Hitting Geometric Objects Online via Points in $\mathbb{Z}^{d}$ *

Minati De[1] [0000−0002−1859−8800] and Satyam Singh[1]

Dept. of Mathematics,
Indian Institute of Technology Delhi,
New Delhi, India
{minati,satyam.singh}@maths.iitd.ac.in

Abstract. In this paper, we consider the online version of the minimum hitting set problem, where geometric objects arrive one by one. The online algorithm must maintain a hitting set for the arrived objects by making irrevocable decisions. Here the centers of objects and hitting points are in $\mathbb{R}^d$ and $\mathbb{Z}^d$, respectively. First, we show that for hitting unit intervals using points from $\mathbb{Z}$, one can achieve a tight bound of 2. Then, we consider the case of d-dimensional unit hypercubes. At first, we prove that every deterministic online algorithm for hitting unit hypercubes has a competitive ratio of at least $d+1$ when $d \in \mathbb{N}$. Later, for $d = 2$, we propose a deterministic online algorithm with a competitive ratio of at most 4 and for $d > 2$, we propose a randomized algorithm having a competitive ratio of $O(d^2)$. Next, we consider the case of unit balls in $\mathbb{R}^d$. We show that every deterministic online algorithm for hitting balls has a competitive ratio of at least $d+1$ when $d < 4$. Then, we propose a deterministic algorithm having a competitive ratio of at most $O(d^4)$ and $O(1)$, for $d \geq 4$ and $d = 3$, respectively. At last, for unit disks in $\mathbb{R}^2$, we propose a simple deterministic algorithm that achieves a competitive ratio of at most 4.

Keywords: Hitting set · Online algorithm · Competitive ratio · Unit covering · Geometric objects.

1 Introduction

The minimum hitting set problem is one of the most important problems in computational geometry, combinatorial optimization and other areas [1,4,5,6,7,9]. For a given set $\mathcal{S}$ of objects in $\mathbb{R}^d$ and a set $\mathcal{P}$ of points in $\mathbb{Z}^d$, a subset $\mathcal{H} \subseteq \mathcal{P}$ is said to be a *hitting set* for the set $\mathcal{S}$, if each object of $\mathcal{S}$ contains at least one point of $\mathcal{H}$. The objective of the minimum hitting set problem is to find a hitting set $\mathcal{H}$ of minimum cardinality. This problem has applications in wireless networks, VLSI design, resource allocation, etc.

* Work on this paper by the first author has been partially supported by SERB MATRICS Grant MTR/2021/000584, and work by the second author has been supported by CSIR (File Number-09/086(1429)/2019-EMR_I).

In this paper, we consider this problem in an online setup, in which geometric objects of unit size arrive one by one, and the algorithm must maintain a valid hitting set for the already arrived objects. On the arrival of a new object $\sigma \subset \mathbb{R}^d$, if σ does not contain any point from the existing hitting set, the algorithm needs to add a point $p \in \mathbb{Z}^d$ to the existing hitting set to hit the new object σ. The decision to add a point is irrevocable, i.e., the online algorithm can not remove any point in the future from the existing hitting set. The aim is to minimize the cardinality of the hitting set.

By interchanging the role of objects and points, one can formulate an equivalent *online unit covering problem*. Here, the points belonging to $\mathbb{R}^d$ are arriving one by one, and we need to cover each point with a unit object whose center belongs to $\mathbb{Z}^d$. We aim to minimize the number of unit objects to cover all the presented points. One application of this problem is as follows. Let us consider a planned city where one can install base stations at specific locations from a rectilinear grid. Each base station can cover clients who are within its covering radius. Clients can appear from any location in the city. The objective is to minimize the number of base stations that can cover all clients. Since installing a base station is expensive, the decision is considered irrevocable. The need is to place a base station that serves a new uncovered client.

We use competitive analysis to analyze the quality of our online algorithm [2]. Let $\mathcal{A}$ be an online algorithm for the minimization problem, the algorithm $\mathcal{A}$ is said to be *c-competitive*, if $c = \sup_\beta \frac{\mathcal{A}_\beta}{\mathcal{O}_\beta}$, where $\mathcal{A}_\beta$ and $\mathcal{O}_\beta$ are the costs of the solution produced by the online algorithm $\mathcal{A}$ and an optimal offline algorithm, respectively, with respect to an input sequence β. For randomized algorithm, $\mathcal{A}_\beta$ is replaced by the expectation $\mathbb{E}[\mathcal{A}_\beta]$, and the competitive ratio of the algorithm $\mathcal{A}$ is $\sup_\beta \frac{\mathbb{E}[\mathcal{A}_\beta]}{\mathcal{O}_\beta}$ [5].

1.1 Related Work

Consider the set cover problem. Let $\mathcal{P}$ be a set of n elements, and let $\mathcal{S}$ be a family of subsets of $\mathcal{P}$ such that $|\mathcal{S}| = m$. A *set cover* is a collection of subsets $\mathcal{S}^* \subset \mathcal{S}$ such that their union covers the whole set $\mathcal{P}$. The goal of the *set cover problem* is to find a set cover of minimum cardinality. Note that by interchanging the roles of subsets and points, the (online) set cover problem and the (online) hitting set problems are equivalent. Both problems are classical NP-hard problems [10]. In the offline setup, if the set $\mathcal{P}$ contains points on the real line and $\mathcal{S}$ consists of intervals in $\mathbb{R}$, the minimum hitting set problem can be solved in polynomial time using a greedy algorithm. However, these problems remain NP-hard, even for simple geometric objects like unit disks or unit squares in $\mathbb{R}^2$ [8]. Alon et al. [1] considered the online set cover problem. In their setting, the sets $\mathcal{P}$ and $\mathcal{S}$ are already known, but the order of arrivals of points is unknown. Upon the arrival of an uncovered point, the online algorithm must choose a subset that covers that point. The online algorithm presented by Alon et al. [1] achieves a competitive ratio of $O(\log n \log m)$. The results obtained in [1] also hold for the online hitting set problem. Later, Even and Smorodinsky [6] studied the hitting set problem in an online setup where the set $\mathcal{S}$ and $\mathcal{P}$ are known in advance, but the order of arrival of the input objects is unknown. In this setup, they proposed an algorithm with a competitive ratio of $O(\log n)$ for half-planes and unit disks. They gave a matching lower bound of the competitive ratio for these cases. They also proposed an online algorithm that achieves a tight bound of $\Theta(\log n)$ when objects are intervals and points are integer. In this paper, we restrict our attention when the set of objects are unit objects in $\mathbb{R}^d$ and points are in $\mathbb{Z}^d$. But, we don't know the set of objects in advance.

