Variants of the dispersion problem

Toshihiro Akagi[1,a)] Tetsuya Araki[2,b)] Shin-ichi Nakano[1,c)]

Abstract:
The dispersion problem is a variant of the facility location problem, which has been extensively studied. In this paper we design some algorithms for some variants of the dispersion problem.
Given a set P of n points on the horizontal line and an integer k we wish to find a subset S of P such that $|S| = k$ and maximizing the cost $\min_{x \in S} cost(x)$, where $cost(x)$ is the half of the sum of distances to its left neighbour and right neighbour in S. (For the leftmost point in S its cost is the half of the distance to its right neighbour. Similar for the rightmost points.) The problem is called the LR-dispersion problem. In this paper we give a simple $O(kn^2 \log n)$ time algorithm to solve the LR-dispersion problem.
Also we give some algorithms to solve some variants of the dispersion problem.

1. Introduction

The facility location problem and many of its variants have been studied [5], [6]. A typical problem is to find a set of locations to place facilities with the designated cost minimized. By contrast, in this paper we consider the dispersion problem(or obnoxious facility location problem), which finds a set of locations with a certain objective function maximized.

Given a set P of n possible locations, and the distance d for each pair of locations, and an integer k with $k \leq n$, we wish to find a subset $S \subset P$ with $|S| = k$ such that some designated objective function is maximized [3], [4], [8], [9], [10], [11], [12].

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. So we wish to find k locations so that some objective function based on the distance is maximized. See more applications, including *result diversification*, in [9], [10], [11].

In one of basic cases the objective function to be maximized is the minimum distance between two points in S. Then papers [10], [12] show if P is a set of points on the plane then the problem is NP-hard, and if P is a set of points on the line then the problem can be solved in $O(\max\{n \log n, pn\})$ time by dynamic programming approach, and in $O(n \log \log n)$ time by the sorted matrix search method [7].

1 Department of Computer Science, Gunma University, Kiryu, 376–8515, Japan
2 National Institute of Informatics, Chiyoda, Tokyo 101–8430, Japan
a) akagi@nakano-lab.cs.gunma-u.ac.jp
b) araki@nii.ac.jp
c) nakano@cs.gunma-u.ac.jp

In this paper we define some variants of the dispersion problem. Let P be a set of n points on the horizontal line, and we wish to find a subset $S \subset P$ with $|S| = k$ maximizing the following cost $cost(S)$.

Let the cost of a point f in S be the sum of (1) the half of the distance to its immediate left neighbour point in S and (2) the half of the distance to its immediate right neighbour point in S. We denote the cost for f by $cost(f)$. Intuitively the cost of $f \in S$ corresponds to the length of the segment in which possible customers for f live. (We assume each customer go to the nearest point(facility) in S.) Especially for the leftmost point the cost is consisting of just (2), and for the rightmost point the cost is consisting of just (1). And the cost of S, denoted by $cost(S)$, is the minimum cost among the costs of the points in S, which is $\min_{f \in S}\{cost(f)\}$. We call the problem above the LR-dispersion problem.

In this paper we design an algorithm to solve the LR-dispersion problem in $O(kn^2 \log n)$ time by dynamic programming approach.

The remainder of this paper is organized as follows. Section 2 contains our first algorithm for the LR-dispersion problem. Section 3 gives our second algorithm for the LR-dispersion problem. In Section 4 and Section 5 we consider more variants of the dispersion problem. Finally Section 6 is a conclusion.

2. The first algorithm

In this section we design an algorithm to solve the LR-dispersion problem, based on *dynamic programming* approach. We consider the subproblem $P(h, i; k)$ defined below, and systematically solve them.

Let P_i be the subset of the points in P locating on the left of $i \in P$ (including i). Given $h \in P_i$ and an integer k, we wish to find a subset $S \subset P_i$ such that (1) $|S| = k$, (2) the rightmost two points in S is h and i, with $h < i$, and (3) maximizing $cost(S)$. We denote by $cost(h,i;k)$ the optimal cost of a solution of $P(h,i;k)$. This is the LR-dispersion problem with the rightmost two points in S are designated.

We can assume $k \geq 2$ since otherwise we cannot define the cost. We have the following lemma.

Lemma 1. *$P(h,i;k)$ has a solution S containing the leftmost and rightmost points in P_i.*

Proof. Assume otherwise. If the leftmost point 1 is not contained in S then remove the leftmost point in S from S and append 1 to S, and similarly if the rightmost point i is not contained in S then remove the rightmost point in S from S and append i to S. Those modification never decrease $cost(S)$, so resulting S is also a solution, and it contains the leftmost and rightmost points in P_i. □

Now we explain how to solve $P(h,i;k)$. We have three cases.

Case 1: (Base case) $k = 2$.

