PAPER *Special Section on Discrete Mathematics and Its Applications*

Max-Min 3-Dispersion Problems

Takashi HORIYAMA[†], **Shin-ichi NAKANO**[††a)], **Toshiki SAITOH**[†††], ***Members***, **Koki SUETSUGU**[††††], ***Nonmember***, **Akira SUZUKI**[†††††], **Ryuhei UEHARA**[††††††], **Takeaki UNO**[††††], ***and*** **Kunihiro WASA**[*], ***Members***

SUMMARY Given a set P of n points on which facilities can be placed and an integer k, we want to place k facilities on some points so that the minimum distance between facilities is maximized. The problem is called the k-dispersion problem. In this paper, we consider the 3-dispersion problem when P is a set of points on a plane (2-dimensional space). Note that the 2-dispersion problem corresponds to the diameter problem. We give an $O(n)$ time algorithm to solve the 3-dispersion problem in the L_∞ metric, and an $O(n)$ time algorithm to solve the 3-dispersion problem in the L_1 metric. Also, we give an $O(n^2 \log n)$ time algorithm to solve the 3-dispersion problem in the L_2 metric.
key words: *algorithms, dispersion problem, facility location*

1. Introduction

The facility location problem and many of its variants have been studied. See good textbooks [10], [11]. Typically, given a set of points on which facilities can be placed and an integer k, we want to place k facilities on some points so that a designated function on distance is minimized. By contrast in the *dispersion problem*, we want to place facilities so that a designated function on distance is maximized.

The intuition of the problem is as follows. Assume that we are planning to open several chain stores in a city. We wish to locate the stores mutually far away from each other to avoid self-competition. We want to find k points so that the minimum distance among them is maximized. For more applications, including *result diversification*, see [8], [19], [20].

Now we formally define the *max-min k-dispersion problem*. Given a set P of n possible points, a distance function d for each pair of points (we assume that d is a symmetric nonnegative function satisfying $d(p, p) = 0$ for all $p \in P$), and an integer k with $k \ll n$, we want to find a subset $S \subset P$ with $|S| = k$ such that the cost $cost(S) = \min_{\{u,v\}\subset S}\{d(u, v)\}$ is maximized. Such a set S is called a k-dispersion of P. Note that a 2-dispersion of P on a plane (2-dimensional space) with L_2 metric corresponds to the diameter of P, thus a k-dispersion is a generalization of the diameter, which is one of basic concept in geometry. (Here the diameter of P is the maximum distance between two points in P, and one can compute it in $O(n \log n)$ time [16].) This is the max-min version of the k-dispersion problem [19], [21]. For the max-sum version, see [5]–[9], [13], [17], [19], and for a variety of related problems, see [5], [9]. The max-min k-dispersion problem is NP-hard even when the triangle inequality is satisfied [12], [21]. An exponential-time exact algorithm for the max-min k-dispersion problem is known [2]. The running time is $O(n^{\omega k/3} \log n)$, where $\omega < 2.373$ is the matrix multiplication exponent.

If P is a set of n points on a line (1-dimensional space) and the order of points in P on the line is given, the k-dispersion problem can be solved in $O(kn)$ time [21]. The running time was improved to $O(n \log \log n)$ [3] by the sorted matrix search method [14]. See a good survey for the sorted matrix search method in [1, Sect. 3.3]. Later it was improved to $O(n)$ [2] by a reduction to the path partitioning problem [14]. Even if the order of points in P on the line is not given, the k-dispersion problem can be solved in $O(n)$ time [4] if k is a constant.

If P is a set of n points on a plane (2-dimensional space) the k-dispersion problem is NP-hard [21].

Ravi et al. [19] proved that the max-min k-dispersion problem cannot be approximated within any constant factor in polynomial time, and cannot be approximated within a factor of two in polynomial time when the distance satisfies the triangle inequality, unless P = NP. They also gave a polynomial-time algorithm with approximation ratio two when the triangle inequality is satisfied.

In this paper, we consider the k-dispersion problem only for the case $k = 3$, namely, the max-min 3-dispersion problem.

We first study the case where P is a set of points on a plane (2-dimensional space) and d is the L_∞ metric. We give an algorithm to compute the 3-dispersion of P in $O(n)$ time.

Next we study the case where d is the L_1 metric. We show that a similar algorithm can compute the 3-dispersion of P in $O(n)$ time.

Manuscript received August 24, 2020.
Manuscript revised January 3, 2021.
Manuscript publicized March 19, 2021.
[†]The author is with Hokkaido University, Sapporo-shi, 060-0814 Japan.
[††]The author is with Gunma University, Kiryu-shi, 376-8515 Japan.
[†††]The author is with Kyushu Institute of Technology, Iizuka-shi, 820-8502 Japan.
[††††]The authors are with National Institute of Informatics, Tokyo, 101-8430 Japan.
[†††††]The author is with Tohoku University, Sendai-shi, 980-8579 Japan.
[††††††]The author is with JAIST, Nomi-shi, 923-1292 Japan.
[*]The author is with Toyohashi University of Technology, Toyohashi-shi, 441-8580 Japan.
a) E-mail: nakano@cs.gunma-u.ac.jp
DOI: 10.1587/transfun.2020DMP0003

Finally we study the case where d is the L_2 metric. We give an algorithm to compute the 3-dispersion of P in $O(n^2 \log n)$ time. By slightly improving the algorithm, we can also compute the 3-dispersion of P in D-dimension space in $O(Dn^2 + Tn \log n)$ time where T is the time to compute the diameter of n points in D-dimensional space.

