Fair Covering of Points by Balls

Daniel Lokshtanov* Chinmay Sonar* Subhash Suri* Jie Xue*

Abstract

We consider the problem of covering a multi-colored set of points in $\mathbb{R}^d$ using (at most) k disjoint unit-radius balls chosen from a candidate set of unit-radius balls so that each color class is covered *fairly* in proportion to its size. Specifically, we investigate the complexity of covering the maximum number of points in this setting. We show that the problem is NP-hard even in one dimension when the number of colors is large. On the other hand, for a constant number of colors, we present a polynomial time exact algorithm in one dimension, and a PTAS in any fixed dimension $d \geqslant 2$.

1 Introduction

Given a set P of n points in $\mathbb{R}^d$ each of which is colored by one of t colors, the fair covering problem aims to cover the maximum number of points using k unit-radius balls such that the coverage for each color is in *proportion* to its size. More precisely, let $\mathcal{C}$ be a family of k unit radius balls, c_i be the number of the points of color i that are covered by $\mathcal{C}$, and n_i be the total number of points of color i, for $i \in \{1, \ldots, t\}$. Then we say that the covering $\mathcal{C}$ is *fair* if

$$\lfloor \rho_i \cdot c^* \rfloor \leqslant c_i \leqslant \lceil \rho_i \cdot c^* \rceil$$

for all $i \in \{1, \ldots, t\}$, where $c^* = \sum_{i=1}^{t} c_i$ and $\rho_i = n_i/n$ for $i \in \{1, \ldots, t\}$. Among all *fair* coverings, we want the one that maximizes the total coverage c^*. We note that an empty covering trivially satisfies the fairness condition but covers no points.

Achieving strict fair covering can be computationally hard, so we also define the notion of *approximately fair* covering. A covering $\mathcal{C}$ is called ε-fair for some $\varepsilon \in [0, 1]$, if

$$(1 - \varepsilon) \cdot \lfloor \rho_i \cdot c^* \rfloor \leqslant c_i \leqslant (1 + \varepsilon) \cdot \lceil \rho_i \cdot c^* \rceil$$

for all $i \in \{1, \ldots, t\}$. The goal of the approximately fair covering problem is then to find an ε-fair covering that maximizes the number of covered points.

The topic of algorithmic *fairness* has received significant attention recently [17, 25, 9, 15, 4, 10, 18, 7], especially with the increasing use of machine learning in policy and decision making. Our paper explores the computational implications of fairness as a constraint in geometric optimization by focusing on the specific problem of covering by unit balls, or equivalently, fixed-radius facility location. The different colors in our input represent different demographic groups and *proportionality* is one of the most basic forms of fairness, requiring that each group's share in the solution is proportional to its size. The proportional fairness can be easily extended to *weighted* sharing by assigning nonuniform weights to different points or color classes and measuring fairness on the overall covered weights. The fair covering problem can also be viewed as *fair clustering* under the k-center measure when each cluster is constrained to have unit radius.

Our Results

In this paper, we investigate the aforementioned (approximately) fair covering problem under the *discreteness* and *disjointness* constraints defined below. We require the balls used in a covering to be chosen from a given candidate set of unit-radius balls (discreteness) and to be pairwise disjoint (disjointness). Formally, the input of the problem consists of a set P of n t-colored points in $\mathbb{R}^d$, a candidate set $\mathcal{B}$ of m unit-radius balls in $\mathbb{R}^d$, and a number k that is the budget of balls to be used. Our goal is to find a (approximately) fair covering for P using at most k disjoint balls in $\mathcal{B}$ that covers the maximum number of points. Our main results are the following:

- We show that there exists an exact algorithm solving the fair covering problem in $\mathbb{R}^1$ in $O(mn^t)$ time. Alternatively, the problem can also be solved in $O(nm^k)$ time (Section 2.1).
- We show that the fair covering problem in $\mathbb{R}^1$ is NP-hard if the number of colors is part of the input. We also show that the problem is W[1]-hard parameterized by the number of covering balls k (Section 2.2).
- For a fixed $d \geqslant 2$ and a fixed number of colors, we present a PTAS for the approximately fair covering problem (Section 3).