By the lemma above we only consider the case $h = 1$. Then the solution is $S = \{1, i\}$, and its cost is $cost(1,i;2) = d(1,i)/2$.

Case 2: (Inductive case) $k > 3$.

We can assume we have already computed the solutions of smaller problems $P(h',h;k-1)$ for $h' = k-1, k, ..., h-1$. By appending i to a solution of a smaller problem we can construct a solution of $P(h,i;k)$, as follows.

Note that in the solution of $P(h,i;k)$, $cost(i)$ is $d(h,i)/2$, and $cost(h)$, which is $d(h',i)/2$ for some h', is greater than $cost(i)$.

We need one more subproblem. Let $P'(h',h;k-1)$ be the problem to find a subset S of P_i such that (1) $|S| = k-1$, (2) the rightmost two points in S is h' and $h > h'$, and (3) maximizing the cost $cost'(h',h;k-1)$ defined by $\min_{x \in S-\{h\}}\{cost(x)\}$.

Appending i to the solution S' of problem $P(h',h;k-1)$ for some h' is a solution S of $P(h,i;k)$. We have two subcases.

Subcase 2a: $cost(h,i;k)$ is $cost(i)$.

Then $cost'(h',h;k-1)$ with some $h' < h$ is greater than or equal to $cost(i)$.

In the solution S $cost(h) > cost(i)$ holds since $cost(h) = d(h',i)/2$ and $cost(i) = d(h,i)/2$. In this subcase the minimum cost of the points in $S - \{h,i\}$, which is $cost'(h',h;k-1)$, is greater than $cost(i)$.

Thus if $cost'(h',h;k-1) > cost(i)$ for some $h' \geq k-2$ then this case occurs and $cost(h,i;k)$ is $cost(i)$.

Subcase 2b: $cost(h,i;k)$ is not $cost(i)$.

Then $cost(h,i;k)$ is $cost(x)$ for some $x \in S - \{h,i\}$, which is smaller than $cost(i)$. Note that since $cost(h) > cost(i)$ holds $cost(h)$ is not the minimum.

Then if $cost'(h',h;k-1) < cost(i)$ for all h' then this case occurs and $cost(h,i;k)$ is the maximum of $cost'(h',h;k-1)$ for $h' = k-2, k-1, \cdots, h-1$.

Case 3: (Inductive case) $k = 3$.

Similar to Case 2. However we only refer to the solution of sub problem $P'(h',h;2)$with $h' = 1$ by Lemma 1.

Thus $cost(h,i;k)$ is $\max_{h'=1,2,\cdots,h-1}\min\{cost'(h',h;k-1), d(h,i)/2\}$.

Note that $cost(h)$ in a solution of $P(h,i;k)$ is larger than $cost(h)$ in a solution of $P(h',h;k-1)$.

Thus $cost(h,i;k) = \max_{h'=1,2,\cdots,h-1}\{\min\{cost'(h',h;k-1), d(h,i)/2\}\}$ and we can compute it in $O(n)$ time. Since the number of subproblems $P(h,i;k)$ is at most kn^2 the total time to solve them is $O(kn^3)$.

Finally the cost of a solution of the given problem is $\max_{h=1,2,\cdots,n-1}\{cost(h,n;k)\}$ and we can compute it in $O(n)$ time.

The entire algorithm **find-LR-dispersion**(P,n,k) is shown below.

Algorithm 1 find-LR-dispersion(P,n,k)

```
% Compute P(1,i;2)      (Case k = 2)
for i = 2,3,···,n do
  cost(1,i;2) = d(1,i)/2
  cost'(1,i;2) = d(1,i)/2
end for
% Compute P(h,i;3)      (Case k = 3)
for i = 3,4,···,n do
  for h = 2,3,···,i−1 do
    cost(h,i;3) = min{cost'(1,h;2), d(h,i)/2}
    cost'(h,i;3) = cost'(1,h;2)
  end for
end for
% Compute P(h,i;k)      (Case k > 3)
for k' = 4,5,···,k do
  for i = k',k'+1,···,n do
    for h = k'−1,k',···,i−1 do
      cost(h,i;k') = 0
      cost'(h,i;k') = 0
      % Compute the maximum cost
      for h' = k'−2,k'−1,···,h−1 do
        if min{cost'(h',h;k'−1), d(h,i)/2} > cost(h,i;k')
        then
          cost(h,i;k') = min{cost'(h',h;k'−1), d(h,i)/2}
        end if
        if min{cost'(h',h;k'−1), d(h',i)/2} > cost'(h,i;k')
        then
          cost'(h,i;k') = min{cost'(h',h;k'−1), d(h',i)/2}
        end if
      end for
    end for
  end for
end for
% Compute the optimal cost
cost = 0
for h = k−1,k,···,n−1 do
  if cost(h,n;k) > cost then
    cost = cost(h,n;k)
  end if
end for
Output cost
```

Theorem 1. *One can solve the LR-dispersion problem in $O(kn^3)$ time.*

3. Second Algorithm

In this section we give a faster algorithm to solve the LR-dispersion problem.