In this paper we use the following notations and terms. P is a set of points in the space. For $p \in P$, we write their $x-, y-, z-$ coordinates as $x(p), y(p)$, and $z(p)$. We say $p_\ell \in P$ is located *left* of p_r if $x(p_\ell) \le x(p_r)$, and p' is *a leftmost point* in P if $x(p') \le x(p)$ for every $p \in P$. Similarly, we define *right*, *rightmost*, *highest* and *lowest*. A point $p' \in P$ is a farthest point from $p'' \in P$ if $d(p', p'') \ge d(p, p'')$ for every $p \in P$.

The remainder of this paper is organized as follows. Section 2 gives an $O(n)$ time algorithm to solve the 3-dispersion problem if d is the L_∞ metric. Section 3 gives an $O(n)$ time algorithm to solve the 3-dispersion problem if d is the L_1 metric. Section 4 gives an $O(n^2 \log n)$ time algorithm to solve the 3-dispersion problem if d is the L_2 metric. Finally Sect. 5 is a conclusion.

The preliminary version of the paper is appeared in [15].

2. 3-Dispersion in L_∞ Metric

In this section, we give an $O(n)$ time algorithm to solve the 3-dispersion problem if P is a set of n points on a plane (2-dimensional space) and d is the L_∞ metric.

Let $P = \{p_1, p_2, \cdots, p_n\}$ and assume that $x(p_1) \ge x(p_2) \ge \cdots \ge x(p_n)$. Let $S = \{p_a, p_b, p_c\}$ be a 3-dispersion of P. We say that a pair (p_u, p_v) in P is *type-H* if $d(p_u, p_v) = |x(p_u) - x(p_v)|$, and *type-V* otherwise. Let $E = \{(p_a, p_b), (p_b, p_c), (p_c, p_a)\}$. We have the following four cases for S.

Case 1: All three pairs in E are *type-H*.
Case 2: Two pairs in E are *type-H* and one pair in E is *type-V*.
Case 3: Two pairs in E are *type-V* and one pair in E is *type-H*.
Case 4: All three pairs in E are *type-V*.

Our algorithm computes three points having the maximum cost for each case, and then chooses the maximum one among those four candidates as a 3-dispersion. Now we explain how to compute a 3-dispersion S restricted for each case.

Case 1: All three pairs in E are *type-H*.

The solution consists of (1) the leftmost point p_n in P, (2) the rightmost point p_1 in P, and (3) the point p_m in P which has x-coordinate closest to $(x(p_1) + x(p_n))/2$.

One can find p_n and p_1 in $O(n)$ time, and then find p_m in $O(n)$ time.

Thus we can compute a 3-dispersion $S = \{p_1, p_m, p_n\}$ of P in $O(n)$ time if S is in Case 1.

Case 2: Two pairs in E are *type-H* and one pair in E is *type-V*.

Let $S = \{p_a, p_b, p_c\}$ be a 3-dispersion with $x(p_a) \le x(p_b) \le x(p_c)$, and assume that S satisfied the condition of Case 2.

Now either (p_a, p_b) or (p_b, p_c) is *type-V*. Otherwise, if (p_a, p_c) is *type-V*, then either (p_a, p_b) or (p_b, p_c) is also *type-V*, a contradiction. Assume (p_b, p_c) is *type-V*. The other case is symmetrical. Let $P_i = \{p_1, p_2, \cdots, p_i\}$ be the subset of P consisting of the rightmost i points in P. We have the following two lemmas.

Lemma 1. *There is a 3-dispersion $S = \{p_a, p_b, p_c\}$ such that p_a is the leftmost point p_n in P.*

Proof. Assume for a contradiction that $p_a \ne p_n$. Let S' be $\{p_n, p_b, p_c\}$, which is derived from S by replacing p_a with the leftmost points p_n in P. Now (p_a, p_b) and (p_a, p_c) are *type-H*, so $cost(S') \ge cost(S)$ holds. If $cost(S') = cost(S)$, then the claim is satisfied. If $cost(S') > cost(S)$, then S is not a 3-dispersion, a contradiction. (Note that S' may not be in Case 2.) □

Lemma 2. *There is a 3-dispersion $S = \{p_a, p_b, p_c\}$ satisfying the following conditions. Let $p_i = p_b$. If $y(p_b) \le y(p_c)$, then p_b is a lowest point in P_i, and p_c is a highest point in P_i. If $y(p_b) > y(p_c)$, then p_b is a highest point in P_i, and p_c is a lowest point in P_i.*

Proof. Assume for a contradiction that $y(p_b) \le y(p_c)$ but p_b is not the lowest point p_ℓ in P_i. Then let S' be $\{p_a, p_\ell, p_c\}$, which is derived from S by replacing p_b with p_ℓ. Since (p_b, p_c) is *type-V*, now $cost(S') \ge cost(S)$ holds. If $cost(S') = cost(S)$, then the claim is satisfied for some $p_{b'}$ with $b' < b$. If $cost(S') > cost(S)$, then S is not a 3-dispersion, a contradiction. (Note that S' may not be in Case 2.)