*University of California, Santa Barbara, USA
Emails: {daniello, csonar, suri, jiexue}@cs.ucsb.edu

Related Work

The problem of covering points by balls or other geometric shapes has a long history in computational geometry, operations research, and theoretical computer science, due to its natural connections to clustering and facility location problems [3, 14, 20, 23, 24]. It is known that covering a set of two-dimensional points with a minimum number of unit disks is NP-hard, and so is the problem of maximizing the number of points covered by k unit disks [13, 19, 8, 11]. Recently, a number of researchers have considered clustering and covering problems with an additional constraint of fairness. In this setting, the input consists of points belonging to different colors (classes), and the goal is to find a solution where *each cluster* has approximately equal representation of all colors [21, 10, 6, 1, 22]. These formulations are different from our model because we allow individual clusters to be unbalanced as long as *in aggregate* each color receives its fair share. This non-local form of fair representation seems much harder than requiring each cluster to locally meet the balance condition. In another line of work, [5, 15, 2] consider a *colorful* variant of the k-center problem where the goal is to satisfy a *minimum* coverage for each color type. The colorful covering however does not achieve fairness because some color classes can have arbitrarily high representation in the output, as long as other colors meet the minimum threshold. In fact, enforcing the fairness by controlling *both* the lower and the upper bounds of representation seems to be a much harder problem, as suggested by some of our hardness results in one dimension.

2 Fair Covering in One Dimension

We begin by considering the problem in one dimension. Let $P = \{p_1, \dots, p_n\}$ be a set of n points on the real line each of which belongs to one of the t color classes, and let $\mathcal{B} = \{B_1, \dots, B_m\}$ be the candidate set of unit intervals on the line. (Technically speaking, a unit-radius ball in one dimension would be an interval of length 2, but a *unit-length* interval seems more natural, so that we shall use unit intervals in the following discussion. Note that the problem with intervals of length 2 is equivalent to the problem with unit intervals by simply scaling the points and the intervals.) Our goal is to cover the maximum number of points using at most k disjoint intervals in $\mathcal{B}$ under the fair covering constraint. We show that an optimal covering can be computed in polynomial time when the number t of colors is fixed, but the problem becomes intractable when t is part of the input.

2.1 A Dynamic Programming Algorithm

For simplicity, we describe our algorithm for $t = 2$ and use red/blue as the two colors for easier reference. The extension to an arbitrary number of colors is straightforward.

Given integers r and b, we define an (r, b)-*covering* to be a subset of $\mathcal{B}$ consisting of disjoint intervals that covers exactly r red and b blue points. An optimal (r, b)-covering is an (r, b)-covering that uses the minimum number of intervals. We solve the fair covering problem by computing an optimal (r, b)-covering for all $r, b \in \{1, \dots, n\}$. Without loss of generality, we assume that the unit intervals $B_1, \dots, B_m$ are sorted in the left-to-right order. Let $r(B_i)$ and $b(B_i)$ be the number of the red and blue points covered by B_i, respectively. For each $i \in \{1, \dots, m\}$, let $\pi_i < i$ be the largest integer such that $B_{\pi_i} \cap B_i = \emptyset$; we assume $\pi_1 = 0$. We make a left-to-right pass over the set of input points and the intervals on the real line, and compute π_i, $r(B_i)$, $b(B_i)$ for all $i \in \{1, \dots, m\}$.

Define $F[i, r, b]$ as the size of an optimal (r, b)-covering using only intervals in $\{B_1, \dots, B_i\}$. For the pairs (r, b) such that no (r, b)-covering exists, we set $F[i, r, b] = \infty$. It is easy to see that F satisfies the following recurrence.

Claim 1

$$F[i, r, b] \;=\; \min \left\{ \begin{array}{l} F[i-1, r, b] \\ 1 + F[\pi_i, r - r(B_i), b - b(B_i)] \end{array} \right\}$$

The above recurrence immediately allows us to compute the table F using dynamic programming, which is shown in Algorithm 1. The base case for the dynamic program is $F[i, 0, 0] = 0$ for all $i \in \{1, \dots, m\}$ and $F[0, r, b] = \infty$ for all $r, b \in \{1, \dots, n\}$.

