Noname manuscript No.
(will be inserted by the editor)

Optimal Strategies for the One-Round Discrete Voronoi Game on a Line

Aritra Banik · Bhaswar B. Bhattacharya · Sandip Das

Received: date / Accepted: date

Abstract The one-round discrete Voronoi game, with respect to a n-point user set $\mathcal{U}$, consists of two players Player 1 (P1) and Player 2 (P2). At first, P1 chooses a set $\mathcal{F}_1$ of m facilities following which P2 chooses another set $\mathcal{F}_2$ of m facilities, disjoint from $\mathcal{F}_1$, where $m(= O(1))$ is a positive constant. The payoff of P2 is defined as the cardinality of the set of points in $\mathcal{U}$ which are closer to a facility in $\mathcal{F}_2$ than to every facility in $\mathcal{F}_1$, and the payoff of P1 is the difference between the number of users in $\mathcal{U}$ and the payoff of P2. The objective of both the players in the game is to maximize their respective payoffs. In this paper, we address the case where the points in $\mathcal{U}$ are located along a line. We show that if the sorted order of the points in $\mathcal{U}$ along the line is known, then the optimal strategy of P2, given any placement of facilities by P1, can be computed in $O(n)$ time. We then prove that for $m \geq 2$ the optimal strategy of P1 in the one-round discrete Voronoi game, with the users on a line, can be computed in $O(n^{m-\lambda_m})$ time, where $0 < \lambda_m < 1$, is a constant depending only on m.

A preliminary version of this paper has appeared in the *Proc. 17th Annual International Computing and Combinatorics Conference (COCOON)*, 2011, 213–224.

A. Banik
Advanced Computing and Microelectronics Unit
Indian Statistical Institute, Kolkata 700108, India
E-mail: aritrabanik@gmail.com

B. B. Bhattacharya
Department of Statistics,
Stanford University, Stanford, CA, USA
E-mail: bhaswar.bhattacharya@gmail.com

S. Das
Advanced Computing and Microelectronics Unit
Indian Statistical Institute, Kolkata 700108, India
E-mail: sandipdas@isical.ac.in

Keywords Computational geometry · Facility location · Optimization · Voronoi game.

1 Introduction

The main objective in any facility location problem is to judiciously place a set of facilities serving a set of users such that certain optimality criteria are satisfied. Facilities and users are generally modeled as points in the plane. The set of users (demands) is either *discrete*, consisting of finitely many points, or *continuous*, i.e., a region where every point is considered to be a user. We assume that the facilities are equally equipped in all respects, and a user always avails the service from its nearest facility. Consequently, each facility has its *service zone*, consisting of the set of users that are served by it. For a set $\mathcal{U}$ of users, finite or infinite, and a set $\mathcal{F}$ of facilities, define for every $f \in \mathcal{F}, \mathcal{U}(f, \mathcal{F})$ as the set of users in $\mathcal{U}$ that are served by the facility f. Many variations of facility location problems in both the discrete and continuous user category, under several optimality criteria, have been studied in the literature [8].

Competitive facility location is concerned with the favorable placement of facilities by competing market players, and has been studied in several contexts ([9, 10, 13]). In such a scenario, when the users choose the facilities based on the nearest-neighbor rule, the optimization criteria is to maximize the cardinality or the area of the service zone depending on whether the demand region is discrete or continuous, respectively. For continuous demand regions, this reduces to the problem of maximizing the area of the Voronoi regions point sets. Dehne et al. [7] addressed the problem of finding a new point q amidst a set of n existing points $\mathcal{F}$ such that the Voronoi region of q is maximized. They showed that when the points in $\mathcal{F}$ are in convex position, the area function has only a single local maximum inside the region where the set of Voronoi neighbors do not change. For the same problem, Cheong et al. [5] gave a near-linear time algorithm that locates the new optimal point approximately, when the points in $\mathcal{F}$ are in general position. A variation of this problem, involving maximization of the area of Voronoi regions of a set of points placed inside a circle, has been recently considered by Bhattacharya [2].

In the discrete user case, the analogous problem is to place a set of new facilities amidst a set of existing ones such that the number of users served by the new facilities is maximized. The problem of placing only one new facility has been addressed by Cabello et al. [4] and is referred to as the MaxCov problem. They showed that in the ℓ_1 and ℓ_∞ metrics, the MaxCov problem can be solved in $O(n \log n)$ time. It is shown that in the ℓ_2 metric, if the number of existing facilities $m \geq 2$, then the MaxCov problem is 3SUM hard, and an algorithm for finding the set of all possible optimal placements of the new facility in $O(n^2)$ time is also given. They also showed that for $m = 1$ the MaxCov problem in ℓ_2 metric can be solved in $O(n \log n)$ time, and this is asymptotically optimal under the algebraic decision tree model. The 2-MaxCov problem, which considers the problem of placing two new facilities, has been studied recently by Bhattacharya and Nandy [3].

A game-theoretic analogue of such competitive problems for continuous demand regions is a situation where two players place two disjoint set of facilities in the demand region. A player p is said to own a part of the demand region that is closer to the facilities owned by p than to the other player, and the player which finally owns the larger area is the winner of the game. The area a player owns at the end of the game is called the payoff of the player. In the *one-round game* the first player places m facilities following which the second player places another m facilities in the demand region. In the *m-round game* the two players place one facility each alternately for m rounds in the demand region.

Ahn et al. [1] studied a one-dimensional Voronoi game, where the demand region is a line segment. They showed that when the game takes m rounds, the second player always has a winning strategy that guarantees a payoff of $1/2+\varepsilon$, with $\varepsilon > 0$. However, the first player can force ε to be arbitrarily small. On the other hand, in the one-round game with m facilities, the first player always has a winning strategy. The one-round Voronoi game in $\mathbb{R}^2$ was studied by Cheong et al. [6], for a square-shaped demand region. They proved that for any placement W of the first player, with $|W| = m$, there is a placement B of the second player $|B| = m$ such that the payoff of the second player is at least $1/2 + \alpha$, where $\alpha > 0$ is an absolute constant and m large enough. Fekete and Meijer [11] studied the two-dimensional one-round game played on a rectangular demand region with aspect ratio ρ. They proved that the second player has a winning strategy if and only if $m \geq 3$ and $\rho > \sqrt{2}/n$, or $m = 2$ and $\rho > \sqrt{3}/2$. However, in none of the above cases, the optimal strategy of the two players, that is, the strategy that maximizes the respective payoffs of the players is known.

Similar competitive facility location problems, where the universe is modeled by a graph with weighted edges inducing distances, are also studied in the literature. Weighted nodes of the graph represent users and their demands, which are to be served by the competitive facilities. In the (r, p)-centroid problem [12], the two competitors sequentially place p and r facilities, respectively, on the edges of the given graph such that their individual payoffs are maximized.