For the other case, that is, $y(p_b) > y(p_c)$ but p_b is not the highest point in P_i, the proof is analogous and is omitted. □

Let h_i be the highest point in P_i and ℓ_i the lowest point in P_i.

Suppose that $\{p_a, p_b, p_c\}$ is a 3-dispersion of P such that $x(p_a) \le x(p_b) \le x(p_c)$, (p_b, p_c) is *type-V*, and $\{p_a, p_b, p_c\}$ is in Case 2. Then we can assume $p_a = p_n$ (by Lemma 1), and $\{p_b, p_c\} = \{h_i, \ell_i\}$ for $p_b = p_i$ (by Lemma 2). See Fig. 1.

We first compute $\min\{d(p_a, p_i), d(h_i, \ell_i)\}$ for each i, and then we choose the i satisfying $p_i \in \{h_i, \ell_i\}$. Now

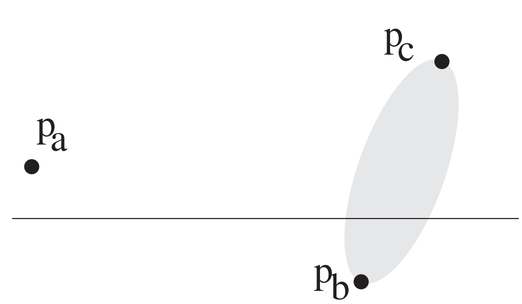


Fig. 1 An illustration for Case 2.

$\{p_a = p_n, h_i, \ell_i\}$ corresponds to a 3-dispersion $\{p_a, p_b, p_c\}$ of P, since $d(p_a, p_i) = \min\{d(p_a, p_b), d(p_a, p_c)\}$ and $d(h_i, \ell_i) = d(p_b, p_c)$.

More efficiently, we can compute a 3-dispersion $\{p_a, p_b, p_c\}$ by binary search as follows.

First we sort P by their x-coordinates in $O(n \log n)$ time.

By scanning P from right to left, we can compute the highest point h_i and the lowest point ℓ_i in each P_i with $1 \le i \le n$ in $O(n)$ time in total. We also compute $d(p_n, p_i)$ for each i with $0 < i < n$ in $O(n)$ time in total. Now we compute $\max_i \min\{d(p_a, p_i), d(h_i, \ell_i)\}$. Clearly, $d(p_a = p_n, p_i)$ is monotonically decreasing with respect to i, and $d(h_i, \ell_i)$ is monotonically increasing with respect to i. Then, by binary search, we can compute the optimal i with $S = \{p_a = p_n, h_i, \ell_i\}$ having the maximum cost $\min\{d(p_n, p_i), d(h_i, \ell_i)\}$ in $\log n$ stages. Each stage of the binary search requires $O(1)$ time.

Thus we can compute a 3-dispersion S in $O(n \log n)$ time in total if S is in Case 2.

Case 3: Two pairs in E are *type-V* and one pair in E is *type-H*.

Similar to Case 2, swap x-axis and y-axis.

Case 4: All three pairs in E are *type-V*.

Similar to Case 1, swap x-axis and y-axis.

Based on the above explanation we can design an algorithm to solve the 3-dispersion problem, and we have the following lemma.

Lemma 3. *If P is a set of n points on a plane and d is the L_∞ metric, then one can solve the max-min 3-dispersion problem in $O(n \log n)$ time.*

We can improve the running time to $O(n)$ by removing the sort in Cases 2 and 3. The binary search proceeds as follows. In the j-th stage, we have a set I of points having consecutive x-coordinates containing optimal p_i and $|I| = n/2^{j-1}$. We also maintain the highest point h_R and the lowest point ℓ_R in the set R (we do not maintain R itself) of points locating right of I. (For the first stage, these two points are not defined because there is no point locating right of I.) We find the median $p_{j'}$ in I in $O(n/2^{j-1})$ time by the linear-time median-finding algorithm. Then find the highest point h_j and the lowest point ℓ_j in the right half points I_R of I consisting of $n/2^j$ points in $O(n/2^{j-1})$ time. By the two points h_j and ℓ_j in I_R and h_R, ℓ_R in R, we can compute the highest point and the lowest point in $P_{j'} = I_R \cup R$ in constant time. Depending on whether $d(p_a, p_{j'}) < d(h_{j'}, \ell_{j'})$ or not we proceed to the next stage with new parameters I, h_R and ℓ_R. (If $d(p_a, p_{j'}) < d(h_{j'}, \ell_{j'})$, then h_R and ℓ_R remain as it was, otherwise set $h_R = \max\{h_R, h_j\}$ and $\ell_R = \min\{\ell_R, \ell_j\}$.)

We now have the following theorem.