Algorithm 1: Computing the F-table

```
Input: P, B
Compute π_i, r(B_i), b(B_i) for i ∈ {1, ..., m}
Initialize m × r × b sized table with value ∞
for i ∈ {0, ..., m}; r, b ∈ {0, ..., n} do
    F[i, r, b] ←
      min{F[i−1, r, b], 1 + F[π_i, r − r(B_i), b − b(B_i)]}
end
return F
```

Lemma 2 *Algorithm 1 can be implemented in worst-case time* $O((n + m)\log(n + m) + mn^2)$.

Proof. Sorting P and $\mathcal{B}$ takes $O((n + m)\log(n + m))$ time. Computing $\pi_i, r(B_i), b(B_i)$ for all $i \in \{1, \dots, m\}$ takes additional linear time. After that the F-table can be computed in $O(mn^2)$ time. ☐

Once the F-table is computed, we can solve the fair covering problem by checking all entries in the table for which the (r, b)-covering is fair and has $F[m, r, b] \leqslant k$. Among all such valid pairs, we return the pair (r^*, b^*) with the maximum $r^* + b^*$. Clearly, $c^* = r^* + b^*$ is the optimum of the problem instance. We therefore have the following result.

Theorem 3 *The fair covering problem in $\mathbb{R}^1$ with $t = 2$ colors can be solved in $O((n+m)\log(n+m)+mn^2)$ time.*

The dynamic program easily extends to the case of $t > 2$ colors, by using a $(t + 1)$-dimensional DP table.

Theorem 4 *The fair covering problem in $\mathbb{R}^1$ can be solved in $O((n + m)\log(n + m) + mn^t)$ time.*

Remarks. Recall that the fair covering problem we investigate is defined with the discreteness and disjointness constraints. In fact, the problem without each of these two constraints can also be solved using similar dynamic programming approaches. We omit the details here because our main focus is the problem with the discreteness and disjointness constraints.

2.2 NP and W[1]-Hardness of the Fair Covering

In this section, we show that the one-dimensional fair covering problem is NP-hard if the number of colors t is large. We also show that the problem is W[1]-hard parameterized by the number of intervals k.

Theorem 5 *The one-dimensional fair covering problem with $\Omega(n)$ colors is NP-hard.*

Proof. We reduce the well-known Exact Cover problem [16] to our problem. Given a ground set $\mathcal{U}$, a family $\mathcal{F}$ of subsets of $\mathcal{U}$, and an integer ℓ, the Exact Cover problem is to decide if there exists a $\mathcal{S} \subseteq \mathcal{F}$ of size ℓ that contains each element of $\mathcal{U}$ exactly once. The construction is described below.

Construction. Given an instance of Exact Cover with $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$, $\mathcal{F} = \{S_1, S_2, \ldots, S_m\}$, and an integer ℓ, we construct a set of points P, and a set of centers $\mathcal{M}$ as follows. The i^{th} element of $\mathcal{U}$ is associated with color i; thus, there are n color classes. We also introduce an additional color 0, which we call *special*. The set of points is organized in the following three groups.

1. *Basic Points:* For each set $S_i \in \mathcal{F}$, we introduce $|S_i|$ points, placed arbitrarily within the interval $[3i, 3i + 1)$. Each point has the color of its element.

Figure 1: Constructed fair covering instance for an Exact Cover instance $\mathcal{U} = \{1, 2, 3\}$, $\mathcal{F} = \{(1, 3), (2), (1, 2)\}$, $\ell = 2$. We introduce red (1), green (2), and blue (3) colors corresponding to the elements in the universe, and we also introduce cyan as the special color. First five points are introduced in the basic points group. Since $f^* = 2$ (where f^* is a maximum number of sets to which an element of $\mathcal{U}$ belongs to), next, we introduce one blue point so that each color except for cyan has exactly two points. At last, we introduce 4 cyan points as enforcers (since $f^* = \ell = 2$).

