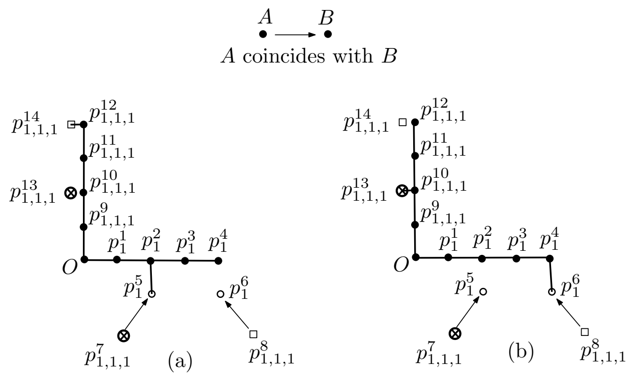

Fig. 2. Suppose x_1 is the first literal in the first clause of 3-SAT formula. (a) the minimum MST when x_1 is false. (b) the minimum MST when x_1 is true.

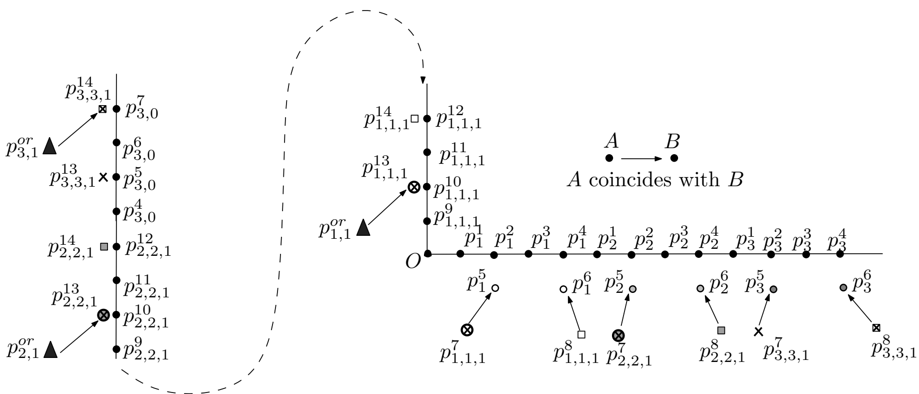

Fig. 3. Gadget for the clause $(x_1 \vee x_2 \vee \neg x_3)$ assuming it is the first clause

then the extra length of one will be added to the length of T_P. If m' clauses are not satisfiable, then the extra length of m' will be added to the length of T_P. Thus the lemma follows:

Lemma 1. *The 3-SAT formula ψ with n variables $x_1, x_2, ..., x_n$ and m clauses is satisfiable if and only if the length of T is equal to* $500n + 1200m + 100$.

3 Max-MST Is NP-Complete

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 compute the MST of those m points and check whether the length is at most L. This process can certainly be done in polynomial time.

