Dispersing and grouping points on planar segments

Xiaozhou He, Wenfeng Lai, Binhai Zhu, Peng Zou

Dispersing and Grouping Points on Planar Segments

Xiaozhou He

Business School, Sichuan University, Chengdu, China

Wenfeng Lai

School of Computer Science and Technology, Shandong University, Qingdao, Shandong 266237, China

Binhai Zhu*

Gianforte School of Computing, Montana State University, Bozeman, MT 59717, USA

Peng Zou

Gianforte School of Computing, Montana State University, Bozeman, MT 59717, USA

Abstract

Motivated by (continuous) facility location, we study the problem of dispersing and grouping points on a set of segments (of streets) in the plane. In the former problem, given a set of n disjoint line segments in the plane, we investigate the problem of computing a point on each of the n segments such that the minimum Euclidean distance between any two of these points is maximized. We prove that this 2D dispersion problem is NP-hard, in fact, it is NP-hard even if all the segments are parallel and are of unit length. This is in contrast to the polynomial solvability of the corresponding 1D problem by Li and Wang (2016), where the intervals are in 1D and are all disjoint. With this result, we also show that the Independent Set problem on Colored Linear Unit Disk Graph (meaning the convex hulls of points with the same color form disjoint line segments) remains NP-hard, and the parameterized version of it is in W[2]. In the latter problem, given a set of n disjoint line

*Corresponding author.

Email addresses: xiaozhouhe126@qq.com (Xiaozhou He), 2290892069@qq.com (Wenfeng Lai), bhz@montana.edu (Binhai Zhu), peng.zou@student.montana.edu (Peng Zou)

segments in the plane we study the problem of computing a point on each of the n segments such that the maximum Euclidean distance between any two of these points is minimized. We present a factor-1.1547 approximation algorithm which runs in $O(n \log n)$ time. Our results can be generalized to the Manhattan distance.

Keywords: Dispersion Problem, NP-hardness, FPT, Manhattan Distance, Geometric Optimization

1. Introduction

Dispersion problems belong to the classic facility location problem and have been extensively studied. The goal of such a problem is to build facilities so that they are as far as possible. A typical example is to build a chain of convenience stores such that they should be far from each other to cover more customers. As a matter of fact, a series of research has been done, either over a point set or over a weighted graph [2, 6, 7, 10, 18, 19, 20].

In [12, 13], Li and Wang studied an interesting variation of the problem, where one is given a set of disjoint intervals in 1D and the objective is to put one point on each interval such that the minimum distance between any two of these computed points is maximized. Assuming the intervals are sorted, an optimal linear time greedy algorithm was given (but the analysis is non-trivial). The scenario corresponding to this problem can be considered as constructing resting areas along a highway, where each interval is some section suitable for constructing a resting area.

A natural question arises: what if we are given some disjoint (rectilinear) segments (where each segment is part of a street)? (Here the objective function is the same while the distance could be either Euclidean (L_2) or Manhattan (L_1).) We show that this problem is NP-hard; in fact, NP-hard even when all the segments are parallel (i.e., along one direction) and are of a unit length. It turns out that this is related to the Independent Set (IS) problem on a unit disk graph (UDG) [15, 4]; in fact, our NP-hardness proof implies that the problem remains NP-hard even when the unit disk graph is *colored* and *linear* (meaning the convex hulls of points of the same color form disjoint line segments). We suspect that the parameterized version remains to be W[1]-hard, though we are only able to show that it is in W[2] at this point.

The symmetric problem of grouping points on a set of disjoint segments in the plane, i.e., selecting one point on each segment such that the maximum distance between the selected points is minimized, is motivated by constructing commodity distribution centers within a road network. These centers should be close to each other to reduce the distribution or transportation costs. It is not known whether the problem is NP-hard yet, though we are able to show that this problem admits a factor-1.1547 approximation running in $O(n \log n)$ time.

This paper is organized as follows. In Section 2, we give the preliminaries. In Section 3, we prove that the 2D dispersion problem is NP-hard. In Section 4, we consider briefly the independent set problem on colored linear unit disk graphs and prove its W[2] membership. In Section 5, we give a simple polynomial time approximation algorithm for the 2D grouping problem. We conclude the paper in Section 6.

2. Preliminaries

2.1. Definitions

Given two points $a = (x_a, y_a), b = (x_b, y_b)$ in the plane (2D), the Euclidean or L_2 distance $d(a, b) = d_2(a, b)$ is defined as $d(a, b) = ((x_a - x_b)^2 + (y_a - y_b)^2)^{1/2}$. The Manhattan or L_1 distance $d_1(a, b)$ is defined as $d_1(a, b) = |x_a - x_b| + |y_a - y_b|$. A line segment with endpoints a and b is denoted as $l = (a, b)$ or $l = ab$.