   The intervals corresponding to S_i and S_j, $i \neq j$, are distance 2 apart, which ensures that any unit interval of $\mathcal{B}$ can cover points of at most one such group.

2. *Balancers:* We add extra points for each color i to ensure that all colors $i = 1, 2, \ldots, n$ end up with the same number of points. Specifically, let f^* be the maximum number of sets to which an element belongs, and let f_i be the number of sets containing the element u_i. We introduce $f^* - f_i$ points of color i in the interval $[3(m + i), 3(m + i) + 1)$.

3. *Enforcers:* Finally, we introduce ℓf^* points of color 0 (special color), at locations $3(m + n + 1), 3(m + n + 2), \ldots, 3(m + n + \ell f^*)$. These are needed in our construction to enforce the fair covering condition. Refer figure 1.

Finally, the set of centers $\mathcal{M}$ is defined as follows.

- For each $S_i \in \mathcal{F}$, we add a center at $3i + 1/2$, which allows all points of that group to be covered by one unit interval.
- Each enforcer point is also a center. We do not need centers for the balancers—their role is primarily to make all color classes have equal size.

Finally, we fix the number of covering intervals to be $k = 2\ell$.

We now argue that the Exact Covering instance is a yes instance if and only if our fair covering instance admits a k-covering with at least $n + \ell$ points.

For the forward direction of the proof, suppose $\mathcal{S} \subseteq \mathcal{F}$ is an exact cover of size ℓ, and $\mathcal{T} = \{i \mid S_i \in \mathcal{S}\}$ be the set of indices. Then we build a covering $\mathcal{C}$ as follows. We place first ℓ intervals centered at $3i + 1/2$ for $i \in \mathcal{T}$, and the remaining ℓ intervals are placed at $3(m + n + j)$ for $j = 1, 2, \ldots, \ell$ covering one special colored point each. Since $\mathcal{S}$ is an exact cover, $\mathcal{C}$ contains exactly $n + \ell$ points. The covering is also fair, since all the colors $i = 1, 2, \dots, n$ have the same number of points f^*, and the special color 0 has ℓf^* points. In the covering, each of the color classes $i = 1, 2, \dots, n$ has one covered point and the special color has ℓ points.

For the reverse direction, let $\mathcal{C}$ be the fair covering with at least $n + \ell$ points. We observe that a fair covering necessarily contains the same number of points, say p, for each color $i = 1, 2, \dots, n$, and contains exactly ℓp points of the special color. For $p = 2$, to cover 2ℓ special colored points only, we need all 2ℓ intervals. Hence, for any *fair* covering, we get $p < 2$. This implies that for the covering $\mathcal{C}$, $p = 1$ to meet the overall covering requirement. Since, we need ℓ intervals to cover ℓ special colored points, it is easy to see that the remaining ℓ intervals cover exactly one point of every other color. Hence, the intervals covered corresponds to an Exact Cover. □

In the reduced instance above, the number of intervals is dependent only upon the size of the Exact Cover (ℓ). The Exact Cover problem is known to be W[1]-hard parameterized by ℓ [12]. Hence, the analogous results for the fair covering problem is summarized as follows:

Theorem 6 *The fair covering problem is* W[1]*-hard parameterized by the number of covering balls (k).*

In dimensions $d \geqslant 2$, the maximum coverage problem is NP-hard [13], and W[1]-hard [19], *even without the fairness constraint.*

3 A PTAS for Fair Covering in d Dimensions

In this section, we describe a PTAS for the approximately fair covering problem in any fixed dimension d. Specifically, given an approximate factor $\varepsilon \in [0, 1]$, we want to compute an ε-fair covering of P (using at most k disjoint balls in $\mathcal{B}$) such that the number of the points covered is at least $(1-\varepsilon)\cdot\mathsf{opt}$, where opt is the size of an optimal fair covering of P. In other words, the approximation is bi-criteria: one criterion is on the fairness of the covering while the other one is on the quality of the solution (i.e., the number of the points covered). For the simplicity of exposition, we describe the algorithm in two dimensions ($d = 2$) and for two colors ($t = 2$). The extension to higher dimensions and the general case of $t > 2$ colors is straightforward.