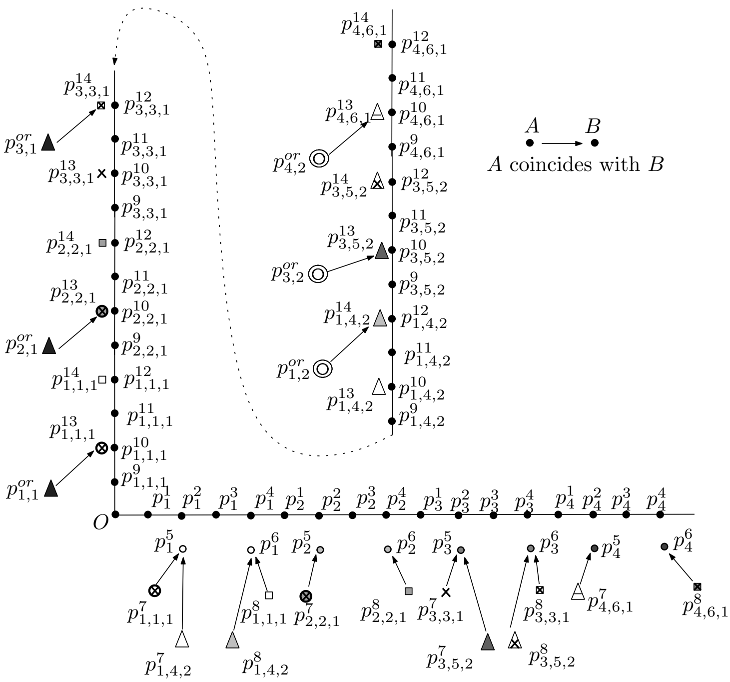

Fig. 4. Gadget for $\psi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_3 \vee x_4)$ for **Min-MST**. y axis is broken to save space.

We prove this problem is NP-hard by reduction from 3-SAT problem. We need several gadgets to represent variables and clauses of 3-SAT formula. The general idea is for a 3-SAT formula, we put some colored points on plane such that the given 3-SAT formula has true assignment if and only if the length of maximum MST of colored points equals some given value.

First we draw a point O with distinct color at $(0, 0)$. For a 3-SAT formula ψ, suppose it has n variables $x_1, x_2, ..., x_n$ and m clauses. Let $x_{i,j,k}$ (or $\neg x_{i,j,k}$) be the variable x_i that appears at the j-th literal in ψ from left to right such that x_i (including $\neg x_i$) appears $k-1$ times already in ψ before it. For each variable x_i $(1 \leq i \leq n)$, five points $p_i^{j'}$ $(1 \leq j' \leq 5)$ are created on $p_i^1 = (201i - 101, 0)$, $p_i^2 = (201i-1, 0)$, $p_i^3 = (201i, 0)$, $p_i^4 = (201i-1, -10)$, $p_i^5 = (201i, -10)$. For each literal $x_{i,j,k}$ (or $\neg x_{i,j,k}$), ten additional points are created on $p_{i,j,k}^6 = (201i-1, -9-2k)$, $p_{i,j,k}^7 = (201i, -9-2k)$, $p_{i,j,k}^8 = (201i-1, -10-2k)$, $p_{i,j,k}^9 = (201i, -10-2k)$, $p_{i,j,k}^{10} = (0, 400j - 300)$, $p_{i,j,k}^{11} = (0, 400j - 200)$, $p_{i,j,k}^{12} = (0, 400j - 100)$, $p_{i,j,k}^{13} = (0, 400j)$, $p_{i,j,k}^{14} = (-0.5, 400j - 200)$, $p_{i,j,k}^{15} = (-0.5, 400j)$ respectively. Among those fifteen points, p_i^4 and p_i^5 have the same color, $p_{i,j,k}^6$ and $p_{i,j,k}^{14}$ have the same color, $p_{i,j,k}^7$ and $p_{i,j,k}^{15}$ have the same color, $p_{i,j,k}^8$ and $p_{i,j,k}^9$ have the same color, and all other points have different colors. Figure 1(b) shows gadget for the first literal x_1. According to above construction, we get a set P of $5n + 30m + 1$

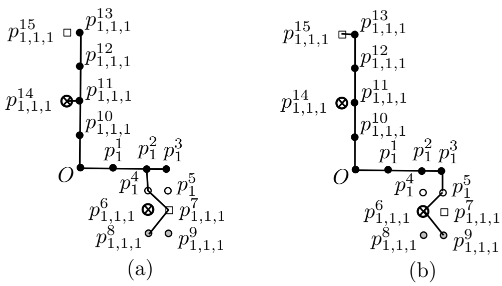

Fig. 5. (a) the maximum MST when x_1 is false. (b) the maximum MST when x_1 is true.