Finally, a planar unit disk graph is one where each vertex corresponds to a given point of an input set of planar points, two vertices u, v share an edge if two disks of radii R centered at u, v intersect each other. (Note that the standard unit disk graph definitions require that $R = 1/2$, in our definition R could be more general as long as its value is fixed.) It is known that while most NP-hard problems on general graphs remain NP-hard on unit disk graphs [5, 8], there are exceptions (e.g., the maximum clique problem is polynomially solvable [8]). Notably, the parameterized version of the independent set problem on unit disk graphs, parameterized by the size of the solution, is known to be W[1]-hard [15]. The more restricted colored version, parameterized by the number of colors, remains to be W[1]-hard [4].

2.2. Problems

The problems studied in this paper are defined as follows:

2D Dispersion Problem: Given a set of disjoint line segments $L = \{L_1, L_2, ..., L_n\} \subset \mathbb{R}^2$, the goal is to find n points $V = \{v_1, v_2, ..., v_n\}$ on the n line segments respectively such that the minimum value of the distances between any two selected points in V is maximized, i.e.,

$$\max_{v_i \in L_i, i=1..n} \min_{v_i, v_j \in V} d(v_i, v_j).$$

2D Grouping Problem: Given a set of disjoint line segments $L = \{L_1, L_2, ..., L_n\} \subset \mathbb{R}^2$, the goal is to find n points $V = \{v_1, v_2, ..., v_n\}$ on the n line segments respectively such that the maximum distance among two points in V is minimized, i.e.,

$$\min_{v_i \in L_i, i=1..n} \max_{v_i, v_j \in V} d(v_i, v_j).$$

We show in the next section that the 2D dispersion problem is NP-hard.

3. NP-hardness for the 2D Dispersion Problem

We reduce the NP-complete planar rectilinear monotone 3-SAT problem to the 2D Dispersion problem. The planar 3-SAT problem is a special case of 3-SAT where the input is a conjunction of a set of disjunctive clauses, each with three literals. Knuth and Raghunathan showed that the planar rectilinear 3-SAT, where all the clauses can be embedded in a rectilinear way below a line, is NP-complete [14]. More recently, de Berg and Khosravi showed that even the planar rectilinear monotone 3-SAT, where all clauses contain either all positive or all negated literals, is NP-complete [3]. In this case, the 3-SAT instance can be embedded onto a rectilinear grid (with no edge crossing) such that all the variables can be embedded on a horizontal line L. The clauses with all literals positive can be embedded as a three-legged grid point below L, connecting to the corresponding variable; symmetrically, the clauses with all literals negated can be embedded as a three-legged grid point above L, connecting to the negated variables. For example, we show in Fig. 1 a layout of 5 clauses over 5 variables, e.g, $C_1 = (\bar{v}_1 \vee \bar{v}_4 \vee \bar{v}_5)$ and $C_2 = (v_1 \vee v_3 \vee v_5)$.

When we reduce planar rectilinear monotone 3-SAT to the 2D Dispersion problem, all the constructed segments in a set U are of unit length and are horizontal.

Theorem 1. *The 2D Dispersion Problem is NP-hard.*

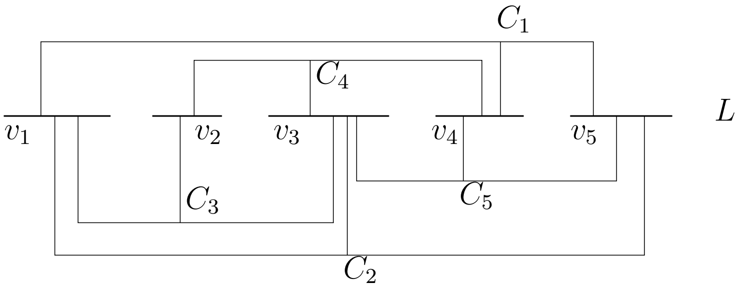

Figure 1: The rectilinear planar graph $G(\phi)$ for the planar rectilinear monotone 3-SAT instance ϕ with 5 clauses over 5 variables.

Proof. First, we embed the planar rectilinear monotone 3-SAT graph $G(\phi)$ onto a rectilinear grid. For the ease of calculation, each grid has a horizontal/vertical length of 1, e.g., in Fig. 2, $d(a, b) = d(a, c) = 1$ and $d(a, d) = \sqrt{2}$. Each variable v on the line L is represented as a sequence of segment pairs (e.g., (a, b) and (c, d)), placed at every other grid (Fig. 2). In Fig. 2, we label the grids horizontally with red integers and vertically with blue integers. If v appears in some clause C positively, we connect a leg of segment pairs from v starting at an oddly numbered grid (e.g., the one starting at grid 5 in Fig. 2); and symmetrically, if v appears in some clause C negatively, we connect a leg of segment pairs from v starting at an evenly numbered grid (e.g., the one starting at grid 10 in Fig. 2). The white points are selected for segments corresponding to v if v is assigned 'True'; otherwise, black points are selected.