Theorem 4. *If P is a set of n points on a plane and d is the L_∞ metric, then one can solve the max-min 3-dispersion problem in $O(n)$ time.*

We cannot simply generalize the algorithm to 3-dimensional space since there is an example in which every 3-dispersion consists of points with no extreme coordinate value. Let $P = \{(1, 0, 0), (0, 1, 0), (0, 0, 1), (-1, 0, 0), (0, -1, 0), (0, 0, -1), (0.9, -0.9, 0), (0, 0.9, -0.9), (-0.9, 0, 0.9)\}$ then the 3-dispersion of P is $\{(0.9, -0.9, 0), (0, 0.9, -0.9), (-0.9, 0, 0.9)\}$, and none of which has an extreme coordinate value.

We now give an $O(n \log n)$ time algorithm to solve the 3-dispersion problem in 3-dimensional space.

We say a pair (p_u, p_v) in P is *type-X* if $d(p_u, p_v) = |x(p_u) - x(p_v)|$, *type-Y* if $d(p_u, p_v) = |y(p_u) - y(p_v)|$ and $|y(p_u) - y(p_v)| > |x(p_u) - x(p_v)|$, *type-Z* otherwise.

Let $P = \{p_1, p_2, \cdots, p_n\}$ be the set of points in 3-dimensional space, and let $S = \{p_a, p_b, p_c\}$ be the 3-dispersion of P.

Let $E = \{(p_a, p_b), (p_b, p_c), (p_c, p_a)\}$. We have 3^3 cases for S since each of $(p_a, p_b), (p_b, p_c), (p_c, p_a)$ is either *type-X*, *type-Y*, or *type-Z*.

Case 1: (p_a, p_b) is *type-X*, (p_a, p_c) is *type-Y*, and (p_b, p_c) is *type-Z*.

We have eight subcases for S depending on the order of p_a, p_b on x-coordinate, p_a, p_c on y-coordinate and p_b, p_c on z-coordinate.

Case 1(a): $x(p_a) \le x(p_b)$, $y(p_a) \le y(p_c)$, and $z(p_b) \le z(p_c)$. (Other cases are analogous and omitted.)

Fix p_a. We want to compute three points $\{p_a, p_b, p_c\}$ with the maximum value d^* satisfying $\min\{x(p_b) - x(p_a), y(p_c) - y(p_a)\} = d^*$, and $z(p_c) - z(p_b) \ge d^*$. Each candidate value for d^* is the distance between p_a and a point in P, so the number of such values is at most n. By binary search, we are going to find the maximum such value d^*. We now need some definitions.

We sort the points with their x-coordinates in $O(n \log n)$ time. Similarly, sort the points with their y-coordinates. Assume that $P = \{p_1, p_2, \cdots, p_n\} = \{p_1', p_2', \cdots, p_n'\}$, $x(p_1) \ge x(p_2) \ge \cdots \ge x(p_n)$, and $y(p_1') \ge y(p_2') \ge \cdots \ge y(p_n')$.

Let $B_i = \{p | x(p) \ge x(p_i)\}$ and $C_j = \{p | y(p) \ge y(p_j')\}$. We compute a table T_B as a preprocessing step so that $T_B(i) = \min\{z(p) | p \in B_i\}$. Similarly, we compute a table T_C as a preprocessing step so that $T_C(j) = \max\{z(p) | p \in C_j\}$. We need $O(n)$ time for these tables.

We maintain the set Pb of candidates for p_b. Initially, we set $Pb = \{p | x(p) > x(p_a)\}$. Similarly, we maintain the set Pc of candidates for p_c. Initially, we set $Pc = \{p | y(p) > y(p_a)\}$.

The binary search proceeds the following way.

Let $p_{b'}$ be the point in Pb having the median of x-coordinate, and $p_{c'}'$ be the point in Pc having the median of y-coordinate. Let $d' = \min\{x(p_{b'}) - x(p_a), y(p_{c'}') - y(p_a)\}$. Assume $d' = x(p_{b'}) - x(p_a)$. (The other case is similar.) If $T_C(c') - T_B(b') \ge d'$, then there exists (p_a, p_b, p_c) satisfying (1) $\min\{x(p_b) - x(p_a), y(p_c) - y(p_a)\} \ge d'$, and (2) $z(p_c) - z(p_b) \ge d'$. Now $b' \ge b$ holds for any 3-dispersion $\{p_a, p_b, p_c\}$. In this case, we can halve the size of the candidate sets Pb. If $T_C(c') - T_B(b') < d'$, then there is no $\{p_a, p_b, p_c\}$ with cost d' or more, so $b' < b$ holds for any 3-

dispersion $\{p_a, p_b, p_c\}$. In this case, we can again halve the size of the candidate set Pb.

If either $|Pb|$ or $|Pc|$ is 1, we can proceed the binary search only for the remaining set having 2 or more points and easily compute a 3-dispersion $\{p_a, p_b, p_c\}$ in $O(\log n)$ time.