points in the plane. Let T_P be the maximum MST over P. $3n + 12m + 1$ points $\subset P$ are on x and y axis and they are all with distinct colors. Therefore, all points on x and y axis and edges with length 100 or 1 connecting those points appear in T_P. Since p_i^4 and p_i^5 have the same color, $p_{i,j,k}^6$ and $p_{i,j,k}^{14}$ have the same color, $p_{i,j,k}^7$ and $p_{i,j,k}^{15}$ have the same color, $p_{i,j,k}^8$ and $p_{i,j,k}^9$ have the same color, we have to choose either $\{p_i^4, p_{i,j,k}^7, p_{i,j,k}^8, p_{i,j,k}^{14}\}$ or $\{p_i^5, p_{i,j,k}^6, p_{i,j,k}^9, p_{i,j,k}^{15}\}$ to get the maximum MST T_P with length $(201+10)n + 3m(400+0.5+2\sqrt{2}) = 211n + 1201.5m + 6\sqrt{2}m$. Let choosing $\{p_i^4, p_{i,j,k}^7, p_{i,j,k}^8, p_{i,j,k}^{14}\}$ correspond x_i is *false* and choosing $\{p_i^5, p_{i,j,k}^6, p_{i,j,k}^9, p_{i,j,k}^{15}\}$ correspond x_i is *true* (see Figure 5).

For each literal $x_{i,j,k}$ (or $\neg x_{i,j,k}$), we create one *or* point $p^{or}_{i,\lceil \frac{j}{3} \rceil}$ at $p_{i,j,k}^{14}$ (or $p_{i,j,k}^{15}$) as in section 2. The remaining part of this section is similar to section 2.

Lemma 2. *The 3-SAT formula ψ with n variables $x_1, x_2, ..., x_n$ and m clauses is satisfiable if and only if the length of T is equal to $211n + 1201.5m + 6\sqrt{2}m$.*

4 Min-Per Problem Is NP-Complete

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 compute 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.

We prove this problem is NP-hard by reduction from 3-SAT problem. We need several gadgets to represent variables and clauses of 3-SAT formula. For a given instance of 3-SAT problem, suppose it has k variables $x_1, x_2, ..., x_k$. First we draw a circle C. The general idea is for a 3-SAT formula, we put some colored points on and inside C such that the given 3-SAT formula has true assignment if and only if the perimeter of convex hull CH_{opt} with minimum perimeter equals some given value.

For two points a, b on C, let $\widetilde{ab}$ be the arc of C from a to b in clockwise order and $\overline{ab}$ be the line segment between a, b. Let the length of $\overline{ab}$ be $|\overline{ab}|$. For

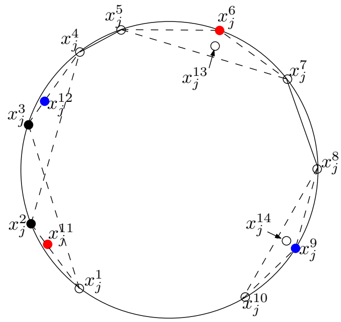

Fig. 6. The gadget for variable x_j. The colors of all empty circles are different from each other and from solid colored points. Solid line segments must appear on CH_{opt}. Dashed line segments mean they are candidates for edges of CH_{opt}.

each variable x_j, we put 10 points x_j^i where $i = 1, 2, ..., 10$ on circle C in such way: x_j^i where $i = 1, 2, ..., 10$ are arranged in clockwise order (see Figure 6). For those ten points, only x_j^2 and x_j^3 have the same color and all other points have different colors. Also $|\overline{x_j^1 x_j^2}| = |\overline{x_j^3 x_j^4}|$. Then we add two more points x_j^{11}, x_j^{12} where x_j^{11} is on line segment $\overline{x_j^1 x_j^2}$ and x_j^{12} is on line segment $\overline{x_j^3 x_j^4}$. x_j^{11} has the same color as the color of x_j^6 and x_j^{12} has the same color as the color of x_j^9. For different variables, the arcs they spanned do not overlap and they are arranged in clockwise order for increasing j. The relative distances between twelve points satisfy following equations:

redFormate changed:

$$|\overline{x_j^1 x_j^2}| + |\overline{x_j^2 x_j^4}| = |\overline{x_j^1 x_j^3}| + |\overline{x_j^3 x_j^4}| = p_1 |\overline{x_j^5 x_j^7}| = |\overline{x_j^8 x_j^{10}}| = p_2$$

$$|\overline{x_j^5 x_j^6}| = |\overline{x_j^6 x_j^7}| = |\overline{x_j^8 x_j^9}| = |\overline{x_j^9 x_j^{10}}| = p_3$$

$$|\overline{x_j^1 x_j^{11}}| + |\overline{x_j^{11} x_j^3}| - |\overline{x_j^1 x_j^3}| = |\overline{x_j^2 x_j^{12}}| + |\overline{x_j^{12} x_j^4}| - |\overline{x_j^2 x_j^4}| = \Delta p_1$$

$$2p_3 - p_2 = \Delta p_2; \Delta p_1 >> \Delta p_2$$

The inequality $\Delta p_1 >> \Delta p_2$ ensures the part of CH_{opt} on arc $\widetilde{x_j^1 x_j^4}$ has to be $\overline{x_j^1 x_j^2} \cup \overline{x_j^2 x_j^4}$ or $\overline{x_j^1 x_j^3} \cup \overline{x_j^3 x_j^4}$. Suppose we choose $\overline{x_j^1 x_j^2} \cup \overline{x_j^2 x_j^4}$ to be the part of CH_{opt}. In order to cover the color of x_j^{12}, we have to choose $\overline{x_j^8 x_j^9} \cup \overline{x_j^9 x_j^{10}}$ for arc $\widetilde{x_j^8 x_j^{10}}$. For the arc $\widetilde{x_j^5 x_j^7}$, since the color of x_j^6 is the same as the color of x_j^{11} and it has been covered by $\overline{x_j^1 x_j^2} \cup \overline{x_j^2 x_j^4}$ already, we can just choose $\overline{x_j^5 x_j^7}$ for arc $\widetilde{x_j^5 x_j^7}$. x_j^{11} acts as the negation of x_j^{12} that means if we choose one of them, the other one will not be chosen. Similarly, x_j^6 acts as the negation of x_j^9. Therefore, if the variable x_j is assigned to be 1, we choose $\overline{x_j^1 x_j^2} \cup \overline{x_j^2 x_j^4} \cup \overline{x_j^4 x_j^5} \cup \overline{x_j^5 x_j^7} \cup \overline{x_j^7 x_j^8} \cup \overline{x_j^8 x_j^9} \cup \overline{x_j^9 x_j^{10}}$ as part of CH_{opt} for the arc $\widetilde{x_j^1 x_j^{10}}$. Otherwise if the variable x_j is assigned to be 0, we choose $\overline{x_j^1 x_j^3} \cup \overline{x_j^3 x_j^4} \cup \overline{x_j^4 x_j^5} \cup \overline{x_j^5 x_j^6} \cup \overline{x_j^6 x_j^7} \cup \overline{x_j^7 x_j^8} \cup \overline{x_j^8 x_j^{10}}$ as part of CH_{opt} for the arc $\widetilde{x_j^1 x_j^{10}}$. Therefore, if we connect all k gadgets together to form CH_{opt}, the perimeter of CH_{opt} is $k \times (p_1 + 2p_2 + \Delta p_2) + p_4$ where $p_4 = 2\sum_{j=1}^{k}(|\overline{x_j^4 x_j^5}| + |\overline{x_j^7 x_j^8}|) + \sum_{j=1}^{k-1}|\overline{x_j^{10} x_{j+1}^1}| + |\overline{x_k^{10} x_1^1}|$.