In this paper, we study the one-round Voronoi game in $\mathbb{R}$ for a discrete demand set and devise algorithms for obtaining the optimal strategies of the two players. The one-round discrete Voronoi game consists of a finite user set $\mathcal{U}$, with $|\mathcal{U}| = n$, and two players Player 1 (P1) and Player 2 (P2) each having $m = O(1)$ facilities. At first, P1 chooses a set $\mathcal{F}_1$ of m facilities following which P2 chooses another set $\mathcal{F}_2$ of m facilities, disjoint from $\mathcal{F}_1$. The payoff of P2 is defined as the cardinality of the set of points in $\mathcal{U}$ which are closer to a facility owned by P2 than to every facility owned by P1. The payoff of P1 is the number of users in $\mathcal{U}$ minus the payoff of P2. The objective of both the players is to maximize their respective payoffs. For any two disjoint sets $\mathcal{F}$ and $\mathcal{S}$, with $|\mathcal{F}| = |\mathcal{S}| = m$, define $\mathcal{U}(\mathcal{F}, \mathcal{S}) = \bigcup_{f \in \mathcal{S}} \mathcal{U}(f, \mathcal{F} \cup \mathcal{S})$, and

$\nu(\mathcal{F}) = \max_{\mathcal{S}} |\mathcal{U}(\mathcal{F}, \mathcal{S})|$, where the maximum is taken over all sets $\mathcal{S}$, with $|\mathcal{S}| = m$. Given any placement $\mathcal{F}$ by P1, the quantity $\nu(\mathcal{F}) = \max_{\mathcal{S}} |\mathcal{U}(\mathcal{F}, \mathcal{S})|$ is the optimal payoff of P2. The placement $\mathcal{F}_2$ at which the above maximum is attained is the optimal strategy of P2, given P1 has placed the facilities at $\mathcal{F}$. Similarly, $\min_{\mathcal{F}} \nu(\mathcal{F})$ is the optimal payoff of P1, and the placement $\mathcal{F}_1$ by P1 at which the minimum is attained is called the optimal strategy of P1. Now the One-Round (m, n) Discrete Voronoi Game can be formally described as follows:

One-Round (m, n) Discrete Voronoi Game: Given a set $\mathcal{U}$ of n users and two players P1 and P2 having m facilities each, P1 chooses a set $\mathcal{F}_1$ of m facilities following which P2 chooses a set $\mathcal{F}_2$ of m facilities, disjoint from $\mathcal{F}_1$, such that:

(a) $\max_{\mathcal{S}} |\mathcal{U}(\mathcal{F}_1, \mathcal{S})|$ is attained at $\mathcal{S} = \mathcal{F}_2$, where the maximum is taken over all sets $\mathcal{S}$, with $|\mathcal{S}| = m$;

(b) $\min_{\mathcal{F}} \nu(\mathcal{F})$ is attained at $\mathcal{F} = \mathcal{F}_1$, where the minimum is taken over all sets $\mathcal{F}$, with $|\mathcal{F}| = m$.

In this paper, we study the One-Round (m, n) Discrete Voronoi Game when the users in $\mathcal{U}$ are located along a single straight line and the two players are also restricted to place their facilities along the same straight line. Henceforth, we shall call this variant of the discrete Voronoi game as the $G(m, n)$ game. We begin by showing that if the sorted order of the users in $\mathcal{U}$ along the line is known, then the optimal strategy of P2, given any placement of P1, in the $G(m, n)$ game can be found out in $O(n)$ time. Clearly, if $m = 1$, the optimal strategy of P1 is to place the facility at the median of $\mathcal{U}$. For $m \geq 2$, we give $O(n^{m+1})$ time algorithm for obtaining the optimal strategy of P1, which we then improve to provide a $O(n^{m-\lambda_m})$ time algorithm, where $0 < \lambda_m < 1$, is a constant depending only on m.

2 Optimal Strategy of P2

In this section we give an algorithm for determining the optimal strategy for P2, given any placement by P1. Let $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$ be a set of n users placed along a line ℓ. Assume that the sorted order of the users in $\mathcal{U}$ along ℓ is known. Moreover, let $\mathcal{F} = \{f_1, f_2, \ldots, f_m\}$ be the placement by P1 along ℓ. Consider any placement $\mathcal{S}$ by P2, with $|\mathcal{S}| = m$. From the definitions of payoffs of P1 and P2, we can assume that if $\min_{f \in \mathcal{F}} |u - f| = \min_{s \in \mathcal{S}} |u - s|$, for some user $u \in \mathcal{U}$, then u is served by the facility in $\mathcal{F}$. In order to find the optimal strategy of P2, we need to find a set $\mathcal{S}_o$ of m points, disjoint from $\mathcal{F}$, along ℓ such that $|\mathcal{U}(\mathcal{F}, \mathcal{S})|$ is maximized at $\mathcal{S} = \mathcal{S}_o$, where the maximum is taken over all sets $\mathcal{S}$ having $|\mathcal{S}| = m$. Note that we are allowed to place facilities on the users but the positions of every two different facility points must be distinct.

Suppose $f_1 < f_2 < \ldots < f_m$ is the sorted order of the facilities along the line ℓ. Define $f_0 = -M$ and $f_{m+1} = M$, where M is such that $\mathcal{U} \cup \mathcal{F} \subset [-M, M]$. Note that if P2 places a facility s in the interior of the interval

$[f_i, f_{i+1}]$, for some $i \in \{0, 1, \ldots, m\}$, then s can serve only those users in $\mathcal{U}$ which lie in the interval (f_i, f_{i+1}).

We now have the following observation which can be verified easily:

Observation 1 *For every $f_i \in \mathcal{F}$, P2 can place two new facilities s and s' in interior of $[f_i, f_{i+1}]$, such that $\mathcal{U}(\mathcal{F}, \{s, s'\}) = \mathcal{U} \cap (f_i, f_{i+1})$.*

Proof Let u and u' be the users in the interior of the interval $[f_i, f_{i+1}]$ nearest to the facilities f_i and f_{i+1}, respectively. The result now follows from placing the two new facilities s and s' at the midpoints of the intervals $[f_i, u]$ and $[u', f_{i+1}]$, respectively. □

This observation implies that by placing two facilities in the interior of some interval $J_i = [f_i, f_{i+1}]$, P2 can serve all the users in the interior of the interval J_i. Therefore, P2 incurs no extra gain by placing more than two facilities in the interior of some interval J_i, for $i \in \{0, 1, \ldots, m\}$. Associated with each interval J_i, we define the following two quantities:

a_i: This denotes the number of users in $\mathcal{U}$ which lie in the interior of the interval $[f_i, f_{i+1}]$. That is, $a_i = |\mathcal{U} \cap (f_i, f_{i+1})|$. Note that by Observation 1, P2 can place two facilities in J_i such that P2 serves all the a_i users in the interior of J_i.

b_i: This denotes the maximum number of users P2 can serve by placing a single new facility in the interior of the interval $[f_i, f_{i+1}]$. Note that $b_0 = a_0$ and $b_m = a_m$. For $i \in \{1, 2, \ldots, m-1\}$, b_i is the maximum number of users in J_i that can be covered by an open interval of length $|J_i|/2$. If the users in $\mathcal{U}$ are assumed to be sorted then this can be determined in $O(a_i)$ time by a simple linear scan.

Let $A = \{a_i | i \in \{0, 1, \ldots, m\}\}$ and $B = \{b_i | i \in \{0, 1, \ldots, m\}\}$. For $i \in \{0, 1, \ldots, m\}$ let $c_i = a_i - b_i$.

Observation 2 *$b_i \geq c_i$ for every $i \in \{0, 1, 2, \ldots, m\}$.*

Proof For $i = 0$ or $i = m$, $c_0 = c_m = 0$, and the result is immediate.