Thus by binary search, we can find the maximum d' in at most $\log 2n$ stages. In each stage, we can compute $T_C(c') - T_B(b')$ in $O(1)$ time since we built two tables T_B and T_C, and we can compute the medians in $O(1)$ time since we sorted the points with x-coordinates and y-coordinates, respectively.

We need to compute above for each possible p_a. Thus the running time for Case 1 is $O(n \log n)$ in total.

For other cases, if all three types appear, it is analogous to above, otherwise if at most two types appear, we can solve it as the problem on a plane with the algorithm in Theorem 1. Since the number of cases is a constant, the total running time is $O(n \log n)$.

3. 3-Dispersion in L_1

In this section, we give an $O(n)$ time algorithm to solve the 3-dispersion problem when P is a set of n points on a plane (2-dimensional space) and d is the L_1 metric.

We consider four coordinate systems each of which is derived from the original coordinate system by rotating 45 (See Fig. 2), 135, 225 or 315 degrees clockwise around the origin respectively. Note that a farthest point in P from a point has extreme x-coordinate in one of those four coordinate systems. We can also observe that there is a 3-dispersion of P containing a point having extreme x-coordinate in one of those four coordinate systems. Note that each coordinate system has two extreme points for each coordinate. We only explain 45 degree case with the point having the minimum x-coordinate. Other cases are similar.

Let x' and y' be the coordinates of the rotated coordinate system. Let $P = \{p_1, p_2, \cdots, p_n\}$ and assume $x'(p_1) \geq x'(p_2) \geq \cdots \geq x'(p_n)$. Let $S = \{p_a, p_b, p_c\}$ be the 3-dispersion of P with $x'(p_a) \leq x'(p_b) \leq x'(p_c)$ and $p_a = p_n$ is the point having the minimum x'-coordinate.

We say two points (p_u, p_v) with $x'(p_u) \leq x'(p_v)$ in P are *type-U* (upward) if $y(p_u) \leq y(p_v)$, and *type-D* (downward) otherwise.

We compute the optimal i, which is the i with the maximum $\min\{d(p_a, p_i), diam(P_i)\}$, where p_i is the i-th farthest point from p_a in the L_1 metric and $diam(P_i)$ is the diameter of P_i where $P_i = \{p_1, p_2, \cdots p_i\}$ is the subset of P consisting of the i farthest points from $p_n = p_a$ in P. Let p_b and p_c be the points corresponding to the diameter of P_i. See Fig. 2. We can assume that if p_b and p_c are *type-U*, then p_b is the point with the minimum $y'(p_b)$ in P_i, and p_c is the point with the maximum $y'(p_c)$ in P_i. Note that any possible point for p_b is on the line parallel to x'-axis containing points farthest from p_c, and any possible point for p_c is on the line parallel to x'-axis containing points farthest from p_b. If p_b and p_c are *type-D*, then p_b is the point p_i with the minimum $x'(p_b)$ in P_i, and p_c is the point p_1 with the maximum $x'(p_c)$ in P_i.

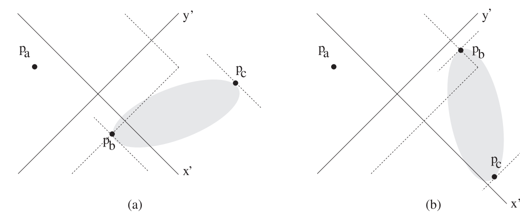


Fig. 2 Illustrations for the diameter of P_i in the L_1 metric with (a) *type-U* and (b) *type-D*.

Similar to the L_∞ metric case, we can compute a max-min 3-dispersion of P in $O(n)$ time, by binary search with the linear-time median-finding algorithm.

Now we have the following theorem.

Theorem 5. *If P is a set of n points on a plane and d is the L_1 metric, then one can solve the max-min 3-dispersion problem in $O(n)$ time.*

4. 3-Dispersion in L_2 Metric

In this section, we design an $O(n^2 \log n)$ time algorithm to solve the 3-dispersion problem when P is a set of n points on a plane (2-dimensional space) and d is the L_2 metric.

Let $S = \{p_a, p_b, p_c\}$ be a 3-dispersion of P, and assume that $d(p_b, p_c)$ is the shortest one among $\{d(p_a, p_b), d(p_b, p_c), d(p_c, p_a)\}$, $d(p_a, p_b) \leq d(p_a, p_c)$, and p_b is the i-th farthest point from p_a in P. Let $P_i = \{p_1, p_2, \cdots, p_i\}$ be the subset of P consisting of the i farthest points from p_a (i.e., $p_i = p_b$ and $c \in \{1, 2, \cdots, i-1\}$). We have the following lemma. Let $diam(P)$ be the diameter of P.

Lemma 6. $d(p_b, p_c) = diam(P_i)$.

Proof. Assume $d(p_b, p_c) \neq diam(P_i)$. Now there are $p_{b'}, p_{c'} \in P_i$ with $diam(P_i) = d(p_{b'}, p_{c'})$. Let $S' = \{p_a, p_{b'}, p_{c'}\}$. Now $d(p_a, p_b) \leq d(p_a, p_{b'})$, $d(p_a, p_b) \leq d(p_a, p_{c'})$, and $d(p_b, p_c) < d(p_{b'}, p_{c'})$ hold. Thus $cost(S) < cost(S')$, a contradiction. □

Thus if, for each p_a, we compute the optimal i which maximizes $\min\{d(p_a, p_i), diam(P_i)\}$, and choose the maximum one, it corresponds to a 3-dispersion of P.