We now describe the clause gadget in detail. We first illustrate a logical construction in Fig. 3, where clause C is composed of three literals u, v and w, and v connects C from L vertically while the other two horizontally through some bending. The core of the clause is a horizontal segment (p, q) of length 1, with a midpoint o. (Note that o is at a grid point while p and q are not. Note also that the dashed segments in this gadget are only for illustration purpose. In fact, they are only used to illustrate the distance between the points.) The length of the dashed segments $(a, p), (a, q), (o, f)$ and (o, g) is $\ell = \sqrt{2}$. The black points $\{a, f, g\}$ are selected for the 'False' assignment of these literals. Clearly, in this case no matter what point r we select for the segment pq, the minimum distance of $d(r, a), d(r, f)$ and $d(r, g)$ would be less than $\ell = \sqrt{2}$. On the other hand, if at least one of $\{a, f, g\}$ is not selected, then such an r can be easily selected. For example, if only b is selected (instead of a) for segment ab then we could select $r = o$ such that the minimum distance of $d(r, b), d(r, f)$ and $d(r, g)$ would be be $\sqrt{2}$.

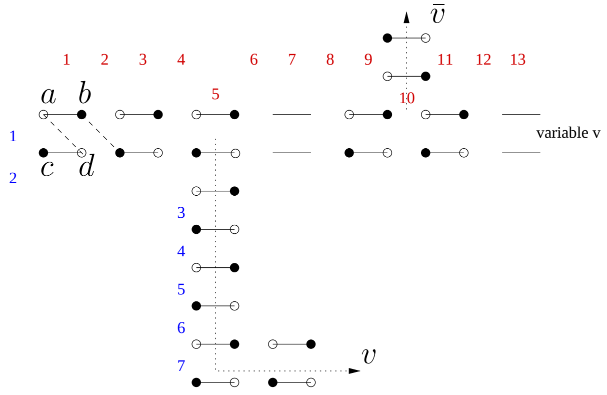

Figure 2: The variable gadget. For convenience, we use red numbers to label the grids horizontally and for vertical labeling we use blue numbers. Clearly, the leg of v starts at an odd horizontal grid (e.g., 5), while $\bar{v}$ starts at an even horizontal grid (e.g., 10).

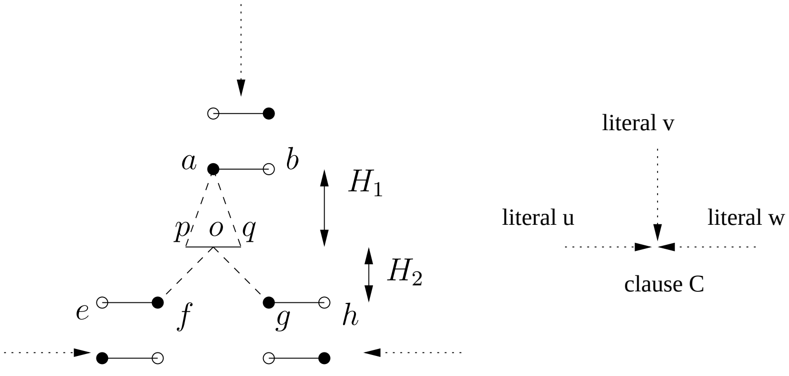

Figure 3: The (logical) clause gadget $C = (u \vee v \vee w)$. In this figure, we have $H_1 = \sqrt{7}/2$ and $H_2 = 1$.

However, it is clear that the clause we depict in Fig. 3 is not yet a valid planar embedding. This is firstly because the vertical distance between ab and pq, $H_1 = \sqrt{7}/2$, is not an integral multiple of the unit grid length 1; moreover, since $d(f, g) = 2$ we cannot lead u or w directly from the corresponding variable gadgets. It turns out that the latter can be easily fixed with a (local) horizontal adjustment on w, as shown in Fig. 4. On the other hand, the vertical adjustment on v, sketched also in Fig. 4, needs more details which is shown in Fig. 5.

Note that, from Fig. 4, the vertical distance between cd and pq is $6 = 4H + H_1 = 4H + \sqrt{7}/2$. In other words, $H = (6 - \sqrt{7}/2)/4 \approx 1.169$. In Fig. 5, we show how such a value of H can be implemented. We put a point t on cd with $d(d,t) > d(c,t)$. Recall that $\ell = \sqrt{2}$. Then $d(d,t) = \sqrt{\ell^2 - H^2} \approx 0.7959$, and $d(c,t) = 1 - d(d,t) \approx 0.2041$. Therefore, with 8 segments which are not on any grid line we successfully implement the (local) vertical adjustment for variable v on clause C, with all exterior segments around C on the grids. Note that when we select c for segment cd, it is possible that any point on segment c_1d_1 within distance 0.2041 from t_1 could be selected; on the other hand, when d is selected for segment cd then c_1 must be selected for segment c_1d_1.