Therefore, suppose $i \in \{1, 2, \ldots, m-1\}$. Note that since $a_i = b_i + c_i$, to prove the above observation it suffices to show that $b_i \geq a_i/2$. Observe that either $|\mathcal{U} \cap (f_i, \frac{f_i + f_{i+1}}{2}]| \geq a_i/2$ or $|\mathcal{U} \cap [\frac{f_i + f_{i+1}}{2}, f_{i+1})| \geq a_i/2$. Without loss of generality assume that $|\mathcal{U} \cap (f_i, \frac{f_i + f_{i+1}}{2}]| \geq a_i/2$. Let u be the user in $[f_i, f_{i+1}]$ nearest to f_i. If P2 places a facility at $s' = \frac{f_i + u}{2}$, then the number of users served by s' is clearly greater than or equal to $a_i/2$. Now, since b_i is the maximum number of users P2 can acquire from the interval $[f_i, f_{i+1}]$ by placing a single facility, it follows that $b_i \geq a_i/2$, and the result follows. □

Since the sorted order of the users in $\mathcal{U}$ is known, the values in the set B can be obtained in $O(\sum_{i=1}^{m} a_i) = O(n)$ time. The sorted order of the facilities in $\mathcal{F}$ along ℓ can be found out in $O(m \log m)$, and so the values in A can also be computed in $O(n + m \log m)$ time. As we have considered m as constant therefore, the sorted order of the numbers in $B \cup C$ can be found in $O(n)$ time, where $C = \{c_i | i \in \{0, 1, 2, \ldots, m\}\}$.

Once the sorted order in $B \cup C$ is known, we select the largest m values from the set. Now, since $b_i \geq c_i$ for every $i \in \{0, 1, 2, \ldots, m\}$, it can be ensured that among the largest m elements in $B \cup C$, no c_i is selected without selecting the corresponding b_i. If for some $i \in \{0, 1, 2, \ldots, m\}$ only b_i is selected, then P2 places only one facility in the interval J_i at the point where it can serve b_i users in $\mathcal{U}$. If for some $i \in \{0, 1, 2, \ldots, m\}$ both b_i and c_i are selected, then P2 places two new facilities in the interval J_i as described in Observation 1. Clearly, the placement by P2 obtained in this way gives the optimal strategy for P2, and we summarize this result in the following theorem:

Theorem 1 *The optimal placement by P2 in the $G(m, n)$ game, given any placement by P1, can be obtained in $O(n)$ time, when the sorted order of the users along the line is known.* □

Let $f_1, f_2, \ldots, f_m$ be any placement of P1, in the sorted order along the line ℓ. Let $\mathcal{U}$ denote the set of n users on ℓ. As these m facilities may coincide with some of the users, we have $\sum_{i=0}^{m} a_i \geq n - m$. Note that P2 can always serve b_i users in the interval J_i by placing a single facility. Let $s_1, s_2, \ldots, s_{m-1}$ be facilities placed by P2, such that $s_i \in J_i$ and s_i serves b_i users in J_i, for $i \in \{1, 2, \ldots, m-1\}$. Now, without loss of generality, assume that $b_0 \geq b_m$. Therefore, by placing one more facility s^* just to the left of f_1, P2 can serve b_0 of $\mathcal{U}$. Hence, in the placement $\mathcal{S} = \{s^*, s_1, s_2, \ldots, s_{m-1}\}$ P2 serves a total of $\sum_{i=0}^{m-1} b_i$ users. From Observation 2, we know that for $i \in \{1, 2, \ldots, m-1\}$ $b_i \geq a_i/2$. Moreover, $b_0 \geq \frac{b_0+b_m}{2} = \frac{a_0+a_m}{2}$ which implies that $\sum_{i=0}^{m} b_i \geq \sum_{i=0}^{m-1} a_i/2 = (n-m)/2$. This shows that irrespective of the arrangement of the users and the placement of P1, at least $(n-m)/2$ users can always be served by P2.

Using this fact we shall now construct an arrangement of users on ℓ such that P1 wins the One-Round (m, n) Discrete Voronoi Game on ℓ.

Remark 1: Suppose that $k = n/m$ is an odd-positive integer. Consider an arrangement of set of users $\mathcal{U}_o$ along ℓ as follows. Let $A_1, A_2, \ldots, A_m$ is a partition of $\mathcal{U}_o$ such that A_i is on the left of A_{i+1} for all $1 \leq i < m$. Cardinality of each subset A_i is k. Let the diameter of all subset A_i are same and equal to d. For any $1 \leq i < m$, gap between rightmost point of A_i and left most point of A_{i+1} is $2d$ (refer to Figure 1). Consider the One-Round (m, n) Discrete Voronoi Game on ℓ with user set $\mathcal{U}_o$. If $f_1, f_2, \ldots, f_m$ is a placement of P1, with f_i in the median of A_i as shown in Figure 1, then P2 can serve at most $(k-1)/2$ many users from each of the set A_i. Hence, P2 can serve at most $(n-m)/2$ users in total, and this is attained by P2 by placing the facility s_i just to the right of f_i, for $m \in \{1, 2, \ldots, m\}$. This shows that there is an arrangement of users along ℓ for which, given the optimal placement of P1, at most $(n-m)/2$ users can be served by P2, that is, P2 loses the game.

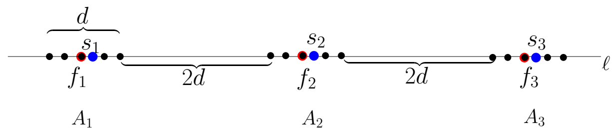

Fig. 1 An instance of the One-Round (m, n) Discrete Voronoi Game where P2 loses

There are however instances where P2 wins the game. An example of such an arrangement is given as follows:

Remark 2: Identify the line ℓ with $\mathbb{R}$, and let $\mathcal{U}_o$ be a set of n users with co-ordinates $1, 2, 4, \ldots, 2^{n-1}$ on ℓ. We shall prove that the One-Round (m, n) Discrete Voronoi Game with user set $\mathcal{U}_o$ is always won by P2, for $m \geq 2$ and large enough n. To prove this suppose that f and f' are any two consecutive facilities placed by P1 such that $f \in [2^a, 2^{a+1})$ and $f' \in (2^{a+k}, 2^{a+k+1}]$, for integers $a, k \geq 1$. Assume P2 places a single facility at $s = 2^{a+1}$. Then $(s + f')/2 = 2^a + f'/2 \geq 2^a + 2^{a+k-1} > 2^{a+k-1}$, which implies that s serves at least $k - 1$ users out of the k users in the interval (f, f'). Now, consider any placement $f_1, f_2, f_3, \ldots, f_m$ of P1 in the increasing order. Define $f_0 = 0$ and $f_{m+1} = 2^{n-1} + 1$. For $i \in \{0, 1, \ldots, m\}$, let $k_i = (f_i, f_{i+1}) \cap \mathcal{U}_o$. By placing a facility just to the left or right of f_1 or f_m, P2 can serve all the users in the interval $[f_0, f_1)$ or $(f_m, f_{m+1}]$, respectively. Therefore, by placing a facility s_i in the interval (f_i, f_{i+1}), P2 can always serve at least $k_i - 1$ users in that interval. As $\sum_{i=0}^{m} k_i \leq n$, there exists an index $j \in \{0, 1, \ldots, m\}$ such that $k_j \leq \frac{n}{m+1}$. Consider the placement $\mathcal{S} = \{s_0, s_1, \ldots, s_m\} \backslash \{s_j\}$ of m facilities by P2. Therefore, the total number of users served by P2 is at least

$$\sum_{i \neq j} (k_i - 1) \geq (n - m) - k_j - m \geq n \left(1 - \frac{1}{m+1}\right) - 2m > n/2,$$

whenever $m \geq 2$ and $n > \frac{4m(m+1)}{m-1}$. Therefore, for $m \geq 2$ and sufficiently large values of n, P2 wins the One-Round (m, n) Discrete Voronoi Game with user set $\mathcal{U}_o$.