A closely related problem is the unit covering problem (a variant of the set cover problem), where the objective is to cover a given set of n points in $\mathbb{R}^d$ with the minimum number of translated copies of a unit object. Charikar et al. [3] studied the problem in the online setting, where the points (in $\mathbb{R}^d$) arrive one by one at the online algorithm. The objective is to assign a newly arrived point to an existing unit ball in $\mathbb{R}^d$ or a new unit ball to cover it. They proposed an online algorithm with a competitive ratio of $O(2^d d \log d)$ and showed that the lower bound of any deterministic online algorithm for the problem is $\Omega(\log d/\log\log\log d)$. Dumitrescu et al. [4] considered the online set cover problem in the same model of computation as of [3]. They improved both the upper and lower bound of the problem to $O(1.321^d)$ and $\Omega(d+1)$ for unit balls in $\mathbb{R}^d$, respectively [4]. They also proved that for every centrally symmetric convex body $C \in \mathbb{R}^d$, the competitive ratio of every deterministic online algorithm for the unit covering problem in $\mathbb{R}^d$ under the L_C norm is at least $I(C)$, where $I(C)$ is the illumination number of the convex body C (for definition, see [4]). Recently, Dumitrescu and Tóth [5] studied the problem in the online setting for integer hypercubes (defined in Section 1.2) under L_∞ norm in $\mathbb{R}^d$, where the points in $\mathbb{Z}^d$ arrive one by one to the online algorithm. The objective is to cover the newly arrived point using an existing or new integer hypercube. Dumitrescu and Tóth [5] proved that the competitive ratio of every deterministic online algorithm for covering in $\mathbb{Z}^d$ under the L_∞ norm is at least $d+1$ for every $d \geq 1$. They also proved that the competitive ratio of their algorithm for unit covering in $\mathbb{Z}^d$ under the L_∞ norm is $O(d^2)$ for every $d \in \mathbb{N}$. All the results obtained here also hold for the online hitting set problem, where the center of objects comes from $\mathbb{Z}^d$ and the set of points that can be used for hitting $\mathcal{P} \subseteq \mathbb{R}^d$. To complement their result, in this paper, we consider the case, when the centers of unit objects are coming one by one from $\mathbb{R}^d$ and the points that can be used for hitting are from $\mathbb{Z}^d$.

1.2 Notation and Preliminaries

We use $[n]$ to denote the set $\{1, 2, \ldots, n\}$. We use the term *object* to denote a simply connected compact set in $\mathbb{R}^d$ with a nonempty interior. An *integer point* is a point $z = (z_1, z_2, \ldots, z_d) \in \mathbb{R}^d$ such that each coordinate z_i is integer, for $i \in [d]$. We use $\mathcal{Q}(\sigma)$ to denote the set of integer points contained in an object σ. The term *integer hypercube* refers to a hypercube $H \subset \mathbb{R}^d$ of side length one with all corners as integer points. A *unit hypercube* $H_d(c, 1) \in \mathbb{R}^d$ centring at c, is defined as $H_d(c, 1) = \{x \in \mathbb{R}^d : dist_\infty(x, c) \leq 1\}$. Here $dist_\infty(x, y)$ is the

L_∞ distance between x and y. Note that, according to our definition, an integer hypercube is not a unit hypercube. A *unit ball* $B_d(c,1) \in \mathbb{R}^d$ centring at c, is defined as $B_d(c,1) = \{x \in \mathbb{R}^d : dist(x,c) \leq 1\}$. Here $dist(x,y)$ is the Euclidean distance between x and y.

1.3 Our Contributions

First, for hitting unit intervals using points in $\mathbb{Z}$, we propose a deterministic online algorithm that achieves a tight bound of 2. Next, we consider the case of d-dimensional unit hypercubes. At first, we prove that the competitive ratio of every deterministic online algorithm for hitting unit hypercubes in $\mathbb{R}^d$ using points in $\mathbb{Z}^d$ is at least $d+1$, where $d \in \mathbb{N}$. Later, we propose a simple deterministic algorithm that achieves a competitive ratio of at most 4 when $d = 2$. For $d > 2$, we propose a randomized algorithm having a competitive ratio of at most $O(d^2)$. Finally, we consider the case of unit balls in $\mathbb{R}^d$. In this case, initially, we prove that the competitive ratio of every deterministic online algorithm for hitting unit balls in $\mathbb{R}^d$ using points in $\mathbb{Z}^d$ is at least $d+1$, for $d < 4$. Then, we propose a deterministic algorithm having a competitive ratio of at most $O(d^4)$ and $O(1)$, for $d \geq 4$ and $d = 3$, respectively. At last, for unit disks in $\mathbb{R}^2$, we propose a simple deterministic algorithm that achieves a competitive ratio of 4. All the above-mentioned results are also valid for the equivalent unit covering problem where the points $p \in \mathbb{R}^d$ are coming one by one, and we need to cover each point with a unit object centered at some integer point $c \in \mathbb{Z}^d$.