In algorithm **find-LR-dispersion**(P, n, k) we compute $\min\{cost'(h', h; k-1), d(h, i)/2\}$ for each $h' = 1, 2, \cdots, h-1$ and find the minimum one.

We have the following lemma.

Lemma 2. *$cost'(h', h; k-1)$ is a non-decreasing function with respect to h'.*

Proof. Assume otherwise. Then for some h_L, h_R in P with $h_L < h_R$, $cost'(h_L, h; k-1) > cost'(h_R, h; k-1)$ holds. Note that $cost'(h', h; k-1)$ is $\min_{x \in S-\{h\}}\{cost(x)\}$.

Let S_L be the solution of $P(h_L, h; k-1)$ and S' be the set of points derived from S_L by removing h_L then appending h_R. Also let h_x be the left neighbour of h_L in S_L. Then $cost(h_L)$ in S_L is $d(h_x, h)/2$ and $cost(h_R)$ in S' is also $d(h_x, h)/2$. And $cost(h_x)$ in S_L is smaller than $cost(h_x)$ in S'. Thus $cost'(h_L, h; k-1) \leq \min_{x \in S'-\{h\}}\{cost(x)\} \leq cost'(h_R, h; k-1)$ holds. A contradiction. □

Thus $\max_{h'}\{\min\{cost'(h', h; k-1), d(h, i)/2\}\} = \min\{cost'(h'', h; k-1), d(h, i)/2\}$, where h'' is the left neighbour of h in P, and we can compute it in constant time.

By Lemma 2 $\min_{h'}\{cost'(h', h; k'-1), d(h', i)/2\}$ is a non-decreasing function with respect to h' up to some points, then it is a decreasing linear function with respect to h', so we can find the maximum one by binary search in $O(\log n)$ time.

We have the following theorem.

Theorem 2. *One can solve the LR-dispersion problem in $O(kn^2 \log n)$ time.*

4. PS2-dispersion

In this section we consider one more variant of the dispersion problem, then design an algorithm to solve the problem. First define another cost for a set S of points with $|S| \geq 3$.

Given a subset S of P, let the cost $cost(f)$ of a point f in S be the sum of (1) the distance to the nearest point of f in S and (2) the distance to the 2nd nearest point of f in S. Intuitively this cost models competition to the nearest two stores. Then the cost $cost(S)$ of S is the minimum cost among the costs of the points in S.

Given a set P of n points on the horizontal line and an integer k we wish to find a subset $S \subset P$ with $|S| = k$ maximizing $cost(S)$. The problem is called PS2-dispersion problem, here PS2 means partial sum of the two nearest points in S. Some experimental results (for more general problems) are known. See [9].

We consider the subproblem $PS2(h, i; k)$ defined below.

Let P_i be the subset of the points in P locating on the left of $i \in P$ (including i). Given $h \in P_i$ and an integer $k \geq 3$, we wish to find a subset $S \subset P_i$ such that (1) $|S| = k$ and (2) the rightmost two points in S is h and i, with $h < i$, (3) maximizing $cost(S)$. We denote by $cost(h, i; k)$ the optimal cost of a solution of $PS2(h, i; k)$. This is the PS2-dispersion problem with the rightmost two points in S are designated.

Similar to Lemma 1 $PS2(h, i; k)$ has a solution S containing the leftmost and rightmost points in P_i. Thus we can assume $1, i \in S$.

We have the following lemma.

Lemma 3. *Let S be a solution of $PS2(h, i; k)$, and h, i the rightmost two points in S. Then the following (a)–(c) holds. (a) The two nearest points of $i \in S$ are located on the left of i, (b) The two nearest points of $h \in S$ are located either on the left of h, or one on the left and one on the right (it is i), (c) $cost(h) < cost(i)$.*

Proof. (a)(b) Immediately. (c) Let h' be the 3rd rightmost point in S. Then $cost(h) \leq d(h', h) + d(h, i) < d(h', i) + d(h, i) = cost(i)$. □

Thus when we compute $cost(h, i; k)$ which is the minimum over $cost(x)$ for $x \in S$, we can ignore i since $cost(i) > cost(h)$.

Now we explain how to solve $PS2(h, i; k)$. We have two cases.

Case 1: (Base case) $k = 3$.

If h=1 then no solution exists. Otherwise the solution is $S = \{1, h, i\}$ for some h, and its cost is $cost(h) = d(1, i)$.

Case 2: (Inductive case) $k > 3$.