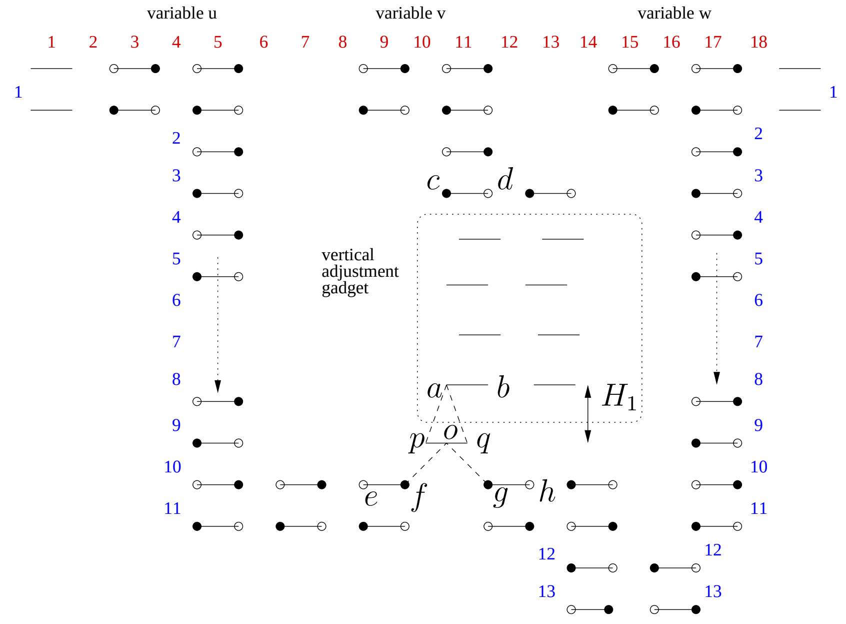

Figure 4: The clause gadget $C = (u \vee v \vee w)$ with a horizontal adjustment on w and a vertical adjustment on v. Note that $H_1 = \sqrt{7}/2$. Again, we use red numbers to label the grids horizontally and we use blue numbers for vertical labeling. Clearly, the legs of u, v and w all start at odd horizontal grids (e.g., 5, 11 and 17 respectively).

Now we have finished showing a complete example of the construction for the clause $C = (u \vee v \vee w)$ in Fig. 4 and Fig. 5. Note that all the segments connecting the variable and clause gadgets are horizontal and are of a unit length. A clause $C' = (\bar{u} \vee \bar{v} \vee \bar{w})$ can be constructed symmetrically above the line L. Due to the structure in Fig. 1, once $G(\phi)$ is given we could construct the corresponding rectilinear embedding clause by clause, starting from the innermost one from L, where the innermost clause is simply defined as the one whose vertical distance to the line L is minimum. (For example, for the clauses below L, C_5 in Fig. 1 would be constructed first, then C_3, and finally C_2.) The total number of segments is proportional to the size of $G(\phi)$, i.e., the construction can be done in linear time.

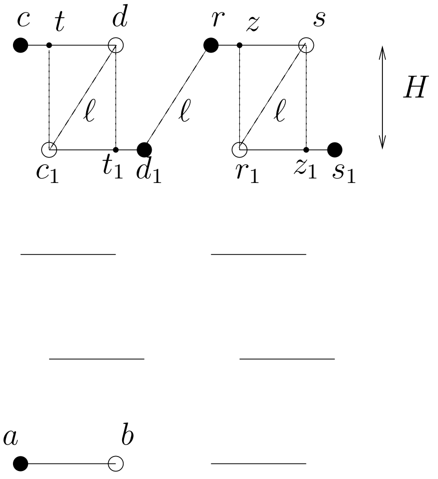

Figure 5: A (local) vertical adjustment on variable v.

Finally, we claim that the planar rectilinear monotone 3-SAT instance ϕ has a valid truth assignment if and only if in the converted 2D Dispersion instance the minimum distance of any two chosen points is equal to $\ell = \sqrt{2}$.

'If' part: If the planar rectilinear monotone 3-SAT instance ϕ has a truth assignment, then either all black points or all white points in all the variable gadgets would be selected. The closest distance of two chosen points in the variable gadgets is equal to ℓ. In the clause gadget, at least one variable needs to be assigned 'True' (or, white points need to be selected) for the clause. Hence, at least one black point $\{a, f, g\}$ is not selected in the corresponding clause gadget. The minimum distance of any two chosen points in the clause gadget is ℓ (except possibly the points in the vertical adjustment gadget for variable v in Fig. 4). Therefore, in the converted instance for 2D Dispersion, the minimum distance of any two selected points is ℓ.

'Only if' part: Suppose that the converted instance for 2D Dispersion has a minimum distance of ℓ between two selected points. Firstly, we note that in all the variable gadgets, if we want to maximize the minimum distance of any two selected points, then either all white points or all black points must be chosen. At this point the minimum distance of any two such selected points is exactly ℓ. In the clause gadget, if the maximum of the minimum distance of any two chosen points is at least ℓ, then at least one white point $x \in \{b, e, h\}$ needs to be selected. This implies that in the corresponding clause, the corresponding variable v_x is assigned 'True' if the clause contains all positive literals; or, the corresponding literal $\bar{v}_x$ is assigned 'False' if the clause contains all negative literals. This implies that the corresponding clause is evaluated 'True' for either cases. Therefore, if in the converted 2D dispersion instance the minimum distance of any two chosen points is at least ℓ, the corresponding planar rectilinear monotone 3-SAT instance ϕ is satisfied. □

Since the problem is NP-hard even when all the input segments are horizontal and are of unit length, the general version of this problem is also NP-hard. We comment that the NP-hardness holds even when the L_1 (or Manhattan) distance is used, with $\ell = 2$ and minor modification on the local vertical adjustment gadget.