These two remarks show that, unlike the continuous version, the One-Round (m, n) Discrete Voronoi Game with users on a line may be won by either one of the two players, depending on the arrangement of the users on the line. This motivates the design of algorithms for obtaining optimal placements of the players, given any arrangement of users on the line. The optimal strategy of the P2, given any placement of the P1, was given in this section. In the following section we develop the optimal strategy of P1.

3 Optimal Strategy of P1

It is easy to see that if $m = 1$, that is, every player places one facility, then the optimal strategy of P1 is to place the facility at the median of the set of users

$\mathcal{U}$, which can be obtained in $O(n)$ time. In this section we shall prove a combinatorial result, which will be used to obtain a characterization of the optimal placement by P1. Using this characterization we shall provide an algorithm for determining the optimal placement by P1, when $m \geq 2$.

3.1 An Important Lemma

Let $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$ be the set of users sorted along the line ℓ and $M \in \ell$ be such that $\mathcal{U} \subset [-M, M]$. For any two points $x_1, x_2 \in \ell$ denote by $a([x_1, x_2])$ the total number of users in $\mathcal{U} \cap (x_1, x_2)$. Also, let $b([x_1, x_2])$ be the maximum number of users in $\mathcal{U} \cap (x_1, x_2)$ that can be covered by an open interval of length $|x_2 - x_1|/2$. For any fixed point $x \in \ell$ and a non-negative integer k define $x(k) \in \ell$ to be the largest value such that $b([x, x(k)]) = k$. Note that if $a([x, M]) < k$, $x(k)$ does not exist. If $a([x, M]) = k$, then according to the definition $x(k)$ goes to infinity. To avoid this, we define $x(k) = x + 2|M - x|$, whenever $a([x, M]) = k$.

We shall now prove that if $a([x, M]) > k$, then $x(k)$ always exists and can be obtained in $O(n)$ time.

Lemma 1 *Let k be any fixed positive integer, and $x_1 \in \ell$ be a point such that $a([x_1, M]) > k$. Then there exists a point $x_2 \in \ell$ such that $x_2 > x_1$ and $b([x_1, x_2]) = k$. In this case, the maximum value $x_1(k) \in \ell$ such that $b([x_1, x_1(k)]) = k$ also exists, and can be computed in $O(a([x_1, x_1(k)])) = O(n)$ time.*

Proof For any i such that $u_i > x_1$, define

$$J_i = [u_{\kappa(i)}, u_{\kappa(i)+k}], \text{ where } |u_{\kappa(i)+k} - u_{\kappa(i)}| = \min_{\substack{x_1 \leq u_j \leq u_i \\ u_{j+k} \leq u_n}} |u_{j+k} - u_j|.$$

Note that each J_i contains $k+1$ users in $\mathcal{U}$. Let $d_i = x_1 + 2|J_i|$. Observe that for all i such that $u_{i-1} > x_1$ we have,

$$\begin{aligned} |J_i| = |u_{\kappa(i)+k} - u_{\kappa(i)}| &= \min_{\substack{x_1 \leq u_j \leq u_i \\ u_{j+k} \leq u_n}} |u_{j+k} - u_j| \\ &\leq \min_{\substack{x_1 \leq u_j \leq u_{i-1} \\ u_{j+k} \leq u_n}} |u_{j+k} - u_j| \\ &= |u_{\kappa(i-1)+k} - u_{\kappa(i-1)}| = |J_{i-1}|, \end{aligned}$$

which implies that $d_i \leq d_{i-1}$.

Suppose $u_a > x_1$ be any user in $\mathcal{U}$ and $z \geq \max\{u_{a+k}, d_a\}$. We know from the definition of d_a that there exist some user $u_q \in [x_1, u_a]$ such that $d_a = x_1 + 2|u_{q+k} - u_q|$. Therefore, $[u_q, u_{q+k}] \subset [x_1, z]$. Then there exists $\varepsilon > 0$ such that either $u_{q+k} + \varepsilon < z$ or $u_q - \varepsilon > x_1$. Depending upon the situation, define $J = (u_q + \varepsilon, u_{q+k} + \varepsilon)$ or $J = (u_q - \varepsilon, u_{q+k} - \varepsilon)$, respectively. In either case, $J \subset [x_1, z]$ and $|J| = |u_{q+k} - u_q| = (d_a - x_1)/2 \leq (z - x_1)/2$. As J contains

k users, this implies that for all points $z \geq \max\{u_{a+k}, d_a\}$, $b([x_1, z]) \geq k$, whenever $u_a > x_1$.

Now, the following two different cases may arise:

Case 1: There exists some $j \leq n$ such that $d_j < u_{j+k}$. Let j_o be the minimum index j for which this holds. Now, as $d_{j_o} < u_{j_o+k}$, we have $b([x_1, u_{j_o+k}]) \geq k$. Let $J_r = [u_r, u_{r+k}]$ be the minimum length closed interval containing $k+1$ users and contained in $[x_1, u_{j_o+k})$. Clearly, $r < j_o$, and this implies that $d_{j_o} \leq d_r$. Moreover, as j_o is the minimum index such that $d_{j_o} < u_{j_o+k}$, it follows that $u_{r+k} \leq d_r$. This implies that $b([x_1, d_r]) \geq k$. Now depending on whether $d_r < u_{j_o+k}$ or $d_r \geq u_{j_o+k}$ we will show that either $x_1(k) = d_r$ or $x_1(k) = u_{j_o+k}$ respectively.

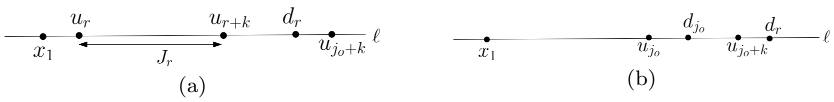

Fig. 2 Illustration for the proof of Lemma 1: (a) *Case* 1.1, and (b) *Case* 1.2.

Case 1.1: $d_r < u_{j_o+k}$ (refer to Fig. 2(a)). We know that $b([x_1, d_r]) \geq k$. Observe that for all $x > d_r$, there exists an $\varepsilon > 0$ such that the open interval $(u_r - \varepsilon, u_{r+k} + \varepsilon)$ contains $k+1$ users in $\mathcal{U}$, and $|u_{r+k} - u_r + 2\varepsilon| \leq |x - x_1|/2$. This implies that $b([x_1, x]) \geq k+1$, for $x > d_r$. Suppose, if possible, $b([x_1, d_r]) > k$. This means that there exist an open interval $G \subset [x_1, d_r]$ such that $|G \cap \mathcal{U}| \geq k+1$ and $|G| \leq |d_r - x_1|/2$. Hence, there exists a closed interval $H \subset G$ contained in $[x_1, d_r]$ containing $k+1$ points of U, with $|H| < |G| \leq |d_r - x_1|/2$. This closed interval H contradicts the minimality of the interval J_r. This proves that $b([x_1, d_r]) = k$ and $b([x_1, x]) \geq k+1$, for all $x > d_r$, implying that in this case $x_1(k) = d_r$.

