Locating Obnoxious Facilities on a Line Segment

Vishwanath R. Singireddy and Manjanna Basappa*

Computer Science Dept., BITS Pilani, Hyderabad Campus, Hyderabad, India.
{p20190420, manjanna}@hyderabad.bits-pilani.ac.in

The obnoxious facility location is a well-known topic in the operations research community. This paper addresses a variant of the problem, namely, the continuous obnoxious facility location on a line segment (COFL) problem, motivated by the following application. We wish to find locations for establishing k obnoxious or undesirable facilities (such as garbage dump yards, industries generating pollution, etc.) along a straight highway road such that the pairwise distance between these new facilities and the distance between each of the new facilities and other existing non-obnoxious facilities (such as hospitals, schools, etc.) are maximized. The formal definition of the problem is below:

The COFL problem: Given a horizontal line segment $\overline{pq}$ and a set P of n points lying above a line through $\overline{pq}$ in the Euclidean plane, we aim to locate points on $\overline{pq}$ for centering k non-overlapping congruent disks of maximum radius r_{max} such that none of the points in P lie interior to any of these disks.

Recently, it has been shown in [2] that we can solve the decision version of this problem (in which we are also given a radius L as input along with $\overline{pq}$ and the set P of n points) in time $O(n + k)$ if the points in P are given in order. Then, using this decision algorithm as a subroutine, an $(1-\epsilon)$-approximation algorithm (FPTAS) has been given to solve the COFL problem in time $O((n+k)\log\frac{||pq||}{2(k-1)\epsilon})$, where $\epsilon > 0$ and $||pq||$ is the length of the segment $\overline{pq}$ [2].

In this work, we show that we can solve the COFL problem exactly in polynomial time. Using the linear-time decision algorithm of [2] again as a subroutine, we present two polynomial-time exact algorithms based on two different approaches: (i) the algorithm is based on doing a binary search on all candidate radii L computed explicitly and runs in $O((nk)^2 \log(nk) + (n+k)\log(nk))$ time, and (ii) the algorithm is based on Megiddo's parametric search [1], and runs in $O((n + k)^2)$ time.

Algorithm based on binary search on candidate radii values: The approach here is that we first compute the set $\mathcal{L}$ of all candidate radii L, based on some observations of possible positions of all k disks in an optimal packing. We sort all the elements in $\mathcal{L}$. Then, by doing a binary search on the sorted $\mathcal{L}$, we invoke the decision algorithm of [2] each time we peek an element at the middle index by setting L to this element as the candidate radius. We continue this search until we find the radius L^* (maximum radius) such that for all $L \in \mathcal{L}$ with $L > L^*$ the decision algorithms returns NO. Hence, $L^* = r_{max}$.

* Corresponding author

Let $r_{\text{CAN}} \in \mathcal{L}$ be some candidate radius. Consider any two points $p_i, p_{i'} \in P$. Assume these two points are determining the radius of the disks in the optimal packing. Then point p_i can lie on the left boundary arc or the right boundary arc of the disk d_j, where $j = 1, 2, \ldots, k$ (see Fig 1). Similarly, the point $p_{i'}$ can also lie on the left boundary arc or the right boundary arc of the disk $d_{j'}$, where $j' = 1, 2, \ldots, k$. It is easy to observe that there are a constant number c of possible positions for the two disks d_j and $d_{j'}$ in an optimal solution such that the two points $p_i, p_{i'}$ lie on their boundaries and does not let their radius r_{CAN} to increase by repositioning all the k disks. Therefore, the number of all candidate radii r_{CAN} is $\binom{n}{2} \sum_{j=1}^{k} \sum_{j'=j}^{k} c = k^2 c \binom{n}{2} = O(n^2 k^2)$. The candidate radii values $r_{\text{CAN}} \in \mathcal{L}$ can be computed from the following equation:

$$\begin{aligned} x(p_{i'}) - x(p_i) =& 2(j' - j) r_{\text{CAN}} \pm \sqrt{r_{\text{CAN}}^2 - (y(p_i) - y(q))^2} \\ & \pm \sqrt{r_{\text{CAN}}^2 - (y(p_{i'}) - y(q))^2} \end{aligned}$$

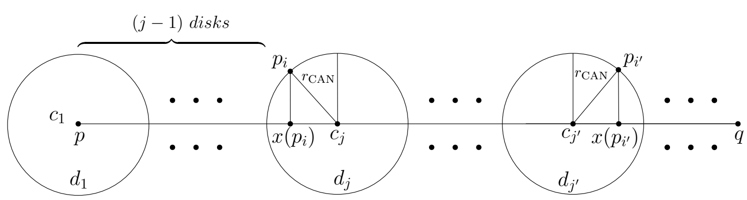

Fig. 1. Two points p_i and $p_{i'}$ influencing the radius r_{CAN}

Hence we have the following theorem.