4. Hardness for IS on Colored Linear Unit Disk Graphs

The NP-hardness result in the previous section has a direct implication on the Independent Set (IS) problem on Colored Linear Unit Disk Graph, which is a unit disk graph such that the convex hull of the points in the same color form a line segment, no two such segments intersect and the problem is to select one node (disk) in each color such that they form an independent set. We briefly go through the implication in this section.

Before doing that, we note that finding k-multicolored clique or independent set was initially motivated in proving the W[1]-hardness of some graph problems [9]. For geometric intersection graphs (specifically, unit disk graphs) Marx showed the the Independent Set problem is W[1]-hard, which implies it is not possible to obtain an FPT (fixed-parameter tractable) algorithm unless FPT=W[1] [15]. Moreover, the W[1]-hardness of the problem implies that it is not possible to obtain an EPTAS (efficient PTAS, namely the running time is $f(\epsilon) \cdot n^c$, where n is the input size). Consequently the PTAS by Hunt et al. (with running time $O(n^{1\epsilon})$) [11] cannot be further improved to have an EPTAS. Bereg et al. investigated the k-multicolored independent set problem on unit disk graphs and proved its W[1]-hardness and with that, they proved that the Largest Closest Pair Color-Spanning Set problem is W[1]-hard [4]. We now briefly sketch our results.

Theorem 2. *The Independent Set problem on Colored Linear Unit Disk Graph is NP-hard, even when all the segments are parallel and are of a unit length.*

Proof. We just use the reduction for Theorem 1. The two changes are: (1) put $1/\epsilon$ points evenly on each segment, (2) the corresponding unit disk graph is formed by drawing an open disk centered at each point of each segment with a radii $\ell/2 = \sqrt{2}/2$; moreover, all the disks centered at the same segment have the same color. Let U be the set of unit-length segments. Calling the resulting unit disk graph G_U, we clearly still have the following statement: the planar rectilinear monotone 3-SAT instance ϕ is satisfiable if and only if G_U has an independent set of size $|U|$. The details are standard and omitted. □

An immediate question is the FPT tractability of the parameterized version of the problem, where the parameter k is the number of segments, or the number of colors. Note that the general version, where the points of the same color could be arbitrarily distributed, is W[1]-hard [4]. However, in the current version the unit disk graph is more restricted. Nevertheless we show below that it is in W[2].

Theorem 3. *The Independent Set problem on Colored Linear Unit Disk Graph, parameterized by the number of colors (e.g. segments), is in W[2].*

Proof. Let the number of segments be k and let the set of linear points on segment L_i be ordered as $V_i = \langle v_{i,1}, ..., v_{i,j}, ..., v_{i,P(i)} \rangle$. We need to decide whether a value R exists such that the set S of points selected, one for each segment (color), has the property that the closest pair has distance at least R (or, equivalently, the unit disk graph with radii $R/2$ on all these points has a colorful independent set, i.e., one for each color).

We construct a circuit C as follows: the inputs are variables corresponding to all the points (for convenience, we still use $v_{i,j}$'s as variables, with a boolean value 1 assigned to $v_{i,j}$ meaning that point is selected). For all the points on the same segment L_i, we construct a large OR ($\vee$) gate. Here, 'large' means the input to the OR gate could be greater than 2; and to make the OR gate output a true value, one of these points must be selected.

For two points on two segments $v_{i,i'}$ and $v_{j,j'}$, if their distance is shorter than R then we cannot select both of them. This can be interpreted as $\neg(v_{i,i'} \wedge v_{j,j'})$. In fact, as the distance function from point $v_{i,i'}$ to all (sorted) points on L_j (i.e., V_j) is unimodal, we could construct these nested $\neg$ and $\wedge$ gates in one pass when $v_{i,i'}$ and V_j are fixed.

Finally, we connect all these OR ($\vee$)gates and NOT ($\neg$) gates to a large AND ($\wedge$) gate, which is the output of this circuit. It is easy to see that the IS problem on Colored Linear Unit Disk Graph (with radii $R/2$) has a solution if and only if k variables are selected to have a 'True' output. As from any input gate to the final output we have at most two wefts (large gates), the W[2] membership is hence shown. □

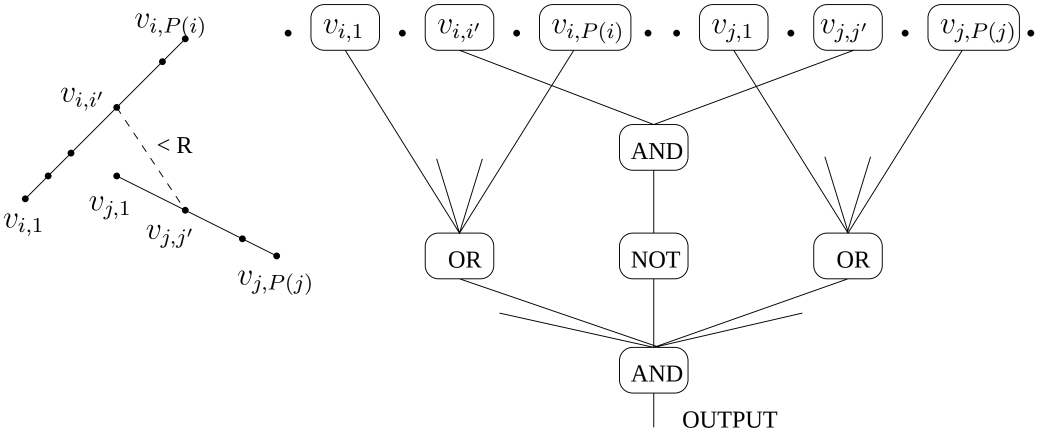

Figure 6: The circuit for the IS problem on Colored Linear Unit Disk Graph.