Case 1.2: $d_r \geq u_{j_o+k}$ (refer to Fig. 2(b)). We know that $b([x_1, u_{j_o+k}]) \geq k$ and $b([x_1, x]) \geq k+1$, for all $x > u_{j_o+k}$. Suppose, if possible, $b([x_1, u_{j_o+k}]) > k$. This means that there exist an open interval $G \subset [x_1, u_{j_o+k})$ such that $|G \cap \mathcal{U}| \geq k+1$ and $|G| < |u_{j_o+k} - x_1|/2$. Hence, there exists a closed interval $H = [u_q, u_{q+k}] \subset G \subset [x_1, u_{j_o+k})$ containing $k+1$ points of U, with $|u_{q+k} - u_q| < |G| \leq |u_{j_o+k} - x_1|/2$. This implies that $d_q < u_{j_o+k} < d_r$. This implies that $|H| = |u_{q+k} - u_q| < |u_{r+k} - u_r| = |J_r|$, which contradicts the minimality of the interval J_r. This proves that $b([x_1, u_{j_o+k}]) = k$ and $b([x_1, x]) \geq k+1$, for all $x > u_{j_o+k}$, implying that in this case $x_1(k) = u_{j_o+k}$.

Note that in order to find $x_1(k)$ in this case we need to check up to u_{j_o+k}. As d_i can be determined in constant time, given d_{i-1}, we can compute $x_1(k)$ in time $O(a([x_1, u_{j_o+k}]))$. If $u_{j_o+k} \leq d_r$, then $x_1(k) = u_{j_o+k}$ and the complexity result is immediate. Now, suppose $u_{j_o+k} > d_r$. In this case $x_1(k) = d_r$. However, as $r \leq j_o - 1$, $d_r \geq d_{j_o-1}$ and by the minimality of j_o

we have $d_r \geq d_{j_o-1} \geq u_{j_o+k-1}$. Therefore, $a([x_1, u_{j_o+k}]) < a([x_1, d_r]) + 1$, and so $x_1(k)$ can be determined in $O(a([x, x_1(k)]))$ time.

Case 2: For all $u_i \geq x_1$, $d_i \geq u_{i+k}$. This implies that $d_n = d_{n-k} \geq u_n$ and the minimum length closed interval containing $k+1$ users is $[u_{\kappa(n)}, u_{\kappa(n)+k}]$. Therefore, from arguments exactly similar to those in the previous case we get $x_1(k) = d_n$. □

3.2 A Characterization of the Optimal Placement by P1

In this section we propose a simple characterization for the optimal placement by P1. Using this and Lemma 1, we then propose an algorithm for determining the optimal placement by P1.

Lemma 2 *There always exists an optimum placement* $\mathcal{F}' = \{f'_1, f'_2, \ldots, f'_m\}$ *by P1, such that* $f'_1 \in \mathcal{U}$*, and for all* $i \in \{2, 3, \ldots, m\}$ $f'_i \in \mathcal{U}$ *or* $f'_i = f'_{i-1}(k)$*, for some* $0 \leq k \leq a([f'_{i-1}, u_n])$.

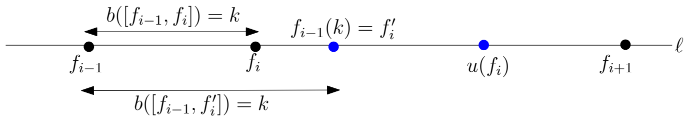

Fig. 3 Illustration for the proof of Lemma 2.

Proof Note that there always exists an optimal placement $\mathcal{F} = \{f_1, f_2, \ldots, f_m\}$ by P1 such that $[f_i, f_{i+1}] \cap \mathcal{U} \neq \emptyset$. We start with any such optimal solution $\mathcal{F}$.

Suppose $f_1 \in (u_{j-1}, u_j)$, for some $j \in \{2, 3, \ldots, n\}$. Then $\mathcal{F} \backslash \{f_1\} \cup \{u_j\}$ is also a placement by P1 such that $\nu(\mathcal{F}) = \nu(\mathcal{F} \backslash \{f_1\} \cup \{u_j\})$. Therefore, define $f'_1 = u_j \in \mathcal{U}$.

Let $i \in \{2, 3, \ldots, m\}$ be the smallest index $i \geq 2$ such that $f_i \in \mathcal{F}$ does not belong to $\mathcal{U}$ and not equal $f_{i-1}(k)$, for all $0 \leq k \leq a([f_{i-1}, u_n])$. Thus, let $u(f_i)$ be the user in the interval $[f_i, f_{i+1}]$ which is closest to the facility f_i. Assume that $b([f_{i-1}, f_i]) = k$. Note that since $[f_i, f_{i-1}(k)]$ is the maximum length such that $b([f_i, f_{i-1}(k)]) = k$, it follows that $f_{i-1}(k) > f_i$. Define f'_i be $u(f_i)$ or $f_{i-1}(k)$, depending on whichever is closer to the point f_i (see Figure 3).

Now, observe $b([f_{i-1}, f_i]) = b([f_{i-1}, f'_i]) = k$ and $a([f_{i-1}, f_i]) = a([f_{i-1}, f'_i])$. Moreover, as $[f'_i, f_{i+1}] \subset [f_i, f_{i+1}]$ we have

$$b([f'_i, f_{i+1}]) \leq b([f_i, f_{i+1}]) \quad \text{and} \quad a([f'_i, f_{i+1}]) \leq a([f_i, f_{i+1}]).$$

From the strategy of P2 described in Section 2, it can now be easily shown that $\nu(\mathcal{F} \backslash \{f_i\} \cup \{f'_i\}) \leq \nu(\mathcal{F})$. From the optimality of $\mathcal{F}$, it follows that $\nu(\mathcal{F} \backslash \{f_i\} \cup \{f'_i\}) = \nu(\mathcal{F})$.

The lemma now follows by applying the same argument on the set $\mathcal{F} \backslash \{f_i\} \cup \{f'_i\}$, which is also an optimal strategy for P1. □

Using this result we can devise a simple algorithm for obtaining the optimal strategy of P1. This is explained in the following theorem:

Theorem 2 *The optimal placement by P1 in the game $G(m,n)$ can be obtained in $O(n^{m+1})$ time.*

Proof From Lemma 2, it can be concluded that the number of possible placements by P1 that needs to be checked for finding the optimal placement is $O(n^m)$. From Lemma 1, we know that for each such placement $\mathcal{F}=\{f_1,f_2,\ldots,f_m\}$, $f_i(k)$ can be computed in $a([f_i,f_i(k)])$ time. Hence, every placement by P1 that needs to be checked can be obtained in $O(n)$ time. Given a placement by P1, the optimal placement by P2 can be obtained in $O(n)$ time (from Theorem 1). Therefore, the placement by P1 that minimizes the payoff of P2 can be obtained in $O(n^{m+1})$ time. □

4 Improving the Algorithm for the Optimal Strategy of P1

In this section we shall investigate into the structure of the game $G(m,n)$ more carefully and propose an improved algorithm for obtaining the optimal strategy for P1. As before, let $\mathcal{U}=\{u_1,u_2,\ldots,u_n\}$ be the set of users in the sorted order along the line ℓ. Let M be such that $\mathcal{U}\subset[-M,M]$. For any placement $\mathcal{F}=\{f_1,f_2,\ldots,f_m\}$ define the $(m+1)\times 1$ *payoff vector* $\boldsymbol{P}_{\mathcal{F}}$ as follows:

$$\boldsymbol{P}_{\mathcal{F}}=\begin{pmatrix}(b([-M,f_1]),c([-M,f_1]))\\(b([f_1,f_2]),c([f_1,f_2]))\\(b([f_2,f_3]),c([f_2,f_3]))\\\vdots\\(b([f_{m-1},f_m]),c([f_{m-1},f_m]))\\(b([f_m,M]),c([f_m,M])),\end{pmatrix},$$

that is, the i-th element of the payoff vector for $2\leq i\leq m$ is $(b([f_{i-1},f_i]),$ $c([f_{i-1},f_i]))$, and the first and the last element of $\boldsymbol{P}_{\mathcal{F}}$ is $(b([-M,f_1]),c([-M,f_1]))$ and $(b([f_m,M]),$ $c([f_m,M]))$, respectively.