For a fixed p_a, we can compute the optimal i^* with the maximum $\min\{d(p_a, p_{i^*}), diam(P_{i^*})\}$ by binary search as follows.

Clearly, $d(p_a, p_i)$ is monotonically decreasing with respect to i, and $diam(P_i)$ is monotonically increasing with respect to i.

First, we sort the points in P by the distance from p_a. Then, we are going to find the optimal i^*.

In the j-th stage, we have (1) a set I of $n/2^{j-1}$ points having consecutive distances from p_a containing p_{i^*}, and (2) the convex hull C_F of points having distance from p_a more than the distances in I. We first compute the median point

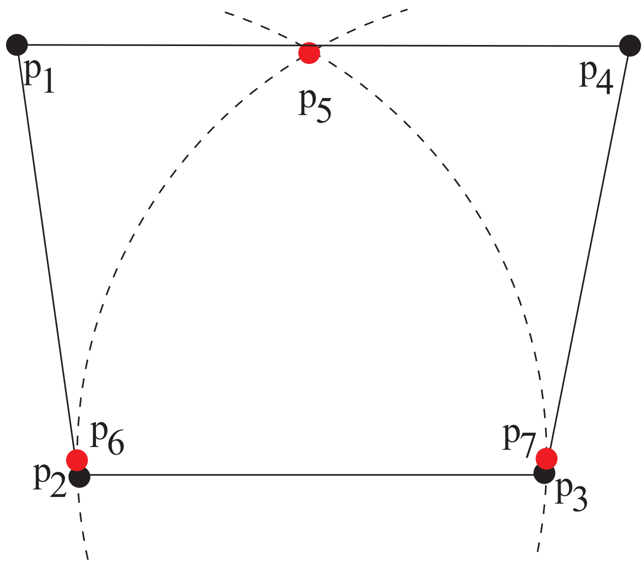


Fig. 3 A 3-dispersion may contain no corner points of the convex hull of P.

p_i in I with the linear-time median finding algorithm, and check whether $d(p_a, p_i) \leq diam(P_i)$ or not. For the check, we need to compute $diam(P_i)$. We compute the convex hull C_j of P_i by constructing the convex hull of $n/2^j$ points in I having distance from p_a more than or equal to $d(p_a, p_i)$ in $O((n/2^j) \log n)$ time (using the $O(n \log n)$ time algorithm in [16]), and then merging it with the convex hull C_F in $O(n)$ time. Using C_j, we can compute $diam(P_i)$ in $O(n)$ time. Depending on the result of $d(p_a, p_i) \leq diam(P_i)$, we proceed to the next stage with new parameters I and C_F.

Since the number of stages is at most $\log n$, the total running time for a fixed p_a is $O(n \log n)$.

We have the following theorem.

Theorem 7. *If P is a set of n points on a plane and d is the L_2 metric, then one can solve the max-min 3-dispersion problem in $O(n^2 \log n)$ time.*

If any P has a 3-dispersion with at least one point on the corner points of the convex hull of P, then we can check p_a only for the corner points of the convex hull of P. However there is a counterexample. In Fig. 3, the dotted circles have centers at p_6 and p_7 with radius $d(p_5, p_6) = d(p_6, p_7) = d(p_7, p_5)$, and $d(p_6, p_7) > d(p_2, p_7) = d(p_3, p_6) > d(p_2, p_3)$ holds. The corner points of the convex hull are $\{p_1, p_2, p_3, p_4\}$, and the points in the 3-dispersion are $\{p_5, p_6, p_7\}$. All points in this 3-dispersion are located in the strict incide of the convex hull.

By generalizing the algorithm to D-dimensional space, we have the following theorem.

Theorem 8. *If P is a set of n points in D-dimensional space and d is the L_2 metric, then one can solve the max-min 3-dispersion problem in $O(Dn^2 + Tn \log n)$ time, where T is the time to compute the diameter of n points in D-dimensional space.*

Note that for a fixed p_a, we need to compute medians in $O(Dn) + O(Dn/2) + O(Dn/2^2) + \cdots = O(Dn)$ time.

One can compute the diameter of n points in 3-dimensional space in $O(n \log n)$ time [18]. Thus we can compute a 3-dispersion of n points in 3-dimensional space in $O(n^2 \log^2 n)$ time, which is faster than the $O(n^{\omega k/3} \log n)$ time algorithm [2] for $k = 3$. (Here ω is the smallest value for which there is a known $O(n^{\omega})$ time matrix multiplication algorithm.) The diameter of n points in dimension D can be computed in time $O(n^{2-a(k)}(\log n)^{1-a(k)})$, where $a(k) = 2^{-(k+1)}$ [22]. Therefore we can compute the 3-dispersion of n points in d dimensional space in $o(n^3)$ time for any D.