We comment that this proof is similar to the one given for the Minimum Diameter Color-Spanning Set problem by Pruente [17]. However, it is tantalizingly open whether this version of the IS problem on unit disk graphs is W[1]-hard. We comment that the unimodality property between $v_{i,i'}$ and V_j might be used either to show its W[1]-hardness or its membership in FPT.

5. Approximation for the 2D Grouping Problem

In this section, we design a factor-1.1547 polynomial-time approximation algorithm for the 2D grouping problem. It turns out that this problem is also related to unit disk graph. Recall that the input to the problem is a set of disjoint line segments $L = \{L_1, L_2, ..., L_n\}$. Assuming that we have an optimal solution $V^* = \{v_1, v_2, ..., v_n\}$ where each v_i is selected from L_i and the maximum distance between v_i, v_j is d^*. Let $C_i(r)$ be a closed disk centered at v_i with radius r. Then, using the so-called *intersection model* for a unit disk graph, if we draw a disk $C_i(\sqrt{3}d^*/3)$ at each v_i, these disks would have a common intersection. We prove this property formally as follows.

Property 1. *Using the so-called* intersection model *for a unit disk graph, the disks $C_i(\sqrt{3}d^*/3)$ centered at every $v_i \in V^*$ respectively, where d^* is the maximum distance between $v_i, v_j \in V^*$, would have a common intersection.*

Proof. By definition, for any two points $v_i, v_j \in V^*$, the distance $d(v_i, v_j) \leq d^*$. Hence, there must exist a circle $C(\sqrt{3}d^*/3)$ centered at some place which contains all the points in V^* — the extreme case occurs when the convex hull of V^* forms a regular triangle of edge length d^*. Therefore, if we draw the circles $C_i(\sqrt{3}d^*/3)$ at each $v_i \in V^*$, these circles have a common intersection. □

5.1. The Minimum Intersecting Disk Problem

Based on this property, we define a related Minimum Intersecting Disk problem:

Minimum Intersecting Disk (MID) Problem: Given a set of disjoint line segments $L = \{L_1, L_2, ..., L_n\} \subset R^2$, compute a disk $C(c, r)$ with center c and radius r such that all L_i's intersect $C(c, r)$ and r is minimized (at r^+).

We present a polynomial-time algorithm to solve the MID problem in this subsection. We first obtain a simple decision procedure *Decide*(r) which decides whether a disk of radius r exists that intersect all segments in L. Note that a segment completely in $C(c, r)$ is also considered being intersected by $C(c, r)$.

Algorithm 1 Decide(r): decides if there is a disk of radius r intersecting L

```
For each segment $L_i$, compute the Minkowski sum of $L_i$ with a disk of radius $r$, denoted as $L_i \oplus C(r)$.
Compute the common intersection $F(r)$ of all $L_i \oplus C(r)$, i.e., $F(r) = \cap_i(L_i \oplus C(r))$.
If $F(r)$ is not empty then return YES and any point in $F(r)$ as a witness for the center of such a disk; otherwise, return NO.
```

Note that all $L_i \oplus C(r)$ are convex, so their common intersection can be computed using a standard method in computational geometry, e.g., the incremental construction method (i.e., maintain the current common intersection, insert the next one, and update to have the new common intersection). Due to convexity, each update takes $O(\log n)$ time. Hence, $F(r)$ can be computed in $O(n \log n)$ time.

To compute r^+ by applying the above decision procedure *Decide*(r) using binary search, we first need to compute all possible values of r^+.

Claim 1. *All possible values of r^+ can be computed in $O(n^3)$ time.*

Proof. We first notice that each possible value is decided by either two or three segments, i.e., as long as the circle $C(r^+)$ intersects these two or three segments it would intersect all the segments in L: (1) In the former case the circle $C(r^+)$ touches two segments, each on a single point exactly, and thus, r^+ is half of the distance between the two closet points on these two segments. (2) In the latter case r^+ is the radius of the circle tangent to three segments. With the above analysis, all possible values of r^+ can be obtained by computing all such distances. For the former case we compute the distance between the two closest points on every pair of segments in L, respective, and the combination of two distinct segments is $O(n^2)$. For the latter case we compute the radius of the inscribed circle formed by three segments in L, and the combination of three distinct segments is $O(n^3)$. All in all, the total time to compute all possible values of r^+ is $O(n^3)$. □

We can then obtain the following algorithm.