We denote the transpose of a vector $\boldsymbol{P}$ by $\boldsymbol{P}'$. By a $(m+1)\times 1$ vector of ordered pairs we mean a vector

$$\boldsymbol{P}=((b_1,c_1),(b_2,c_2),\ldots,(b_{m+1},c_{m+1}))'=((b_i,c_i))_{1\leq i\leq m+1},$$

where b_i,c_i are non-negative integers with $\sum_{i=1}^{m+1}b_i+c_i=O(n)$. With this notation, we have the following definition:

Definition 1 An $(m+1)\times 1$ vector of order pairs $\boldsymbol{P}=((b_i,c_i))_{1\leq i\leq m+1}$, is said to be *feasible* if there exists a placement $\mathcal{F}=\{f_1,f_2,\ldots,f_m\}$ of facilities on the line ℓ such that

(i) $b_1\geq b([-M,f_1])$, $b_{m+1}\geq b([f_m,M])$, and $b_i\geq b([f_{i-1},f_i])$, for $2\leq i\leq m-1$,

(ii) $b_1+c_1 \geq a([-M, f_1])$, $b_{m+1}+c_{m+1} \geq a([f_m, M])$, and $b_i+c_i \geq a([f_{i-1}, f_i])$, for $2 \leq i \leq m-1$.

The placement $\mathcal{F}$ of facilities on the line ℓ is said to satisfy the vector $\boldsymbol{P}$. A vector of ordered pairs is said to be infeasible if it is not feasible.

Interpreting this in terms of the game, a vector $\boldsymbol{P} = ((b_i, c_i))_{1\leq i\leq m+1}$ is feasible if there exist a placement $\mathcal{F} = \{f_1, f_2, \dots, f_m\}$ by P1 such that P2 can serve at most b_i users in $[f_{i-1}, f_i]$ by placing one facility, and can serve at most $b_i + c_i$ users by placing two facilities in the interval $[f_{i-1}, f_i]$.

The following observation is now immediate from the definition:

Observation 3 *If the vector $\boldsymbol{P} = ((b_i, c_i))_{1\leq i\leq m+1}$ is infeasible then the vector $\boldsymbol{Q} = ((\beta_i, \gamma_i))_{1\leq i\leq m+1}$ is also infeasible, whenever $\beta_i \leq b_i$ and $\gamma_i \leq c_i$ for all $1 \leq i \leq m+1$.* □

Lemma 3 *The feasibility of any vector of ordered pairs $\boldsymbol{P} = ((b_i, c_i))_{1\leq i\leq m+1}$ can be determined in $O(n)$ time. Moreover, if $\boldsymbol{P}$ is feasible, then a placement $\mathcal{F} = \{f_1, f_2, \dots, f_m\}$ of facilities on the line ℓ that satisfies $\boldsymbol{P}$ can also be obtained in $O(n)$ time.*

Proof Define $f_1 = u_{b_1+1}$. For any $i \geq 2$, define $f'_i \in U$ to be such that $a([f_{i-1}, f'_i]) = b_i + c_i$, and $f''_i = f_{i-1}(b_i)$. For all $2 \leq i \leq m+1$ define f_i to be f'_{i-1} or f''_{i-1} depending on whichever is closer to f_{i-1}. If for some $1 \leq i \leq m+1$, f_i is not well defined, then we define $f_i = M$. Denote $\mathcal{F} = \{f_1, f_2, \dots f_m\}$. Given f_{i-1} it takes constant time to determine f'_i. From Lemma 1 we know that it is possible to find $f''_i = f_{i-1}(b_i)$ in $O(a([f_{i-1}, f''_i]))$ time. Therefore, the set $\mathcal{F}$ can be constructed in $O(\sum_{i=2}^{m} a([f_{i-1}, f''_i])) = O(\sum_{i=2}^{m} b_i) = O(n)$ time.

Note that if $a([f_m, M]) \leq b_m + c_m$, then by the construction $\mathcal{F}$ satisfies $\boldsymbol{P}$ and it is feasible.

Next, we claim that if $a([f_m, M]) > b_m+c_m$, then $\boldsymbol{P}$ is infeasible. If possible, suppose $\boldsymbol{P}$ is feasible and assume $\tilde{\mathcal{F}} = \{\tilde{f}_1, \tilde{f}_2, \dots \tilde{f}_m\}$ satisfies $\boldsymbol{P}$. This implies that $a([\tilde{f}_m, M]) \leq b_m + c_m$. Hence, $|M - \tilde{f}_m| < |M - f_m|$. This implies that $\tilde{f}_m > f_m$. Let j be the smallest index such that $\tilde{f}_j > f_j$. Note that $\tilde{f}_1 \leq f_1 = u_{b_1+1}$. Define $f_0 = \tilde{f}_0 = -M$ and $f_{m+1} = \tilde{f}_{m+1} = M$. Then $\tilde{f}_{j-1} \leq f_{j-1}$, which implies that $[f_{j-1}, f_j] \subset [\tilde{f}_{j-1}, \tilde{f}_j]$. Now we know that f_j is either of the form $f'_{j-1} \in U$ or of the form $f''_{j-1} = f''_{j-1}(b_j)$. We consider these two cases separately:

Case 1: $f_j = f'_{j-1} \in U$. Then as $\tilde{f}_j > f'_{j-1} \in U$, $b_j + c_j = a([f_{j-1}, f'_{j-1}]) < a([\tilde{f}_{j-1}, \tilde{f}_j])$, $\tilde{\mathcal{F}}$ does not satisfy $\boldsymbol{P}$.

Case 2: $f_j = f''_{j-1} = f_{j-1}(b_j)$. Then from $\tilde{f}_j > f''_{j-1}$, $f_{j-1} \leq \tilde{f}_{j-1}$, and the maximality of $f_{j-1}(b_j)$, we get $b_j \leq b([f_{j-1}, f''_{j-1}]) < b([f_{j-1}, \tilde{f}_j]) \leq b([\tilde{f}_{j-1}, \tilde{f}_j])$. Hence $\tilde{\mathcal{F}}$ does not satisfy $\boldsymbol{P}$.