2 Hypercubes in $\mathbb{R}^d$

2.1 Lower Bound

Theorem 1. *The competitive ratio of every deterministic online algorithm for hitting unit hypercubes in $\mathbb{R}^d$ using points in $\mathbb{Z}^d$ is at least $d+1$, where $d \in \mathbb{N}$.*

Proof. Let us consider a game between two players: Alice and Bob. Here, Alice plays the role of an adversary, and Bob plays the role of an online algorithm. In each round of the game, Alice presents a new unit hypercube σ in $\mathbb{R}^d$ such that Bob needs to hit it by a new hitting point $h \in \mathbb{Z}^d$. To prove the lower bound of the competitive ratio, we show by induction that Alice can present a sequence of unit hypercubes $\sigma_1, \sigma_2, \ldots, \sigma_{d+1} \subset \mathbb{R}^d$ adaptively depending on the position of hitting points placed by Bob such that Bob needs to place $d+1$ integer points $\{h_1, h_2, \ldots, h_{d+1}\}$; whereas the offline optimum needs just one integer point. Let σ_1 be a hypercube presented by Alice in the first round of the game. For the sake of simplicity, we assume that the center of σ_1 is the origin. For $i = 1, \ldots, d+1$, we maintain the following two invariants:

- The hypercube $\sigma_i \subset \mathbb{R}^d$ does not contain any of the previously placed hitting point $h_j \in \mathbb{Z}^d$, for $j < i$.

– The common intersection region $Q_i = \cap_{j=1}^{i}\sigma_j$ contains $3^{(d-i+1)}$ integer points.

For $i = 1$, the first invariant trivially holds. Since the unit hypercube σ_1 is centered at the origin, each coordinate of any integer point $p \in \sigma_1$ has three possible values from $\{-1, 0, 1\}$. As a result, the unit hypercube σ_1 contains 3^d integer points. Thus, the second invariant also holds.

At the beginning of the round i (for $i = 2, \ldots, d$), assume that both invariants hold. Let us define a translation vector $\mathbf{v}_i \in \mathbb{R}^d$ as follows: $\mathbf{v}_i = (s(1)(1 + \epsilon), s(2)(1 + \epsilon), \ldots, s(i-1)(1 + \epsilon), 0, \ldots, 0)$, where $0 < \epsilon < \frac{1}{2}$ is an arbitrary constant, and for any $j < i$, we have

$$s(j) = \begin{cases} +1, & \text{if } h_j(x_j) \leq 0, \text{ where } h_j(x_j) \text{ is } jth \text{ coordinate of } h_j, \\ -1, & \text{otherwise.} \end{cases}$$

We define the hypercube $\sigma_i = \sigma_1 + \mathbf{v}_i$. For any $j < i$, due to the definition of the jth component of the translation vector $\mathbf{v}_i$, the hypercube σ_i does not contain the point h_j. Hence, the first invariant is maintained. Let us count the number of integer points contained in $\sigma_1 \cap \sigma_i$. Consider any integer point $p \in \sigma_1 \cap \sigma_i$. Note that σ_i is centered at $\mathbf{v}_i$ since $\sigma_i = \sigma_1 + \mathbf{v}_i$ and σ_1 is centered at the origin. As a result, for any $j \in [i-1]$, the jth coordinate of p is fixed at $s(j)$. The value of each of the remaining $(d - i + 1)$ coordinates of p has three possibilities from $\{-1, 0, 1\}$. Therefore, $\sigma_1 \cap \sigma_i$ contains $3^{(d-i+1)}$ integer points. Because of the above argument, observe that all the integer points that belong to $\sigma_1 \cap \sigma_i$ is also contained in $\sigma_1 \cap \sigma_j$, where $j < i$. Hence, Q_i contains $3^{(d-i+1)}$ integer points. □

2.2 Upper Bound

Structural Properties: We first present some concepts that we utilize to analyze the algorithm that we propose for unit hypercubes in $\mathbb{R}^d$, where $d \geq 3$. Let $\mathcal{F}$ be the family of all possible unit hypercubes in $\mathbb{R}^d$. Any pair of unit hypercubes σ_i and σ_j in $\mathcal{F}$ are said to be *related* if $\mathcal{Q}(\sigma_i) = \mathcal{Q}(\sigma_j)$, in other words, each of them contains the same set of integer points. So, we have an equivalence relation on $\mathcal{F}$ where each *equivalence class* corresponds to a set S of unit hypercubes such that each $\sigma \in S$ contains the same set of integer points.

Let σ be a unit hypercube centered at c. Partition $[d]$ into two sets $\mathcal{K}_1$ and $\mathcal{K}_2$ such that for each $i \in \mathcal{K}_1$, the value of $c(x_i)$, the ith coordinate of c is non-integer and for each $i \in \mathcal{K}_2$, the value of $c(x_i)$ is integer. Let $r \in \mathcal{Q}(\sigma)$ be an integer point. The value of $r(x_i)$, can be any one from the two possible values: $\{\lfloor c(x_i) \rfloor, \lceil c(x_i) \rceil\}$ when $i \in \mathcal{K}_1$, and it has exactly three possibilities from $\{c(x_i) - 1, c(x_i), c(x_i) + 1\}$ when $i \in \mathcal{K}_2$. Hence, $\mathcal{Q}(\sigma)$ contains exactly $2^{|\mathcal{K}_1|}3^{|\mathcal{K}_2|}$ integer points. The following lemma is an important ingredient for the classification of the equivalence classes.

Lemma 1. *Two unit hypercubes σ_1 and σ_2 centered at c_1 and c_2, respectively, contains the same set of integer points if and only if $[d]$ can be partitioned into two sets $\mathcal{K}_1$ and $\mathcal{K}_2$ such that*

– *for each $i \in \mathcal{K}_1$, the value of the ith coordinate of c_1 and c_2 is non-integer and $\lfloor c_1(x_i) \rfloor = \lfloor c_2(x_i) \rfloor$,*
– *for each $i \in \mathcal{K}_2$, the integer value of the ith coordinate of c_1 and c_2 is same, i.e., $c_1(x_i) = c_2(x_i)$.*

We have $d+1$ types of equivalence classes depending on the number of integer points they cover. We refer to an equivalence class that contains exactly $2^k 3^{d-k}$ integer points as *an equivalence class of Type-(k)*, where $k \in [d] \cup \{0\}$. By a careful observation, one can note the following.