Algorithm 2 Solution for MID

```
Compute a sorted list $\mathcal{L}$ which contains all the possible values of the radii $r^+$ as shown in Claim 1.
Binary search in $\mathcal{L}$ using Decide$(r)$ to compute the disk $C(c^+, r^+)$ with center $c^+$ and with the minimum radius $r^+$, which intersects all segments by construction.
Return $C(c^+, r^+)$.
```

We comment that binary search would find the optimal solution as its value must be contained in $\mathcal{L}$. Regarding the time complexity, we note that $|\mathcal{L}| = O(n^3)$, hence $\mathcal{L}$ can be constructed and then sorted in $O(n^3 \log n)$ time. Using the binary search, with $D(r)$ as a subroutine, the cost is $O(n \log n) \times$

$O(\log n^3) = O(n \log^2 n)$. The total cost is therefore $O(n^3 \log n) + O(n \log^2 n) = O(n^3 \log n)$. Note that this algorithm, though easy to implement, might be too slow when n is large.

We could in fact use the farthest Voronoi diagram for line segments [1, 16] to improve this time bound. The definition of the *farthest line segment Voronoi diagram* is as follows.

Definition 1. *Given n line segments $L = \{L_1, ..., L_n\}$ in the plane, the farthest Voronoi region of a line segment $L_i \in L$ is $freg(L_i) = \{x \in \mathbb{R}^2 | d(x, L_i) \geq d(x, L_j), \forall 1 \leq j \leq n\}$, where $d(x, L_i)$ is the distance between a point x with the closet point on L_i, and the collection of all* farthest Voronoi *regions, together with their bounding edges and vertices, constitute the farthest line segment Voronoi diagram of the n segments.*

Then, we claim that computing such a farthest line segment Voronoi diagram, the center of MID must lie either on a Voronoi edge or at a Voronoi vertex.

Claim 2. *The center of the minimum intersecting disk of n line segments $L = \{L_1, ..., L_n\}$ lie either on a bounding edge or a bounding vertex of the farthest line segment Voronoi diagram of the n line segments.*

Proof. As stated in the proof of Claim 1, the radius of the MID is decided by two or three segments, respectively. Thus, we assume w.l.o.g. that the MID $C(c, r^+)$ touches segments L_i and L_j exactly on one point, e.g., v_i and v_j respectively. Then, $d(c, v_i) = d(c, v_j) = r^+$, and any other point on L_i or L_j is of a farther distance to the center c; hence, $d(c, L_i) = d(c, L_j) = r^+$. On the other hand, for any segment $L_k \in L$ that is distinct from L_i and L_j, the MID intersects L_k on at least one point, assumed as v_k, and $d(c, v_k) \leq r^+$; hence, $d(c, L_k) \leq r^+$. Therefore, the center c lies on the bounding edge of the Voronoi regions of segments L_i and L_j. Similarly, for the case where the MID is decided by three segments, we can also show that the center c lies on the common vertex of the Voronoi regions of these three segments. □

With Claim 2, a candidate intersecting disk can be computed in $O(1)$ time no matter when the center lies on a bounding vertex or a bounding edge. We thus have the following theorem.

Theorem 4. *The Minimum Intersecting Disk problem can be solved in $O(n \log n)$ time.*

Proof. The farthest line-segment Voronoi diagram can be computed in $O(n \log n)$ time with n input segments [1, 16]. Moreover, such a diagram has a linear size (i.e., $O(n)$ number of edges and vertices). Hence, with a linear search we could identify $O(n)$ candidate intersecting disks and return the smallest one. Therefore, the MID problem can be solved in $O(n \log n)$ time. □

5.2. Approximation Factor Analysis

Our approximation algorithm for the 2D Grouping problem is to compute the minimum intersecting disk $C^+ = C(c^+, r^+)$ for L, and return the maximum distance $d(C^+)$ between the two or three points defining this disk $C(c^+, r^+)$.

Let C^* be the minimum radius disk, with radius $r(C^*)$, enclosing all the selected points of L in an optimal solution for the 2D grouping problem on L. Let $d(C^*)$ be the maximum distance between the two or three points defining C^*. Let *Opt* be the the maximum distance between the selected points in the optimal 2D grouping problem.

We have the following lemma.

Lemma 1. $r(C^*) \leq \frac{d(C^*)}{\sqrt{3}}$.

Proof. When C^* is defined by two points, apparently we have $r(C^*) = \frac{d(C^*)}{2}$.

When C^* is defined by three points, we have $r(C^*) \leq \frac{d(C^*)}{\alpha}$. This α is minimized when the three defining points for C^* form a regular triangle, where we have $\alpha = \sqrt{3}$. □

Since *Opt* is the maximum distance between the selected points on L (which are inside the circle C^*) and $d(C^*)$ is the maximum distance between the two or three points defining C^*, we have $d(C^*) \leq Opt$. Let *App* be the solution of the approximation solution. We then combine all the arguments together as follows.

$$
\begin{aligned}
App &= d(C^+) \\
&\le 2 \cdot r^+ \\
&\le 2 \cdot r(C^*) \quad (by\ the\ optimality\ of\ r^+) \\
&\le \frac{2}{\sqrt{3}} \cdot d(C^*) \quad (by\ Lemma\ 1) \\
&\le \frac{2}{\sqrt{3}} \cdot Opt \quad (by\ the\ optimality\ of\ Opt) \\
&\le 1.1547 \cdot Opt.
\end{aligned}
$$

Therefore, we have the following theorem.