Hence, we have proved that in linear time it is possible to determine whether a vector of order pairs $\boldsymbol{P}$ is feasible or not. □

Let $\mathcal{F} = \{f_1, f_2, \ldots, f_m\}$ be any placement by P1. As before, let $\boldsymbol{P}_{\mathcal{F}}$ be the payoff vector and define multisets $B_{\mathcal{F}} = \{b([-M, f_1]), b([f_1, f_2]), \ldots, b([f_m, M])\}$, and $C_{\mathcal{F}} = \{c([-M, f_1]), c([f_1, f_2]), \ldots, c([f_m, M])\}$. Clearly, $\mathcal{F}$ satisfies $\boldsymbol{P}_{\mathcal{F}}$ and so $\boldsymbol{P}_{\mathcal{F}}$ is feasible. Define $\nu_1(\mathcal{F}), \nu_2(\mathcal{F}), \ldots, \nu_m(\mathcal{F})$ as follows:

$$\begin{aligned}
\nu_1(\mathcal{F}) &= \max\{B_{\mathcal{F}} \cup C_{\mathcal{F}}\} \\
\nu_2(\mathcal{F}) &= \max\{(B_{\mathcal{F}} \cup C_{\mathcal{F}}) \backslash \{\nu_1(\mathcal{F})\}\} \\
&\vdots \\
\nu_m(\mathcal{F}) &= \max\{(B_{\mathcal{F}} \cup C_{\mathcal{F}}) \backslash \{\nu_1(\mathcal{F}), \nu_2(\mathcal{F}), \ldots, \nu_{m-1}(\mathcal{F})\}\} \qquad (1)
\end{aligned}$$

Let $\boldsymbol{R}_{\mathcal{F}}$ be the vector of ordered pairs obtained from $\boldsymbol{P}_{\mathcal{F}}$ by replacing all elements of $(B_{\mathcal{F}} \cup C_{\mathcal{F}}) \backslash \{\nu_1(\mathcal{F}), \nu_2(\mathcal{F}), \ldots, \nu_{m-1}(\mathcal{F}), \nu_m(\mathcal{F})\}$ by the element $\nu_m(\mathcal{F})$. Then by the definition of feasibility, $\boldsymbol{R}_{\mathcal{F}}$ is feasible. Moreover, if $\mathcal{F}'$ satisfies $\boldsymbol{R}_{\mathcal{F}}$, then the optimal payoff of P2 given P1 has placed at $\mathcal{F}'$ is $\nu(\mathcal{F}') \leq \sum_{i=1}^{m} \nu_i(\mathcal{F})$.

For non-negative integers $i_1 \geq i_2 \geq \ldots \geq i_m$, let $A(i_1, i_2, \ldots, i_m)$ be the multiset of $2m + 2$ elements where each i_k is repeated once, for $1 \leq k \leq m - 1$, and i_m is repeated $m + 3$ times. Define $\mathcal{S}(i_1, i_2, \ldots, i_m)$ to be the class of sub-multisets of $A(i_1, i_2, \ldots, i_m) \times A(i_1, i_2, \ldots, i_m)$ of cardinality $m + 1$. Every element of $\mathcal{S}(i_1, i_2, \ldots, i_m)$ being a multiset of order $m + 1$ can be considered as an $(m+1) \times 1$ vector of ordered pairs. For every such vector $\boldsymbol{P} \in S(i_1, i_2, \ldots, i_m)$, denote by $\pi(\boldsymbol{P})$ as the collection of all $(m + 1) \times 1$ vectors of ordered pairs obtained by permuting the rows of $\boldsymbol{P}$. Define $\mathcal{M}(i_1, i_2, \ldots, i_m) = \bigcup \pi(\boldsymbol{P})$, where the union is taken over all $\boldsymbol{P} \in S(i_1, i_2, \ldots, i_m)$.

We say that the class $\mathcal{M}(i_1, i_2, \ldots, i_m)$ is *feasible* if there exists some vector in $\mathcal{M}(i_1, i_2, \ldots, i_m)$ which is feasible. The class $\mathcal{M}(i_1, i_2, \ldots, i_m)$ is said to be *infeasible* if it is not feasible. The *weight* of the class $\mathcal{M}(i_1, i_2, \ldots, i_m)$ is defined $\sum_{k=1}^{m} i_k$. Observe, if $\boldsymbol{P} \in \mathcal{M}(i_1, i_2, \ldots, i_m)$ is feasible and $\mathcal{F}$ satisfies $\boldsymbol{P}$, then the payoff of P2, when P1 places at $\mathcal{F}$, is $\nu(\mathcal{F}) \leq \sum_{k=1}^{m} i_k$.

For every placement $\mathcal{F}$ by P1, there exists a vector $\boldsymbol{P} \in \mathcal{M}(\nu_1(\mathcal{F}), \nu_2(\mathcal{F}), \ldots, \nu_m(\mathcal{F}))$ such that $\boldsymbol{P}$ is feasible. Therefore, by Observation 3 if for given non-negative integers $\nu_1 \geq \nu_2 \geq \ldots \geq \nu_m$, $\mathcal{M}(\nu_1, \nu_2, \ldots, \nu_m)$ is infeasible, then there cannot exist any placement $\mathcal{F}$ by P1 with $\nu_i = \nu_i(\mathcal{F})$, for all $1 \leq i \leq m$.

With these observations, we now have the following lemma:

Lemma 4 *Suppose $\mathcal{M}(i_1, i_2, \ldots, i_m)$ is the minimum weight feasible class. If $\boldsymbol{P} \in \mathcal{M}(i_1, i_2, \ldots, i_k)$ is feasible and $\mathcal{F}$ satisfies $\boldsymbol{P}$, then $\mathcal{F}$ is an optimal placement by P1.*

Proof If possible, suppose $\mathcal{F}$ is not optimal. Then there exits some other placement $\mathcal{F}'$ by P1, with $\nu(\mathcal{F}') = \sum_{i=1}^{m} \nu_i(\mathcal{F}') < \nu(\mathcal{F})$. Then from the discussions preceding the lemma, the class $\mathcal{M}(\nu_1(\mathcal{F}'), \nu_2(\mathcal{F}'), \ldots, \nu_m(\mathcal{F}'))$ is feasible. This contradicts the minimality of the weight of the class $\mathcal{M}(i_1, i_2, \ldots, i_m)$, as $\sum_{i=1}^{m} \nu_i(\mathcal{F}') = \nu(\mathcal{F}') < \nu(\mathcal{F}) \leq \sum_{k=1}^{m} i_k$. □

Note that as the number of elements in any class $\mathcal{M}(i_1, i_2, \ldots, i_m)$, depends only on m, from Lemma 3 we can determine whether a class $\mathcal{M}(i_1, i_2, \ldots, i_m)$

is feasible or not in $O(n)$ time. Therefore, to obtain the optimal placement by P1, it suffices to find the minimum feasible class $\mathcal{M}(i_1, i_2, \ldots, i_m)$.

Using the results developed in this section, we now propose a technique for finding the minimum feasible class. We begin by studying the case where the players place two facilities each, that is, $m = 2$. We then extend this to the general case where the two players each place m facilities simultaneously.

4.1 $m = 2$

Construct the $n \times n$ symmetric matrix $\boldsymbol{A} = ((a_{ij}))$ as follows:

$$a_{ij} = \begin{cases} i + j & \text{if the class } \mathcal{M}(i,j) \text{ is feasible} \\ \infty & \text{otherwise} \end{cases}$$

Suppose $n = 2p$ is even. Then it is possible to partition $\boldsymbol{A}$ into 4 submatrices $\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{A}_3, \boldsymbol{A}_4$ each of which are of order $p \times p$ as follows:

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 \\ \boldsymbol{A}_3 & \boldsymbol{A}_4 \end{pmatrix}$$

We say that the element a_{ij} is feasible if the class $\mathcal{M}(i,j)$ is feasible. Now, our objective is to find the minimum feasible element in the matrix $\boldsymbol{A}$. We have the following two cases depending upon whether the a_{pp} is feasible or not:

Case 1: a_{pp} is feasible. In this case, any feasible element a_{ij} of $\boldsymbol{A}_4$ has more weight than the weight of a_{pp}. Therefore, we report the weight of class $\mathcal{M}(p,p)$, discard the submatrix $\boldsymbol{A}_4$.