Lemma 2. *Let σ be a unit hypercube in $\mathbb{R}^d$, centered at c, belonging to some equivalence class of Type-(k). There exists a set $\mathbb{S}_\sigma$ of distinct $2^{(d-k)}$ equivalence classes of Type-(d) such that $\mathcal{Q}(\sigma) = \cup_{\sigma' \in \mathbb{S}_\sigma} \mathcal{Q}(\sigma')$, where $k \in [d-1] \cup \{0\}$.*

The following lemma plays an important role to analyse the algorithm that we describe next.

Lemma 3. *Each integer point $p \in \mathbb{Z}^d$ is contained in exactly 2^d distinct equivalence classes of Type-(d), $d \in \mathbb{N}$.*

To obtain the upper bound, for $d \geq 3$, we propose an $O(d^2)$ competitive randomized iterative reweighting algorithm that is similar in nature to an algorithm from [5]. It was presented for covering integer points using integer hypercubes in the online setup.

Algorithm: Let $\mathcal{I} \subset \mathbb{R}^d$ be the set of unit hypercubes presented to the algorithm and $\mathcal{A} \subset \mathbb{Z}^d$ be the set of points chosen by our algorithm such that each unit hypercube in $\mathcal{I}$ contains at least one point from $\mathcal{A}$. The algorithm maintains two disjoint sets $\mathcal{A}_1$ and $\mathcal{A}_2$ such that $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2$. The algorithm also maintains another set $\mathcal{B}$ of points for bookkeeping purposes; initially, each of the set $\mathcal{I}, \mathcal{A}$ and $\mathcal{B}$ are empty. A weight function w over all integer points is also maintained by the algorithm; initially, $w(p) = 3^{-(d+1)}$, for all points $p \in \mathbb{Z}^d$. One iteration of the algorithm is described below.

Let σ be a new unit hypercube; update $\mathcal{I} = \mathcal{I} \cup \{\sigma\}$. Note that $|\mathcal{Q}(\sigma)|$ is at least 2^d and at most 3^d.

1. If the unit hypercube σ contains any point from $\mathcal{A}$, then do nothing.
2. Else if the unit hypercube σ contains any point from $\mathcal{B}$, then let $p \in \mathcal{B} \cap \mathcal{Q}(\sigma)$ be an arbitrary point, and update $\mathcal{A}_1 = \mathcal{A}_1 \cup \{p\}$.
3. Else if $\sum_{p \in \mathcal{Q}(\sigma)} w(p) \geq 1$, then let p be an arbitrary point in $\mathcal{Q}(\sigma)$, and update $\mathcal{A}_2 = \mathcal{A}_2 \cup \{p\}$.
4. Else, the weights give a probability distribution on $\mathcal{Q}(\sigma)$. Successively choose points from $\mathcal{Q}(\sigma)$ at random with this distribution in $\lceil \frac{5d}{2} \rceil$ independent trails and add them to $\mathcal{B}$. Let $p \in \mathcal{B} \cap \mathcal{Q}(\sigma)$ be an arbitrary point, and update $\mathcal{A}_1 = \mathcal{A}_1 \cup \{p\}$. Triple the weight of every point in $\mathcal{Q}(\sigma)$.

Now, we analyze the performance of the above algorithm.

Theorem 2. *For hitting unit hypercubes using points in $\mathbb{Z}^d$, there exists a randomized algorithm whose competitive ratio is at most $O(d^2)$, where $d \geq 3$.*

Proof. Let $\mathcal{I}$ be the set of n unit hypercubes presented to our algorithm. Let $\mathcal{O}$ be an offline optimum hitting set for $\mathcal{I}$. Note that our algorithm creates two disjoint sets $\mathcal{A}_1$ and $\mathcal{A}_2$ such that $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2$ is a hitting set for $\mathcal{I}$. From the description of the algorithm, it is easy to follow that $\mathcal{A}_1 \subseteq \mathcal{B}$. We prove that $\mathbb{E}[|\mathcal{B}|] = O(d^2|\mathcal{O}|)$, and $\mathbb{E}[|\mathcal{A}_2|] = O(|\mathcal{O}|)$. This immediately implies that $\mathbb{E}[|\mathcal{A}|] \leq \mathbb{E}[|\mathcal{A}_1|] + \mathbb{E}[|\mathcal{A}_2|] \leq \mathbb{E}[|\mathcal{B}|] + \mathbb{E}[|\mathcal{A}_2|] = O(d^2|\mathcal{O}|)$.

First, consider $\mathbb{E}[|\mathcal{B}|]$. Note that in the set $\mathcal{B}$, new points are added only in step 4 of the algorithm. In this case, the algorithm adds at most $\lceil \frac{5d}{2} \rceil$ points (independently) in $\mathcal{B}$ and triples the weight of every point in $\mathcal{Q}(\sigma)$. Let $\mathcal{O}$ denote the offline optimum set of integer points. Each hypercube $\sigma \in \mathcal{I}$ contains some point $p \in \mathcal{O}$. Initially, the weight of p is $3^{-(d+1)}$, and it will never exceed 3. Since $p \in \mathcal{Q}(\sigma)$, its weight before the last tripling must have been at most 1 in step 4 of the algorithm; thus, its weight is tripled in at most $d+2$ iterations. Consequently, the algorithm invokes step 4 of the algorithm in at most $(d+2)|\mathcal{O}|$ iterations. In each such iteration, the algorithm adds at most $\lceil \frac{5d}{2} \rceil$ points (independently) in the set $\mathcal{B}$. Therefore, we have $|\mathcal{B}| \leq \lceil \frac{5d}{2} \rceil (d+2)|\mathcal{O}| = O(d^2|\mathcal{O}|)$.