Theorem 5. *There is a factor-1.1547 approximation algorithm for the 2D grouping problem which runs in $O(n \log n)$ time.*

We comment that the algorithm would still work if L_1 or Manhattan distance is used, with the approximation factor increased by an additional $\sqrt{2}$ factor.

6. Concluding Remarks

We study a general version of the 2D dispersing problem, whose 1D counterpart was recently studied by Li and Wang [12, 13]. We prove that the 2D dispersion problem is NP-hard, in fact, NP-hard even when all segments are parallel and are of unit length. The proof can be applied to show the maximum independent set problem on colored linear unit disk graph remains to be NP-hard. We also consider the symmetric problem of grouping a set of n points, one on each input segment, such that the maximum distance between any two selected ones is minimized. For the latter problem, we give a factor-1.1547 approximation which runs in $O(n \log n)$ time.

We have several open problems related to this paper:

1. Does the 2D dispersion problem admit a constant factor approximation?
2. Is the Independent Set problem on Colored Linear Unit Disk Graph W[1]-hard?
3. Is the 2D grouping problem polynomial-time solvable?

Acknowledgments

This research is supported by NSF of China under project 72001153 and 61628207. XH is supported by the Fundamental Research Funds for the Central Universities under Project skbsh2020-19. We thank anonymous reviewers for several useful comments which greatly improve the presentation of this paper.

References

[1] F. Aurenhammer, R.L.S. Drysdale and H. Kraser. Farthest line segment Voronoi diagrams. *Infor. Proc. Lett.*, 100:220-225, 2006.

[2] C. Baur and S. Fekete. Approximation of geometric dispersion problems. *Proc. of APPROX'98*, pp. 63-75, 1998.

[3] M. de Berg and A. Khosravi. Optimal binary space partition for segments in the plane. *Intl. J. Comput. Geom. Appl.*, 22(3):187-206, 2012.

[4] S. Bereg, F. Ma, W. Wang, J. Zhang, and B. Zhu. On some matching problems under the color-spanning model. *Theor. Comput. Sci.*, 786:26-31, 2019.

[5] H. Breu. *Algorithmic aspects of constrained unit disk graphs*. PhD dissertation, Department of Computer Science, UBC, Canada, 1996.

[6] A. Cevallos, F. Eisenbrand and R. Zenklusen. Local search for max-sum diversification. *Proc. of SODA'17*, pp. 130-142, 2017.

[7] B. Chandra and M. Halldorsson. Approximation algorithms for dispersion problems. *J. of Algorithms*, 38:438-465, 2001.

[8] B. Clark, C. Colbourn, and D. Johnson. Unit disk graphs. *Discrete Mathematics*, 86(1-3):165-177, 1990.

[9] M. Fellows, D. Hermelin, F. Rosamond, S. Vialette On the parameterized complexity of multiple-interval graph problems. *Theor. Comput. Sci.*, 410(1):53-61, 2009.

[10] R. Hassin, S. Rubinstein and A. Tamir. Approximation algorithms for maximum dispersion. *Oper. Res. Lett.*, 21:133-137, 1997.

[11] H.B. Hunt III, M.V. Marathe, V. Radhakrishnan, S.S. Ravi, D.J. Rosenkrantz, R.E. Stearns. NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs. *J. Algorithms*, 26(2):238274, 1998.

[12] S. Li and H. Wang. Dispersing points on intervals. *Proc. of ISAAC'16*, LIPIcs 64, 52:1-52:12, 2016.

[13] S. Li and H. Wang. Dispersing points on intervals. *Disc. Appl. Math.*, 239:106-118, 2018.

[14] D.E. Knuth and A. Raghunathan. The problem of compatible representatives. *SIAM J. Disc. Math.*, 5(3):422-427, 1992.

[15] D. Marx. Efficient approximation schemes for geometric problems? *Proc. 13th European Symposium on Algorithms (ESA'05)*, pp. 448-459, 2005.

[16] E. Papadopoulou and S.K. Dey. On the farthest line-segment Voronoi diagram. *Intl. J. Comp. Geom. and Appl.*, 23(6):443-460, 2013.

[17] J. Pruente. Minimum diameter color-spanning sets revisited. *Disc. Optim.*, 34:100550, 2019.

[18] R. Ravi, D. Rosenkrantz and G. Tayi. Heuristic and special case algorithms for dispersing problems. *Oper. Res.*, 42:299-310, 1994.

[19] M. Sydow. Approximation guarantees for max sum and max min facility dispersion with parameterised triangle inequality and applications in result diversification. *Mathematica Applicanda*, 42:241-257, 2014.

[20] D.W. Wang and Y-S Kuo. A study on two geometric location problems. *Infor. Proc. Lett.*, 28:281-286, 1988.