Now we compute the solution of $PS2(h, i; k)$. We can assume we have already computed the solutions of smaller problems $PS2(h', h; k-1)$ for $h' = 1, 2, ..., h-1$. By appending i to a solution of a smaller problem $PS2(h', h; k-1)$ for some h'. we can construct a solution S of $PS2(h, i; k)$, as follows. We have four subcases.

Subcase 2a: $cost(h, i; k)$ is $cost(h', h; k-1)$ for some h'.

Subcase 2b: $cost(h, i; k)$ is $cost(h)$ and the two nearest neighbors of h is located on the left of h.

Since $cost(h) > cost(h')$ holds where h' is the left neighbour of h this case does not occur.

Subcase 2c: $cost(h, i; k)$ is $cost(h)$ and the two nearest neighbors of h is located on the left and right of h.

In this case $cost(h, i; k)$ is $cost(h) = d(h', i)$.

Subcase 2d: $cost(h, i; k)$ is $cost(i)$.

Since $cost(i) > cost(h)$ this case does not occur.

Thus $cost(h, i; k)$ is $\max_{h'=1,2,\cdots,h-1} \min\{cost(h', h; k-1), d(h', i)\}$ and we can compute it in $O(n)$ time. The number of the subproblems is at most kn^2 and we can solve each subproblem in $O(n)$ time.

Theorem 3. *One can solve the PS2-dispersion problem in $O(kn^3)$ time.*

Similar to Lemma 2 we can prove that $cost(h', h; k-1)$ is a non-decreasing function with respect to h'. Then $\min\{cost(h', h; k-1), d(h', i)\}$ is a non-decreasing function with respect to h' up to some points, then it is a decreasing linear function with respect to h', so we can find the maximum one by binary search in $O(\log n)$ time.

We have the following thorem.

Theorem 4. *One can solve the PS2-dispersion problem in $O(kn^2 \log n)$ time.*

5. PSc-dispersion

We can naturally generalize $PS2$ problem to PSc problem for any integer $c \geq 3$ as follows.

Given a set P of n points on the horizontal line and an integer k, we wish to find a subset $S \subset P$ with $|S| = k$ maximizing the cost $\min_{x \in S}\{$ the sum of the distances to the nearest (constant) c points in S from x $\}$. We call the problem PSc-dispersion problem.

Then we can design subproblem $PSc(h_{c-1}, h_{c-2}, \cdots, h_1, i; k)$. The number of the subproblems is at most kn^c and we can solve each subproblem in $O(c^2 n) = O(n)$ time. Thus we can solve the PSc-dispersion problem in $O(kn^{c+1})$ time.

Theorem 5. *One can solve the PSc-dispersion problem in* $O(kn^{c+1})$ *time.*

6. Conclusion

In this paper we defined the LR-dispersion problem and gave an algorithm to solve the problem.

In this paper we gave an algorithm for the LR-dispersion problem. The running time of the algorithm is $O(kn^2 \log n)$. Also we gave some algorithms to solve some variants of the dispersion problem.

Can we apply the matrix search method [7] to solve those problem?

References

[1] T. Akagi and S. Nakano, On r-Gatherings on the Line, Proc. of FAW 2015, LNCS 9130, pp. 25-32 (2015).
[2] T. Akagi and S. Nakano, Dispersion on the Line, Technical Report, 2016-AL-158-3, IPSJ (2016)
[3] C. Baur and S.P. Feketee, Approximation of Geometric Dispersion Problems, Pro. of APPROX '98, Pages 63-75 (1998).
[4] B. Chandra and M. M. Halldorsson, Approximation Algorithms for Dispersion Problems, J. of Algorithms, 38, pp.438-465 (2001).
[5] Z. Drezner, Facility Location: A Survey of Applications and Methods, Springer (1995).
[6] Z. Drezner and H.W. Hamacher, Facility Location: Applications and Theory, Springer (2004).
[7] G. Frederickson, Optimal Algorithms for Tree Partitioning, Proc. of SODA '91 Pages 168-177 (1991).
[8] R. Hassin, S. Rubinstein and A. Tamir, Approximation Algorithms for Maximum Dispersion, Operation Research Letters, 21, pp.133-137 (1997).
[9] T. L. Lei and R. L. Church, On the unified dispersion problem: Efficient formulations and exact algorithms, European Journal of Operational Research, 241, pp.622-630 (2015).
[10] S. S. Ravi, D.J. Rosenkrantz and G. K. Tayi, Heuristic and Special Case Algorithms for Dispersion Problems, Operations Research, 42, pp.299-310 (1994).
[11] M. Sydow, Approximation Guarantees for Max Sum and Max Min Facility Dispersion with Parameterised Triangle Inequality and Applications in Result Diversification, Mathematica Applicanda, 42, pp.241-257 (2014).
[12] D. W. Wang and Yue-Sun Kuo, A study on Two Geometric Location Problems, Information Processing Letters, 28, pp.281-286 (1988).