Theorem 1. *For a given line segment $\overline{pq}$ and an ordered set P of n points in the Euclidean plane, we can solve the COFL problem optimally in $O((nk)^2 \log(nk) + (n+k)\log(nk))$ time.*

Algorithm based on parametric search: Here, we discuss an algorithm based on the general parametric search approach of Megiddo [1] so that its running time is independent of any numerical precision parameter $(\frac{1}{\epsilon})$ unlike the `FPTAS` given in [2]. As we know, in a solution based on parametric search, there is a test algorithm, and a decision algorithm wherein the test algorithm is typically a step-by-step simulation of the decision algorithm. Hence, we use the decision algorithm of [2] and simulate its steps without knowing what the value of L is.

Consider a point $p_i \in P$, having the two center points $c_{i,1}$ and $c_{i,2}$ on the segment $\overline{pq}$ such that the distance from p_i to either of these center points is L (see Fig. 2). Let the coordinates of these points be $p_i = (x(p_i), y(p_i))$, $c_{i,1} = (x_{i,1}, y_{i,1})$ and $c_{i,2} = (x_{i,2}, y_{i,2})$. Let $I = \{[x_{i,1}, x_{i,2}] | p_i \in P\}$ be the

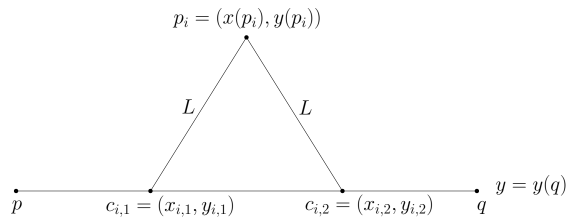

Fig. 2. A point p_i is having two center points of $\overline{pq}$ which are at unknown distance L

set of intervals on $\overline{pq}$ ordered from p to q. Let $I^{\mathsf{c}} = \{[x_{i,2}, x_{j,1}] | p_i, p_j \in P, i < j\} \cup \{[x_{0,2}, x_{1,1}], [x_{n,2}, x_{n+1,1}]\}$ be the set of complemented intervals of I, where every point in each interval $[x_{i,2}, x_{j,1}]$ is at distance at least L from every point in P and $x_{0,2} = x(p)$, $x_{n+1,1} = x(q)$.

On the simulation of the decision algorithm, we encounter a comparison step that involves the following inequality.

$$2((j+1-k)^2 - 1)L^2 + 4(j+1-k)(x(p_i) - x(p_{i-1}))L + (x(p_i) - x(p_{i-1}))^2 + 2\sqrt{(L^2 - (y(p_i) - y(q))^2)(L^2 - (y(p_{i-1}) - y(q))^2)} \leq 0 \qquad (1)$$

where i is the index of the current interval from I^{c}, $j \leq k$ is the number of disks packed so far, and $y(q)$ is the y-coordinate of $\overline{pq}$. In the polynomial on the right of the inequality (1), the only unknown term is L, and the maximum degree is 2. We simulate all the steps by solving polynomials in L with a degree at most 2. We get the values of L, then invoke the decision algorithm with each of these values as a candidate radius. If this call returns YES, then proceed to the simulation of the next step of the algorithm by appropriately resetting the variable such as j. Essentially, each time the decision algorithm returns YES for the guessed value of L, the radius L will near to r_{max} (optimal radius). Ultimately, when the simulation ends, we have that L equals r_{max}.

Finally, to construct the set of k disks $D = \{d_1, d_2, \ldots, d_k\}$, we run the decision algorithm with the computed L one more time. Therefore, we have the following theorem.

Theorem 2. *We have an algorithm to solve the COFL problem exactly in time* $O((n+k)^2)$ *using the parametric search technique.*

Keywords: Obnoxious facilities · Parametric search · Binary search · Exact algorithm

References

1. Megiddo, N.: Applying parallel computation algorithms in the design of serial algorithms. *Journal of the ACM*, 30(4), pp. 852–865 (1983).
2. Sigireddy V., R. and Basappa, M.: Constrained Obnoxious Facility Location on a Line Segment. In: Proceedings of the 33rd Canadian Conference on Computational Geometry, pp. 362–367 (2021).