Case 2: a_{pp} is infeasible. Then any element a_{ij} of $\boldsymbol{A}_1$ will be not be feasible (follows from Observation 3). Therefore, the submatrix $\boldsymbol{A}_1$ can be discarded.

In both of the above two cases, one of the 4 submatrices can be discarded. Applying this technique to the remaining 3 of the 4 submatrices and proceeding recursively, we can find the minimum weight feasible class. The same technique can be applied to the case when $n = 2p + 1$ is odd. Let $T(n)$ denote the time complexity of determining the minimum weight feasible class in the $n \times n$ matrix $\boldsymbol{A}$. As it takes $O(n)$ time to determine whether a class $\mathcal{M}(i,j)$ is feasible or not, the time $T(n)$, required to obtain the minimum weight feasible class, satisfies the following recurrence equation: $T(n) = 3T(\lceil n/2 \rceil) + O(n)$. This solves to $T(n) = O(n^{1.59})$, and we have the following theorem:

Theorem 3 *The optimal placement by P1, in the game $G(2, n)$ can be obtained in $O(n^{1.59})$ time.* □

4.2 Extending to the general case

The arguments described in the previous subsection can be extended to the general case, where each player places m facilities. We construct the m-dimensional $n \times n \times \ldots \times n$ symmetric matrix $\boldsymbol{A} = ((a_{i_1 i_2 \ldots i_m}))$ as follows:

$$a_{i_1 i_2 \ldots i_m} = \begin{cases} \sum_{i=1}^{m} i_k \text{ if the class } \mathcal{M}(i_1, i_2, \ldots, i_m) \text{ is feasible} \\ \infty \quad \text{otherwise} \end{cases}$$

As before, we can subdivide $\boldsymbol{A}$ into 2^m submatrices, and depending upon whether the class $\mathcal{M}(\lceil n/2 \rceil, \lceil n/2 \rceil, \ldots, \lceil n/2 \rceil)$ is feasible or not, one of these submatrices can be discarded at each stage. Let $T(n)$ denote the time complexity of determining the minimum weight feasible class in the matrix $\boldsymbol{A}$. As it takes $O(n)$ time to determine whether a class $\mathcal{M}(i_1, i_2, \ldots, i_m)$ is feasible or not, $T(n)$ satisfies the following recurrence equation: $T(n) = (2^m - 1)T(\lceil n/2 \rceil) + O(n)$. This solves to $T(n) = O(n^{m-\lambda_m})$, where $\lambda_m = m - \log_2(2^m - 1) \in (0, 1)$ is a constant that depends only on m. Hence, we have the following theorem:

Theorem 4 *The optimal placement by P1, in the $G(m, n)$ game, with $m \geq 2$, can be obtained in $O(n^{m-\lambda_m})$ time, where $0 < \lambda_m < 1$, is a constant depending only on m.* □

Remark 3: The One Round Discrete Voronoi Game with m facilities can be easily generalized to the One Round (m_1, m_2) Discrete Voronoi Game, where P1 places m_1 facilities following which P2 places m_2 facilities. The algorithms presented in the preceding sections for determining the optimal strategies of P1 and P2 generalize immediately to the One Round (m_1, m_2) Discrete Voronoi Game on a line. Therefore, the optimal placement of P2, in the One Round (m_1, m_2) Discrete Voronoi Game on a line, given any placement of P1, can be obtained in $O(n)$ time, when the sorted order of the users in $\mathcal{U}$ along ℓ is known. Similarly, the optimal placement of P1, in the One Round (m_1, m_2) Discrete Voronoi Game on a line can be obtained in $O(n^{m_2-\lambda_{m_2}})$ time, where $0 < \lambda_{m_2} < 1$, is a constant depending only on m_2.

5 Conclusions

In this paper, we study the optimal strategies for the one-round discrete Voronoi game, when the users and facilities are restricted to lie on a line. The game consists of a discrete users set $\mathcal{U}$, with $|\mathcal{U}| = n$, and two players P1 and P2 having m facilities each. The objective of both the players is to maximize their respective payoffs. We have showed that if the sorted order of the users in $\mathcal{U}$ along the line is known, then the optimal strategy of P2, given any placement of P1 can be found out in $O(n)$ time. Next, we gave a $O(n^{m+1})$ time algorithm for obtaining the optimal strategy of P1, which we subsequently improve to an $O(n^{m-\lambda_m})$ time algorithm, where $0 < \lambda_m < 1$, is a constant depending only on m.

Analogous to the continuous demand case, there are several other problems that arise in the context of discrete Voronoi games. Determining the optimal strategies in the n-round discrete Voronoi game is an open problem. This appears to be quite difficult, even when the users are assumed to be located on a line. Considering the generalizations of both the one-round and the n-round discrete games in $\mathbb{R}^2$ is also an interesting area to study.

Acknowledgement: The authors wish to thank anonymous referees for valuable comments which improved the presentation of the paper.

References

1. H.-K. Ahn, S.-W. Cheng, O. Cheong, M. Golin, and R. van Oostrum, Competitive facility location: the Voronoi game, *Theor. Comput. Sci.*, Vol. 310, 457–467, 2004.
2. B. B. Bhattacharya, Maximizing Voronoi regions of a set of points enclosed in a circle with applications to facility location, *J. Math. Model. Algor.*, Vol. 9 (4), 375–392, 2010.
3. B. B. Bhattacharya and S. C. Nandy, New variations of the reverse facility location problem, *Proc. 22nd Canadian Conference on Computational Geometry*, 241–244, 2010.
4. S. Cabello, J. Miguel Diáz-Báñez, S. Langerman, C. Seara, and I. Ventura, Facility location problems in the plane based on reverse nearest neighbor queries, *Eur. J. Operations Research*, Vol. 202(1), 99–106, 2010.
5. O. Cheong, A. Efrat, and S. Har-Peled, On finding a guard that sees most and a shop that sells most, *Discrete Computational Geometry*, Vol. 37, 545–563, 2007.
6. O. Cheong, S. Har-Peled, N. Linial, and J. Matouŝek, The one-round Voronoi game, *Discrete and Computational Geometry*, Vol. 31(1), 125–138, 2004.
7. F. Dehne, R. Klein, and R. Seidel, Maximizing a Voronoi region: the convex case, *Int. Journal of Computational Geometry and Applications*, Vol. 15, 463-475, 2005.
8. Z. Drezner and H. W. Hamacher (Eds.), *Facility Location: Applications and Theory*, Springer, 2002.
9. H. A. Eiselt and G. Laporte, Competitive spatial models, *Eur. J. Operations Research*, Vol. 39, 231–242, 1989.
10. H. A. Eiselt, G. Laporte, and J. F. Thisse, Competitive location models: A framework and bibliography, *Transportation Science*, Vol. 27, 44–54, 1993.
11. S. P. Fekete and H. Meijer, The one-round Voronoi game replayed, *Comput. Geom. Theory Appl.*, Vol. 30 (2), 81–94, 2005.
12. J. Spoerhase and H.-C. Wirth, (r,p)-centroid problems on paths and trees, *Theor. Comput. Sci.* Vol. 410 (47-49), 5128–5137, 2009.
13. R. Tobin, T. Friesz, and T. Miller, Existence theory for spatially competitive network facility location models, *Annals of Operations Research*, Vol. 18, 267–276, 1989.