3.1 Shifted Partitions & Approximate Covering

When solving the fair covering problem in $\mathbb{R}^1$, we were able to compute an optimal (r, b)-covering for any (r, b) pair. This seems quite difficult in higher dimensions, and so we resort to solving an approximate version of this problem as follows. We want to compute a table $\varGamma[1 \dots n, 1 \dots n]$ of integers such that for each pair (r, b), we have the following:

1. $\varGamma[r, b]$ is at least the size of an optimal (r, b)-covering, and
2. there exists $r^* \in [(1-\varepsilon)r, r]$ and $b^* \in [(1-\varepsilon)b, b]$ such that $\varGamma[r^*, b^*]$ is at most the size of an optimal (r, b)-covering.

For convenience, we call such a table $\varGamma$ an *ε-approximate covering table* (ε-ACT) for the instance $(P, \mathcal{B})$. Note that to solve the approximately fair covering problem, it suffices to compute an ε-ACT.

Lemma 7 *Given an ε-ACT $\varGamma$ for $(P, \mathcal{B})$, one can solve the approximately fair covering problem in polynomial time.*

Proof. Suppose an optimal fair covering covers r_0 red points and b_0 blue points. We call a pair (r, b) with $r, b \in \{1, \dots, n\}$ *feasible* if **(1)** an (r, b)-covering is fair and **(2)** there exists $r^* \in [(1-\varepsilon)r, r]$ and $b^* \in [(1-\varepsilon)b, b]$ such that $\varGamma[r^*, b^*] \leqslant k$. We compute all feasible pairs, which can clearly be done in polynomial time given $\varGamma$, and find the feasible pair (r, b) that maximizes $r + b$. By definition, we can find $r^* \in [(1-\varepsilon)r, r]$ and $b^* \in [(1-\varepsilon)b, b]$ such that $\varGamma[r^*, b^*] \leqslant k$. Note that an (r^*, b^*)-covering is ε-fair. Furthermore, $r + b \geqslant \mathsf{opt}$ since (r_0, b_0) is feasible, hence $r^* + b^* \geqslant (1-\varepsilon)\cdot\mathsf{opt}$. Because $\varGamma$ is an ε-ACT, there exists an (r^*, b^*)-covering using at most k (disjoint) disks in $\mathcal{B}$. Therefore, $r^* + b^*$ is a $(1-\varepsilon)$-approximate solution for the approximately fair covering problem. □

In order to compute an ε-ACT $\varGamma$, we use the shifting technique [14]. Let $h = h(\varepsilon)$ be an integer parameter to be determined later. For an integer $i \in \mathbb{Z}$, let $\square_{i,j}$ denote the $h \times h$ square $[i, i+h] \times [j, j+h]$; we say $\square_{i,j}$ is *nonempty* if it contains at least one point in P. We first compute the index set $I = \{(i, j) : \square_{i,j} \text{ is nonempty}\}$. This can be easily done in time polynomial in n and h, by computing for each $p \in P$, the $O(h^2)$ squares $\square_{i,j}$ that contains p. For each $(i, j) \in I$, define $P_{i,j} = P \cap \square_{i,j}$ and $\mathcal{B}_{i,j} = \{B \in \mathcal{B} : B \subseteq \square_{i,j}\}$. In the next step, we compute a 0-ACT $\varGamma_{i,j}$ for each $(P_{i,j}, \mathcal{B}_{i,j})$ with $(i, j) \in I$. We will show later in Section 3.2 how to compute $\varGamma_{i,j}$ in $(n_{i,j} + m_{i,j})^{O(h^2)}$ time, where $n_{i,j} = |P_{i,j}|$ and $m_{i,j} = |\mathcal{B}_{i,j}|$. At this point, let us assume we have the 0-ACTs $\varGamma_{i,j}$ and finish the description of our PTAS. We have the following key observation.