5. Conclusion

In this paper, we designed some algorithms to solve the 3-dispersion problem for a set of points on a plane. We have designed $O(n)$ time algorithms to solve the 3-dispersion problem when d is the L_∞ metric or the L_1 metric. Also, we designed an $O(n^2 \log n)$ time algorithm to solve the 3-dispersion problem when d is the L_2 metric.

There is a linear time reduction from the diameter problem to the 3-dispersion problem as follows. Given P, we append a dummy point p' so that it is far enough from P. Now a 3-dispersion of $P \cup \{p'\}$ always contains p' and the other two points correspond to the diameter of P. It is known that any algorithm to solve the diameter problem requires $\Omega(n \log n)$ time [16]. Thus any algorithm to solve the 3-dispersion problem requires $\Omega(n \log n)$ time. Therefore there is a chance to either design a faster algorithm to solve the 3-dispersion problem with the L_2 metric, or show a greater lower bound.

For a set P of points in a metric space, we can compute the 3-dispersion of P as follows. By replacing $(+, \cdot)$ to $(\max, \min)$ in the matrix multiplication algorithm, we can compute $\max_c\{\min\{d(p_a, p_c), d(p_b, p_c)\}\}$ for each $p_a, p_b \in P$ in $O(n^{\omega})$ time. Therefore we can compute $\min\{d(p_a, p_b), \max_c\{\min\{d(p_a, p_c), d(p_b, p_c)\}\}\}$ for each $p_a, p_b \in P$ in $O(n^{\omega})$ time and choose the maximum one among them as a 3-dispersion. Thus we can compute a 3-dispersion of n points in a metric space in $O(n^{\omega})$ time, where $\omega < 2.373$.

Acknowledgments

Shin-ichi Nakano was supported by JST CREST Grant Number JPMJCR1402. Ryuhei Uehara was supported by JSPS KAKENHI Grant Number 18H04091 and 20K20311. Takeaki Uno was supported by JST CREST Grant Number JPMJCR1401. Kunihiro Wasa was supported by JSPS KAKENHI Grant Number 19K20350 and JST CREST Grant Number JPMJCR1401.

References

[1] P. Agarwal and M. Sharir, "Efficient algorithms for geometric optimization," ACM Comput. Surv., vol.30, no.4, pp.412–458, 1998.

[2] T. Akagi, T. Araki, T. Horiyama, S. Nakano, Y. Okamoto, Y. Otachi, T. Saitoh, R. Uehara, T. Uno, and K. Wasa, "Exact algorithms for the max-min dispersion problem," Proc. FAW 2018, LNCS, vol.10823, pp.263–272, 2018.
[3] T. Akagi and S. Nakano, "Dispersion on the line," IPSJ SIG Technical Reports, 2016-AL-158-3, 2016.
[4] T. Araki and S. Nakano, "The max-min dispersion on a line," Proc. COCOA 2018, LNCS, vol.11346, pp.672–678, 2018.
[5] C. Baur and S.P. Fekete, "Approximation of geometric dispersion problems," Proc. APPROX 1998, pp.63–75, 1998.
[6] B. Birnbaum and K.J. Goldman, "An improved analysis for a greedy remote-clique algorithm using factor-revealing LPs," Algorithmica, vol.50, pp.42–59, 2009.
[7] A. Cevallos, F. Eisenbrand, and R. Zenklusen, "Max-sum diversity via convex programming," Proc. SoCG 2016, pp.26:1–26:14, 2016.
[8] A. Cevallos, F. Eisenbrand, and R. Zenklusen, "Local search for max-sum diversification," Proc. SODA 2017, pp.130–142, 2017.
[9] B. Chandra and M.M. Halldorsson, "Approximation algorithms for dispersion problems," J. Algorithms, vol.38, no.2, pp.438–465, 2001.
[10] Z. Drezner, Facility Location: A Survey of Applications and Methods, Springer, 1995.
[11] Z. Drezner and H.W. Hamacher, Facility Location: Applications and Theory, Springer, 2004.
[12] E. Erkut, "The discrete p-dispersion problem," Eur. J. Oper. Res., vol.46, no.1, pp.48–60, 1990.
[13] S.P. Fekete and H. Meijer, "Maximum dispersion and geometric maximum weight cliques," Algorithmica, vol.38, pp.501–511, 2004.
[14] G. Frederickson, "Optimal algorithms for tree partitioning," Proc. SODA 1991, pp.168–177, 1991.
[15] T. Horiyama, S. Nakano, T. Saitoh, K. Suetsugu, A. Suzuki, R. Uehara, T. Uno, and K. Wasa, "Max-min 3-dispersion problems," Proc. COCOON 2019, LNCS, vol.11653, pp.291–300, 2019.
[16] F.P. Preparata and M.I. Shamos, Computational Geometry: An Introduction. Springer-Verlag, 1985.
[17] S. Rubinstein R. Hassin, and A. Tamir, "Approximation algorithms for maximum dispersion," Oper. Res. Lett., vol.21, no.3, pp.133–137, 1997.
[18] E.A. Ramos, "Deterministic algorithms for 3-D diameter and some 2-D lower envelopes," Proc. Symposium on Computational Geometry 2000, pp.290–299, 2000.
[19] S.S. Ravi, D.J. Rosenkrantz, and G.K. Tayi, "Heuristic and special case algorithms for dispersion problems," Oper. Res., vol.42, no.2, pp.299–310, 1994.
[20] M. Sydow, "Approximation guarantees for max sum and max min facility dispersion with parameterised triangle inequality and applications in result diversification," Mathematica Applicanda, vol.42, no.2, pp.241–257, 2014.
[21] D.W. Wang and Y.-S. Kuo, "A study on two geometric location problems," Inform. Process. Lett., vol.28, no.6, pp.281–286, 1988.
[22] A.C. Yao, "On constructing minimum spanning trees in k-dimensional spaces and related problems," SIAM J. Comput., vol.11, no.4, pp.721–736, 1982.