Next, we consider $\mathbb{E}[|\mathcal{A}_2|]$. Note that in the set $\mathcal{A}_2$, new points are added only in step 3 of the algorithm. In this case, when a hypercube σ arrives, none of the points of $\mathcal{Q}(\sigma)$ is in $\mathcal{B}$ and $\sum_{p \in \mathcal{Q}(\sigma)} w(p) \geq 1$, and the algorithm increments the cardinality of the set $\mathcal{A}_2$ by one. At the beginning of the algorithm, we have $W_{initially} = \sum_{p \in \mathcal{Q}(\sigma)} w(p) = \sum_{p \in \mathcal{Q}(\sigma)} 3^{-(d+1)} \leq 3^d 3^{-(d+1)} = \frac{1}{3}$. Suppose that the weights of the points in $\mathcal{Q}(\sigma)$ is increased in k iterations (starting from the beginning of the algorithm), and the sum of weights of points in $\mathcal{Q}(\sigma)$ is increased by $\delta_1, \delta_2, \ldots, \delta_k > 0$. When σ arrives, the sum of the weights of all the points in $\mathcal{Q}(\sigma)$ is $W_{now} = W_{initially} + \sum_{i=1}^{k} \delta_i \geq 1$ and we know $W_{initially} \leq \frac{1}{3}$. This implies that $\sum_{i=1}^{k} \delta_i \geq \frac{2}{3}$. For every $i \in [k]$, the sum of weights of some points in $\mathcal{Q}(\sigma)$, say $Q_i \subset \mathcal{Q}(\sigma)$ is increased by δ_i in step 4 of the algorithm. Since the weights are tripled, therefore, the sum of weights of these points was $\frac{\delta_i}{2}$ at the beginning of that iteration. The algorithm added a point from Q_i to $\mathcal{B}$ with probability at least $\frac{\delta_i}{2}$ in one random draw, which was repeated $\lceil \frac{5d}{2} \rceil$ times independently. As a result, the probability that the algorithm does not add any point from Q_i to the set $\mathcal{B}$ is at most $\left(1 - \frac{\delta_i}{2}\right)^{\lceil \frac{5d}{2} \rceil}$. The probability that none of the points of $\mathcal{Q}(\sigma)$ are added to $\mathcal{B}$ before the arrival of σ is at most $\prod_{i=1}^{k} \left(1 - \frac{\delta_i}{2}\right)^{\lceil \frac{5d}{2} \rceil} \leq e^{-\lceil \frac{5d}{2} \rceil \sum_{i=1}^{k} \frac{\delta_i}{2}} \leq e^{-\frac{5d}{4} \sum_{i=1}^{k} \delta_i} \leq e^{-\frac{5d}{6}}$. Since $\mathcal{I}$ is the set of hypercubes presented to the algorithm, therefore, step 3 of the algorithm can be invoked at most $|I|$ times $e^{-\frac{5d}{6}}$. As a result $\mathbb{E}[|\mathcal{A}_2|] \leq |\mathcal{I}| e^{-\frac{5d}{6}}$. Note that this is a very loose upper bound. Let N be the set of distinct equivalence classes containing all the hypercubes in $\mathcal{I}$. Observe that if the algorithm hits one hypercube from an equivalence class, then the algorithm executes only step 1 for all subsequent hypercubes coming from the same equivalence class. Therefore, step 3 of the algorithm can be invoked at most $|N| e^{-\frac{5d}{6}}$ times. We can further improve this bound as follows.

Let $\sigma \in \mathcal{I}$. According to Lemma 2, we have a set $\mathbb{S}_\sigma$ of equivalence classes of Type-(d) such that $\mathcal{Q}(\sigma) = \cup_{\sigma' \in \mathbb{S}_\sigma} \mathcal{Q}(\sigma')$. Observe that if some hypercube σ

arrives, and our algorithm needs to place a hitting point p for it, then it implies that none of the hypercubes belonging to $\mathbb{S}_\sigma$ arrived before σ to the algorithm. Let $p \in \mathcal{Q}(\sigma')$ for some $\sigma' \in \mathbb{S}_\sigma$. Note that the point p acts as a hitting point for any hypercube in $\mathcal{I}$ belonging to the same class of σ'. Not only that but p also acts as a hitting point for all hypercubes $\sigma'' \in \mathcal{I}$ such that $\sigma' \in \mathbb{S}_{\sigma''}$. Therefore, step 3 of the algorithm is invoked at most $|N_d|e^{-\frac{5d}{6}}$ times, where $N_d = \cup_{\sigma \in \mathcal{I}} \mathbb{S}_\sigma$. Hence, $\mathbb{E}[|\mathcal{A}_2|] \leq |N_d|e^{-\frac{5d}{6}}$. Now we give an upper bound of $|N_d|$ in terms of $|\mathcal{O}|$. Due to Lemma 3, we know that any arbitrary integer point $p \in \mathcal{O}$ can be contained in at most 2^d equivalence classes of Type-(d) hypercubes. Thus we have $|N_d| \leq 2^d|\mathcal{O}|$). Since we have, $\mathbb{E}[|\mathcal{A}_2|] \leq |N_d|e^{-\frac{5d}{6}}$, and $|N_d| \leq 2^d|\mathcal{O}|$, therefore $\mathbb{E}[|\mathcal{A}_2|] \leq O\left(\left(\frac{2}{e^{\frac{5}{6}}}\right)^d |\mathcal{O}|\right) \leq |\mathcal{O}|$. Hence the theorem follows. □

For $d = 1$ and 2, unit hypercube becomes unit interval and square, respectively. For these special cases, we have the following result.