Lemma 8 *Let $\{P_1, \dots, P_s\}$ be a partition of P and $\mathcal{B}_1, \dots, \mathcal{B}_s \subseteq \mathcal{B}$ be disjoint subsets such that the disks* *in $\mathcal{B}_i$ do not cover any points in $P \backslash P_i$. Given 0-ACTs for $(P_1, \mathcal{B}_1), \dots, (P_s, \mathcal{B}_s)$, we can compute a 0-ACT for $(P, \bigcup_{i=1}^{s} \mathcal{B}_i)$ in polynomial time.*

Proof. Computing a 0-ACT for $(P, \bigcup_{i=1}^{s} \mathcal{B}_i)$ is equivalent to computing for all pairs (r, b) the size of the smallest (r, b)-covering of $(P, \bigcup_{i=1}^{s} \mathcal{B}_i)$. Since the disks in $\mathcal{B}_i$ can only cover the points in P_i, the entire problem instance can be divided into independent sub-problems $(P_1, \mathcal{B}_1), \dots, (P_s, \mathcal{B}_s)$. This allows us to solve the problem in polynomial time using dynamic programming; see Algorithm 2. $\square$

Algorithm 2: Computing the 0-ACT

Input: $\Gamma_1, \dots, \Gamma_s$, where Γ_i is a 0-ACT for $(P_i, \mathcal{B}_i)$

Initialize a $s \times n \times n$ table F with value ∞

for $t \in \{1, \dots, s\}$; $r, b \in \{1, \dots, n\}$ **do**

  $F[t, r, b] \leftarrow \min_{0 \leqslant r' \leqslant r, \ 0 \leqslant b' \leqslant b} \{\Gamma_t[r', b'] + F[t-1, r-r', b-b']\}$

end

$\Gamma^*[r, b] = F[s, r, b]$ for all $r, b \in \{1, \dots, n\}$.

return Γ^*

For $x, y \in \{0, \dots, h-1\}$, let $L_{x,y}$ be the set of all integer pairs (i, j) such that $i \bmod h = x$ and $j \bmod h = y$ (See Fig. 2a). We write $I_{x,y} = I \cap L_{x,y}$.

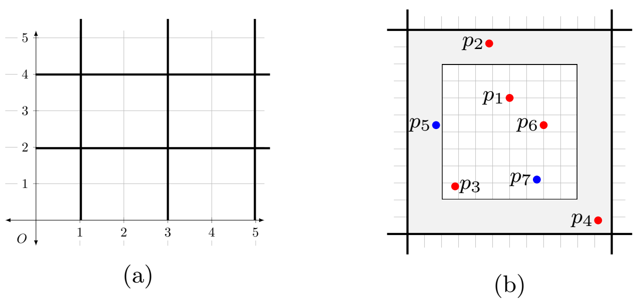

Figure 2: (a) The squares $\square_{i,j}$ for $(i, j) \in L_{1,0}$, with $h = 2$. (b) An illustration of the boundary points. The outer square is $\square_{i,j}$ and the inner square is $[i+2, i+h-2] \times [j+2, j+h-2]$, with $h = 12$. The points in the gray region (i.e., p_2, p_4, p_5) are the boundary points in $\square_{i,j}$.

Lemma 9 *For all $x, y \in \{0, \dots, h-1\}$, the squares $\square_{i,j}$ for $(i, j) \in I_{x,y}$ are interior-disjoint and cover all points in P.*

Proof. Note that the squares $\square_{i,j}$ for $(i, j) \in L_{x,y}$ are interior-disjoint and cover the entire plane $\mathbb{R}^2$ (see Figure 2a for an example). It directly follows that the squares $\square_{i,j}$ for $(i, j) \in I_{x,y}$ are interior-disjoint. Consider a point $p \in P$ and let $(i, j) \in L_{x,y}$ such that $p \in \square_{i,j}$. Clearly, $(i, j) \in I$ as $\square_{i,j}$ is nonempty and hence $(i, j) \in I_{x,y}$. Therefore, all points in P are covered by the squares $\square_{i,j}$ for $(i, j) \in I_{x,y}$. $\square$