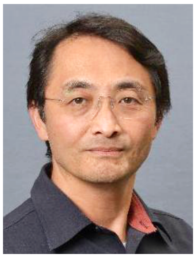
Takashi Horiyama received the B.E. and M.E. degrees in information science and Ph.D. in informatics from Kyoto University, Kyoto, Japan in 1995, 1997 and 2004, respectively. He was a research associate at Nara Institute of Science and Technology from 1999, a research associate atKyotoUniversity from 2002, and an associate professor at Saitama University from 2007. Since 2019, he is a professor at Hokkaido University. His current interests include computational geometry and algorithm design.

Shin-ichi Nakano received his B.E. and M.E. degrees from Tohoku University, Sendai, Japan, in 1985 and 1987, respectively. In 1987 he joined Seiko Epson Corp. and in 1990 he joined Tohoku University. In 1992, he received Dr. Eng. degree from Tohoku University. Since 1999 he has been a faculty member of Department of Computer Science, Faculty of Engineering, Gunma University. His research interests are graph algorithms and graph theory. He is a member of IPSJ and ACM.

Toshiki Saitoh received B.S.E. degree from Shimane University in 2005, and M.S. and Ph.D. degrees (Information Science) from Japan Advanced Institute of Science and Technology in 2007 and 2010, respectively. He was a researcher at ERATO MINATO Discrete Structure Manipulation System Project (JST) during 2010–2012, and an assistant professor at Kobe University during 2012–2017. He also works at The University of British Columbia as a visiting assistant professor in 2014–2015. He is an associate professor at Kyushu Institute of Technology from 2017.

Koki Suetsugu was born in 1991. He received his bachelor's, master 's, and doctoral degrees from Kyoto University in 2014, 2016, and 2019 respectively. Since 2019, he has been a postdoctoral fellow in the Takeaki Uno laboratory, National Institute of Informatics, Japan. His research interests are combinatorial game theory, game informatics, and computational complexity.

Akira Suzuki received his B.E., M.S. and Ph.D. degrees from Tohoku University, Japan, in 2010, 2011 and 2013, respectively. He is currently associate professor at Graduate School of Information Sciences, Tohoku University. His research interests include combinatorial reconfiguration, computational complexity, graph algorithms and neural networks.



Ryuhei Uehara is a professor in School of Information Science, Japan Advanced Institute of Science and Technology (JAIST). He received B.E., M.E., and Ph.D. degrees from the University of Electro-Communications, Japan, in 1989, 1991, and 1998, respectively. He was a researcher in CANON Inc. during 1991–1993. In 1993, he joined Tokyo Woman's Christian University as an assistant professor. He was a lecturer during 1998–2001, and an associate professor during 2001–2004 at Komazawa University. He moved to JAIST in 2004. His research interests include computational complexity, algorithms and data structures, and graph algorithms. Especially, he is engrossed in computational origami, games and puzzles from the viewpoints of theoretical computer science. He is a member of IPSJ, IEICE, and EATCS, and he is the chair of Japan Chapter of EATCS.

Takeaki Uno received the Ph.D. degree (Doctor of Science) from Department of Systems Science, Tokyo Institute of Technology Japan, 1998. He was an assistant professor in Department of Industrial and Management Science in Tokyo Institute of Technology from 1998 to 2001, and have been an associate professor of National Institute of Informatics Japan, from 2001. His research topic is discrete algorithms, especially enumeration algorithms, algorithms on graph classes, and data mining algorithms. On the theoretical part, he studies low degree polynomial time algorithms, and hardness proofs. In the application area, he works on the paradigm of constructing practically efficient algorithms for large scale data that are data oriented and theoretically supported. In an international frequent pattern mining competition in 2004 he won the best implementation award. He got Young Scientists' Prize of The Commendation for Science and Technology by the Minister of Education, Culture, Sports, Science and Technology in Japan, 2010.

Kunihiro Wasa received B.S. in Engineering in 2011, M.S. in Computer Science in 2013, Ph.D. in Computer Science in 2016 from Hokkaido University, Sapporo Japan. During 2016–2020, he was a researcher at National Institute of Informatics, Tokyo, Japan. He is currently an assistant professor at Toyohashi University of Technology. His research interests include enumeration algorithms, reconfiguration problems, data mining, and the design and analysis of algorithms in these fields.