Theorem 3. *For hitting unit intervals and unit squares using points in $\mathbb{Z}$ and $\mathbb{Z}^2$, respectively, there exists a deterministic online algorithm whose competitive ratio is at most* 2 *and* 4*, respectively.*

3 Balls in $\mathbb{R}^d$

3.1 Lower Bound for Balls in $\mathbb{R}^d$, for $d < 4$

To obtain a lower bound of the competitive ratio, we think of a game between two players: Alice and Bob. Here, Alice plays the role of the adversary, and Bob plays the role of the online algorithm. In each round of the game, Alice presents a unit ball such that Bob needs to place a new hitting point. We show that Alice can present an input sequence of balls $\sigma_1, \sigma_2, \ldots, \sigma_{d+1} \subset \mathbb{R}^d$, centered at $c_1, c_2, \ldots, c_{d+1}$, respectively, depending on the position of hitting points placed by Bob, for which Bob needs to place $d+1$ integer points; while the offline optimum needs just one point. For the sake of simplicity, let us assume that the center c_1 of the first ball σ_1 coincides with the origin. Note that the ball σ_1 contains exactly $2d$ integer points $\mathcal{P} = \{p_1, p_2, \ldots, p_{2d}\}$ apart from the origin. The coordinates of these points are given below:

$$p_k(x_j) = \begin{cases} 1, & \text{if } k = j, \quad \text{for } k, j \in [d] \\ -1, & \text{if } k = d + j, \quad \text{for } k \in [2d] \setminus [d] \ \& \ j \in [d] \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Let $\mathcal{P}_1 = \{p_1, p_2, \ldots, p_d\}$ and $\mathcal{P}_2 = \{p_{d+1}, p_{d+2}, \ldots, p_{2d}\}$. To hit the input ball σ_1, Bob needs to choose a point $h_1 \in \mathcal{P}_1 \cup \mathcal{P}_2 \cup \{c_1\}$. Depending on the position of h_1, Alice presents a ball σ_2 centered at a point c_2 that satisfies the following:

$$c_2 = \begin{cases} \left(\frac{1}{2} + \epsilon_d, \frac{1}{2} + \epsilon_d, \ldots, \frac{1}{2} + \epsilon_d\right), & \text{if } h_1 \in \mathcal{P}_2 \cup \{c_1\} \\ \left(-(\frac{1}{2} + \epsilon_d), -(\frac{1}{2} + \epsilon_d), \ldots, -(\frac{1}{2} + \epsilon_d)\right), & \text{otherwise (i.e., if } h_1 \in \mathcal{P}_1), \end{cases} \tag{2}$$

where the value of ϵ_d is 0.5 and 0.15 for $d = 2$ and 3, respectively. Note that σ_2 does not contain the point c_1.

Lemma 4. *(i) If $h_1 \in \mathcal{P}_2 \cup \{c_1\}$, then $\mathcal{Q}(\sigma_2)$ contains all the points of $\mathcal{P}_1$ and it does not contain any point of $\mathcal{P}_2 \cup \{c_1\}$.*

(ii) If $h_1 \in \mathcal{P}_1$, then $\mathcal{Q}(\sigma_2)$ contains all the points of $\mathcal{P}_2$ and it does not contain any point of $\mathcal{P}_1$

From now onwards, we assume that Bob chooses $h_1 \in \mathcal{P}_2 \cup \{c_1\}$. The other case is similar in nature. Now we show by induction that Alice and Bob can play the game for next $d+1$ rounds maintaining the following two invariants: For $i = 2, \dots, d+1$, when Alice presents balls $\sigma_2, \dots, \sigma_i$ and Bob presents piercing points $p_{\pi(2)}, p_{\pi(3)} \dots, p_{\pi(i-1)} \in \mathcal{P}_1$,

(I) The ball $\sigma_i \subset \mathbb{R}^d$ does not contain any previously placed hitting point $h_j \in \mathbb{Z}^d$, for $j < i$.

(II) The ball σ_i contains all the points from $\mathcal{P}_1 \setminus \{p_{\pi(2)}, p_{\pi(3)} \dots, p_{\pi(i-1)}\}$.

Invariant (I) ensures that Bob needs a new point to hit σ_i. On the other hand, Invariant (II) ensures that $\cap \sigma_i$ contains a point from $\mathcal{P}_1$ that is not used by Bob. For $i = 2$, due to Lemma 4, both the invariants are maintained. At the beginning of the round i (for $i = 2, \dots, d$), assume that both invariants hold. Let $\Pi = \{\pi(2), \pi(3), \dots, \pi(i)\}$ be the set of indices of integer points chosen from $\mathcal{P}_1$ to hit the previously arrived balls. Depending on the position of the hitting point $p_{\pi(i)}$, Alice presents a ball σ_{i+1}, in the $(i+1)$th round of the game, centering at c_{i+1} that satisfies the following.

$$c_{i+1}(x_j) = \begin{cases} \left(\frac{3}{2}\right)^{(i-1)} c_2(x_j), & \text{for all } j \in [d] \setminus \Pi, \text{ and} \\ 0, & \text{for } j \in \Pi. \end{cases} \tag{3}$$

• First, we prove that σ_{i+1} does not contain the first hitting point h_1. Observe that $dist(c_{i+1}, h_1)^2 = \sum_{j \in [d]} \left(c_{i+1}(x_j) - h_1(x_j)\right)^2$. Note that for $j \in \Pi$, the value of $c_{i+1}(x_j)$ is zero. So we have

$$dist(c_{i+1}, h_1)^2 = \sum_{j \in \Pi} \left(0 - h_1(x_j)\right)^2 + \sum_{j \in [d] \setminus \Pi} \left(\left(\frac{3}{2}\right)^{(i-1)} \left(\frac{1}{2} + \epsilon_d\right) - h_1(x_j) \right)^2 .$$