For three literals in each clause, we add three points with same color into gadgets constructed above corresponding to three literals. For example, if x_j appears in one clause, we put x_j^{13} inside the triangle $\triangle x_j^5 x_j^6 x_j^7$ such that $\Delta p_3 = |\overline{x_j^5 x_j^{13}}| + |\overline{x_j^{13} x_j^7}| - |\overline{x_j^5 x_j^7}| << \Delta p_2$. If $\neg x_j$ appears in one clause, we put x_j^{14} inside the triangle $\triangle x_j^8 x_j^9 x_j^{10}$ such that $\Delta p_3 = |\overline{x_j^8 x_j^{14}}| + |\overline{x_j^{14} x_j^{10}}| - |\overline{x_j^8 x_j^{10}}| << \Delta p_2$. We call x_j^6 and x_j^9 as *apex-point* and x_j^{13} and x_j^{14} as *or-point.* For three literals in one clause, if all three literals are 0, then corresponding *apex-point* will not be chosen as the vertex of CH_{opt}. In order to cover the color of *or-point*, one of three *or-points* has to be chosen as the vertex of CH_{opt}. Then one Δp_3 will be add into the perimeter of CH_{opt}. Therefore, for a given 3-SAT formula with k variables, if it has true assignment, then the perimeter of convex hull CH_{opt} with minimum perimeter over the gadgets constructed as above equals $k \times (p_1 + 2p_2 + \Delta p_2) + p_4$.

If the perimeter of convex hull CH_{opt} over the gadgets constructed as above equals $k \times (p_1 + 2p_2 + \Delta p_2) + p_4$, that means at least one literal of all clauses is true. Because otherwise, the perimeter of convex hull CH_{opt} will be $\geq k \times (p_1 + 2p_2 + \Delta p_2) + p_4 + \Delta p_3$. Therefore, the 3-SAT formula has true assignment.

Theorem 1. *There are n points in the plane and each point is painted by one of m colors where $m \leq n$. To select m different color points such that the perimeter of the convex hull of m different color points is as small as possible is NP-complete.*

5 Approximation Algorithms for Min-Per Problem

5.1 π-Approximation Algorithm

The algorithm is simple: for each point p, select $m-1$ points $p_1, ..., p_{m-1}$ such that the colors of $p, p_1, ..., p_{m-1}$ are different and the distance from p_i $(1 \leq i \leq m-1)$ to p is less than the distances from other points with the same color as p_i to p. For those m points, construct a convex hull CH_p and compute its perimeter. For all n point, we get n perimeters and the minimum one is what we want. The running time of this algorithm is $O(n(n + m \log m))$. Assume the perimeter we get is per_{app} and the optimal perimeter is p_{opt}. Now we prove $per_{app} \leq \pi per_{opt}$.

Suppose two vertices that decide the diameter of the optimal convex hull CH_{opt} are p_a and p_b (see Figure 7). Let $r = |\overline{p_a p_b}|$. We draw a circle C with center p_a and radius r. CH_{p_a} is constructed as above algorithm. CH_{p_a} is totally inside C since CH_{opt} is totally inside C that means there are at least one point from each color is inside C. Then perimeter of CH_{p_a} is $per_{p_a} \leq 2\pi r$. We know that $per_{app} \leq per_{p_a}$ and $per_{opt} \geq 2r$. Therefore $per_{app} \leq \pi per_{opt}$.

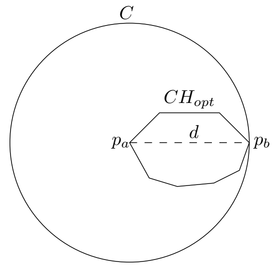

Fig. 7. Illustration of π-approximation algorithm

Theorem 2. *There is a π-approximation algorithm for problem 3 with running time $O(n^2 + nm \log m)$.*

5.2 $\sqrt{2}$-Approximation Algorithm

In [12], Abellanas et al. give an $O(\min\{n(n-m)^2, nm(n-m)\})$ algorithm for computing minimum perimeter axis-parallel rectangle R enclosing at least one point of each color. Let the perimeter of R be per_R and the diameter of R be d. If we construct CH_{app} over the points inside R, CH_{app} contains at least one point of each color and CH_{app} is totally inside R: Therefore $per_{app} \leq per_R$. There are some properties for R. There are at least one point on each edge of R. Colors of those four points are different. There are no other points inside R having the same color as those four points. Let R' be the smallest axis-parallel rectangle enclosing CH_{opt} and d' be the diameter of R'. Then $per_{opt} \geq 2d' \geq 2d$. We know $per_R \leq 2\sqrt{2}d$. Then $per_{app} \leq \sqrt{2} per_{opt}$.