If $h_1 = c_1$, then we have $dist(c_{i+1}, h_1)^2 = \quad 0 + (d-i+1) \left(\frac{3}{2}\right)^{2(i-1)} \left(\frac{1}{2} + \epsilon_d\right)^2 > 1$. If $h_1 = p_k \in \mathcal{P}_2$, then we have following two sub cases. If $(k-d) \in \Pi$, we have $dist(c_{i+1}, h_1)^2 = 1 + (d-i+1) \left(\frac{3}{2}\right)^{2(i-1)} \left(\frac{1}{2} + \epsilon_d\right)^2 > 1$, otherwise ($(k-d) \in [d] \setminus \Pi$), we have $dist(c_{i+1}, h_1)^2 = (d-i+1) \left(\left(\frac{3}{2}\right)^{(i-1)} \left(\frac{1}{2} + \epsilon_d\right) + 1 \right)^2 > 1$. Now, we show that σ_{i+1} does not contain any of the previously placed hitting points of $\mathcal{P}_1$. Here, for any $p_{\pi(k)} \in \{p_{\pi(2)}, p_{\pi(3)} \dots, p_{\pi(i)}\}$, we have

$$dist(c_{i+1}, p_{\pi(k)})^2 = \sum_{j \in [d]} \left(c_{i+1}(x_j) - p_{\pi(k)}(x_j)\right)^2 .$$

Note that for $j \in \Pi$, $c_{i+1}(x_j) = 0$, and $p_{\pi(k)} \in \mathcal{P}_1$ has only one nonzero coordinate that is the $\pi(k)$th coordinate with value 1 and $\pi(k) \in \Pi$. Therefore, we have

$$dist(c_{i+1}, p_{\pi(k)})^2 = \sum_{j \in \Pi} \left(0 - p_{\pi(k)}(x_j)\right)^2 + \sum_{j \in [d] \setminus \Pi} \left(\left(\frac{3}{2}\right)^{(i-1)} \left(\frac{1}{2} + \epsilon_d\right)\right)^2$$
$$= 1 + (d - i + 1) \left(\frac{3}{2}\right)^{2(i-1)} \left(\frac{1}{2} + \epsilon_d\right)^2 > 1.$$

Therefore, the distance between the center c_{i+1} and previously placed hitting points $\{p_{\pi(2)}, p_{\pi(3)} \ldots, p_{\pi(i)}\}$ is greater than one. Hence the invariant (I) holds.

• Now, we show that σ_{i+1} contains all $(d - i + 1)$ integer points from $\mathcal{P}_1 \setminus \{p_{\pi(2)}, p_{\pi(3)} \ldots, p_{\pi(i)}\}$. Here, for any $p_k \in \mathcal{P}_1 \setminus \{p_{\pi(2)}, p_{\pi(3)} \ldots, p_{\pi(i)}\}$, we have

$$dist(c_{i+1}, p_k)^2 = \sum_{j \in [d]} (c_{i+1}(x_j) - p_k(x_j))^2$$
$$= \sum_{j \in \Pi} (c_{i+1}(x_j) - p_k(x_j))^2 + \sum_{j \in [d] \setminus \Pi} \left(\left(\frac{3}{2}\right)^{(i-1)} \left(\frac{1}{2} + \epsilon_d\right) - p_k(x_j)\right)^2.$$

Note that for $j \in \Pi$, both $c_{i+1}(x_j)$ and $p_k(x_j)$ are zero. Here, p_k has only one nonzero coordinate that is the kth coordinate with value 1 and $k \notin \Pi$. Therefore, we have

$$dist(c_{i+1}, p_k)^2 = 0 + \left(\left(\frac{3}{2}\right)^{(i-1)} \left(\frac{1}{2} + \epsilon_d\right) - 1\right)^2 + \sum_{j \in [d] \setminus \{\Pi \cup \{k\}\}} \left(\left(\frac{3}{2}\right)^{(i-1)} \left(\frac{1}{2} + \epsilon_d\right)\right)^2$$
$$= \left(\left(\frac{3}{2}\right)^{(i-1)} \left(\frac{1}{2} + \epsilon_d\right) - 1\right)^2 + (d - i) \left(\left(\frac{3}{2}\right)^{(i-1)} \left(\frac{1}{2} + \epsilon_d\right)\right)^2 \leq 1.$$

The last inequality follows by placing specific values of ϵ_d, i.e., 0.5 and 0.15 for $d = 2$ and 3, respectively. Hence invariant (II) is maintained.

As a result, any online algorithm needs $d+1$ hitting points $\{p_{\pi(2)}, p_{\pi(3)} \ldots p_{\pi(d+1)}\}$ and h_1; whereas offline optimum needs just one point $p_{\pi(d+1)}$. Thus, we have the following theorem:

Theorem 4. *The competitive ratio of every deterministic online algorithm for hitting unit balls in $\mathbb{R}^d$ using points in $\mathbb{Z}^d$ is at least $d + 1$, for $d < 4$.*

3.2 Upper Bound

Theorem 5. *For hitting unit balls using points in $\mathbb{Z}^d$, there exists a deterministic online algorithm whose competitive ratio is at most $O(1)$ and $O(d^4)$, for $d = 3$ and $d \geq 4$, respectively.*

Proof. **Algorithm:** On receiving a new input ball $\sigma \subset \mathbb{R}^d$ centered at $c \in \mathbb{R}^d$, if it has not been hit by the existing hitting set, then our online algorithm adds the nearest integer point from the center c as the hitting point. If the center is equidistant from several integer points, our algorithm arbitrarily chooses one of the nearest points as the hitting point.