Theorem 3. *There is a $\sqrt{2}$-approximation algorithm for problem 3 with running time $O(\min\{n(n-m)^2, nm(n-m)\})$.*

6 Conclusions

In this paper, we discussed three variations of spread colored points problems and proved NP-completeness of those three problems. For the third problem, we gave π and $\sqrt{2}$ approximation algorithms with $O(n^2 + nm \log m)$ and $O(\min\{n(n-m)^2, nm(n-m)\})$ running time respectively, where n is the number of points and m is the number of colors. In future work, it would be interesting to investigate more geometric problems for colored points. For example, maximizing the closest pair for spreading colored points.

References

1. Beresford, A.R., Stajano, F.: Location Privacy in pervasive Computing. IEEE Pervasive Computing 2(1), 46–55 (2003)
2. Gedik, B., Liu, L.: A customizable k-anonymity model for protecting location privacy. In: ICDCS, pp. 620–629 (2004)

3. Abellanas, M., Hurtado, F., Icking, C., Klein, R., Langetepe, E., Ma, L., Palop, B., Sacristan, V.: The farthest color Voronoi diagram and related problems. In: Proceedings of the 17th European Workshop on Computational Geometry (EWCG 2001), pp. 113–116 (2001)
4. Das, S., Goswami, P.P., Nandy, S.C.: Smallest Color-spaning Object Revisited. International Journal of Computational Geometry and Applications (IJCGA) 19(5), 457–478 (2009)
5. Pfoser, D., Jensen, C.: Capturing the uncertainty of moving-objects representations. In: Proc. SSDBM, pp. 111–132 (1999)
6. Sistla, P.A., Wolfson, O., Chamberlain, S., Dao, S.: Querying the uncertain position of moving objects. In: Temporal Databases: Research and Practice, pp. 310–337 (1998)
7. Löffler, M., van Kreveld, M.: Largest and Smallest Convex Hulls for Imprecise Points. Algorithmica 56(2), 235–269 (2008)
8. Ju, W., Luo, J.: New Algorithms for Computing Maximum Perimeter and Maximum Area of the Convex Hull of Imprecise Inputs Based On the Parallel Line Segment Model. In: CCCG 2009, pp. 1–4 (2009)
9. Chen, Y., Chen, S., Gu, Y., Hui, M., Li, F., Liu, C., Liu, L., Ooi, B.C., Yang, X., Zhang, D., Zhou, Y.: MarcoPolo: A community system for sharing and integrating travel information on maps. In: Proceedings of the 12th International Conference on Extending Database Technology (EDBT 2009), pp. 1148–1151 (2009)
10. Fleischer, R., Xu, X.: Computing Minimum Diameter Color-Spanning Sets. In: Lee, D.-T., Chen, D.Z., Ying, S. (eds.) FAW 2010. LNCS, vol. 6213. Springer, Heidelberg (2010)
11. Borgelt, M.G., van Kreveld, M., Löffler, M., Luo, J., Merrick, D., Silveira, R.I., Vahedi, M.: Planar bichromatic minimum spanning trees. J. Discrete Algorithms 7(4), 469–478 (2009)
12. Abellanas, M., Hurtado, F., Icking, C., Klein, R., Langetepe, E., Ma, L., Palop, B., Sacristan, V.: Smallest Color-Spanning Objects. In: Proc. 9th Annu. European Sympos. Algorithms, pp. 278–289 (2001)
13. Zhang, D., Chee, Y.M., Mondal, A., Tung, A.K.H., Kitsuregawa, M.: Keyword search in spatial databases: Towards searching by document. In: Proceedings of the 25th IEEE International Conference on Data Engineering (ICDE 2009), pp. 688–699 (2009)