Analysis: Let $\mathcal{A}$ and $\mathcal{O}$ be the hitting set returned by our online algorithm and the optimal hitting set, respectively. Let $p \in \mathcal{O}$ be any point in the optimal hitting set. Note that a unit ball $B_d(p, 1)$ centering at p contains all the centers of unit balls that can be hit by the point p. For simplicity, throughout the proof, let us assume that p is the origin. Let $\mathcal{A}_p$ be the set of hitting points placed by our online algorithm to pierce all balls having a center in $B_d(p, 1)$. It is easy to see that $\mathcal{A} = \cup_{p\in\mathcal{O}}\mathcal{A}_p$. Therefore, the competitive ratio of our algorithm is upper bounded by $\max_{p\in\mathcal{O}} |\mathcal{A}_p|$. For any point, $r \in B_d(p, 1)$, the maximum distance from r to its nearest integer point can be at most one (the maximum distance from the center c of the unit ball to any point $r \in B_d(c, 1)$ is at most one). Therefore, a ball $B_d(p, 2)$ centered at p having radius 2 contains $\mathcal{A}_p$. To complete the proof, we only need to calculate the cardinality of the set $\{z \in \mathbb{Z}^d : \sum_{i=1}^{d} |z|^2 \leq 4\}$. In other words, we need to count the number of $z = (z_1, z_2, \ldots, z_d) \in \mathbb{Z}^d$ that satisfies:

$$z_1^2 + z_2^2 + \ldots + z_d^2 \leq 4. \tag{4}$$

Note that to satisfy Equation (4), the coordinates of z cannot be other than $\{-2, -1, 0, 1, 2\}$

- When all the d coordinates are 0. There is only one possibility for this.
- When exactly one coordinate is nonzero. There will be $\binom{d}{1}$ many choices for the position of the nonzero coordinate. Now observe that for each nonzero coordinate, we have four choices $\{-2, -1, 1, 2\}$. So for this case, there will be a total of $4d$ integer points satisfying Equation (4).
- Note that any integer point having more than four nonzero coordinates will not satisfy Equation (4). Now consider exactly i nonzero coordinates, for $i = 2, 3, 4$. There will be $\binom{d}{i}$ many choices for the position of the nonzero coordinates. Now observe that if any of the nonzero coordinates is $\{-2, 2\}$, then the integer point will not satisfy Equation (4). Therefore, for each nonzero coordinate, we have just two choices $\{-1, 1\}$. Thus, there will be a total of $2^i\binom{d}{i}$ integer points satisfying Equation (4).

Now from above cases, there will be at most $1+4d+\sum_{i=2}^{4} 2^i\binom{d}{i} = O(d^4)$ integer points satisfying Equation (4). Hence $|\mathcal{A}| \leq O(d^4)|\mathcal{O}|$. Observe that for $d = 3$, the value of i is at most 3. As a result, for this special case, there will be at most 33 integer points satisfying Equation (4). Hence $|\mathcal{A}| \leq O(1)|\mathcal{O}|$, for $d = 3$. □

For unit disks in $\mathbb{R}^2$, we have the following result.

Theorem 6. *For hitting unit disks using points in $\mathbb{Z}^2$, there exists a deterministic online algorithm whose competitive ratio is at most 4.*

4 Conclusion

Due to the result of Evan and Smorodinsky[6], we know that no online algorithm can obtain a competitive ratio better than $\Omega(\log n)$ for hitting n intervals using points $\mathcal{P} = \{1, 2, \ldots, n\}$. Due to this pessimistic result, we restricted our attention to unit objects- balls and hypercubes in $\mathbb{R}^d$ using integer points in $\mathbb{Z}^d$. On one hand, we obtain almost tight bounds on the competitive ratio in the lower dimension. On the other hand, there is a significant gap between the lower and upper bound of the competitive ratio in higher-dimensional cases. We propose the following open problems.

1. Can the lower bound result of unit balls be extended to any $d \in \mathbb{N}$?
2. Is there a lower bound on the competitive ratio for hitting unit hypercubes that match with the upper bound of the problem? Is there any algorithm for hitting unit hypercubes with a competitive ratio of at most $O(d)$?

References

1. Noga Alon, Baruch Awerbuch, Yossi Azar, Niv Buchbinder, and Joseph (Seffi) Naor. The online set cover problem. *SIAM Journal on Computing*, 39(2):361–370, 2009. arXiv:https://doi.org/10.1137/060661946, doi:10.1137/060661946.
2. Allan Borodin and Ran El-Yaniv. *Online computation and competitive analysis.* Cambridge University Press, 1998.
3. Moses Charikar, Chandra Chekuri, Tomás Feder, and Rajeev Motwani. Incremental clustering and dynamic information retrieval. *SIAM J. Comput.*, 33(6):1417–1440, 2004. doi:10.1137/S0097539702418498.
4. Adrian Dumitrescu, Anirban Ghosh, and Csaba D. Tóth. Online unit covering in Euclidean space. *Theor. Comput. Sci.*, 809:218–230, 2020. doi:10.1016/j.tcs.2019.12.010.
5. Adrian Dumitrescu and Csaba D. Tóth. Online unit clustering and unit covering in higher dimensions. *Algorithmica*, 2022. doi:10.1007/s00453-021-00916-6.
6. Guy Even and Shakhar Smorodinsky. Hitting sets online and unique-max coloring. *Discret. Appl. Math.*, 178:71–82, 2014. doi:10.1016/j.dam.2014.06.019.
7. Uriel Feige. A threshold of ln n for approximating set cover. *J. ACM*, 45(4):634–652, 1998. doi:10.1145/285055.285059.
8. Robert J. Fowler, Michael S. Paterson, and Steven L. Tanimoto. Optimal packing and covering in the plane are np-complete. *Information Processing Letters*, 12(3):133–137, 1981. URL: https://www.sciencedirect.com/science/article/pii/0020019081901113, doi:https://doi.org/10.1016/0020-0190(81)90111-3.
9. Dorit S. Hochbaum and Wolfgang Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *J. ACM*, 32(1):130–136, 1985. doi:10.1145/2455.214106.
10. Richard M. Karp. *Reducibility among Combinatorial Problems*, pages 85–103. Springer US, Boston, MA, 1972. doi:10.1007/978-1-4684-2001-2_9.