Fix $x, y \in \{0, \dots, h-1\}$. We know by Lemma 9 that $\{P_{i,j} : (i, j) \in I_{x,y}\}$ is a partition of P and the collections $\mathcal{B}_{i,j}$ for $(i, j) \in I_{x,y}$ are disjoint. Furthermore, the disks in $\mathcal{B}_{i,j}$ do not cover any point in $P \backslash P_{i,j}$. Therefore, we can apply Lemma 8 to compute a 0-ACT $\Gamma^{(x,y)}$ for $(P, \bigcup_{(i,j) \in I_{x,y}} \mathcal{B}_{i,j})$ in polynomial time. We do this for all $x, y \in \{0, \dots, h-1\}$. Finally, we construct the table Γ by setting $\Gamma[r, b] = \min_{x,y \in \{0,\dots,h-1\}} \Gamma^{(x,y)}[r, b]$. We shall show that Γ is a $\frac{12h-12}{h^2}$-ACT for $(P, \mathcal{B})$. To this end, we introduce some notions. For a point $p \in P$ and a square $\square_{i,j}$, we say p is a *boundary point* in $\square_{i,j}$ if $p \in \square_{i,j}$ and $p \notin [i+2, i+h-2] \times [j+2, j+h-2]$ (See Figure 2b). Now consider some $x, y \in \{0, \dots, h-1\}$. We say $p \in P$ *conflicts* with the pair (x, y) if p is a boundary point in $\square_{i,j}$ where $(i, j) \in I_{x,y}$ is the (unique) pair such that $p \in \square_{i,j}$. One can easily see that each point $p \in P$ conflicts with exactly $h^2 - (h-2)^2$ pairs (x, y).

Lemma 10 *For any $P' \subseteq P$, there exists some $x, y \in \{0, \dots, h-1\}$ such that the number of red (resp., blue) points in P' conflicting with (x, y) is at most $\frac{12h-12}{h^2} \cdot n'_{red}$ (resp., $\frac{12h-12}{h^2} \cdot n'_{blue}$), where n'_{red} (resp., n'_{blue}) is the total number of red (blue) points in P'.*

Proof. Define $\delta_{x,y}^{\text{red}}$ (resp., $\delta_{x,y}^{\text{blue}}$) as the number of the red (resp., blue) points in P' that conflict with (x, y). Because any point $p \in P$ conflicts with exactly $h^2 - (h-2)^2$ pairs (x, y), we have

$$\sum_{x=0}^{h-1} \sum_{y=0}^{h-1} \delta_{x,y}^{\text{red}} = n'_{\text{red}}(h^2 - (h-2)^2) = n'_{\text{red}}(4h-4).$$

Therefore, the number of the pairs (x, y) such that $\delta_{x,y}^{\text{red}} \geqslant 3n'_{\text{red}}(4h-4)/h^2$ is at most $h^2/3$. Equivalently, the number of the pairs (x, y) such that $\delta_{x,y}^{\text{red}} < 3n'_{\text{red}}(4h-4)/h^2$ is at least $2h^2/3$. For the same reason, the number of the pairs (x, y) such that $\delta_{x,y}^{\text{blue}} < 3n'_{\text{blue}}(4h-4)/h^2$ is at least $2h^2/3$. Since $2h^2/3 + 2h^2/3 > h^2$, there exists at least one pair (x, y) that simultaneously satisfies $\delta_{x,y}^{\text{red}} < 3n'_{\text{red}}(4h-4)/h^2$ and $\delta_{x,y}^{\text{blue}} < 3n'_{\text{blue}}(4h-4)/h^2$. This completes the proof of the lemma. $\square$

Now we are ready to prove that Γ is a $\frac{12h-12}{h^2}$-ACT.

Lemma 11 *Γ is a $\frac{12h-12}{h^2}$-ACT for $(P, \mathcal{B})$.*

Proof. Set $\eta = \frac{12h-12}{h^2}$. By the definition of a η-ACT, we have to verify that **(1)** $\Gamma[r,b]$ is at least the size of a smallest (r,b)-covering of $(P,\mathcal{B})$ and **(2)** there exist $r^* \in [(1-\eta)r, r]$ and $b^* \in [(1-\eta)b, b]$ such that $\Gamma[r^*, b^*]$ is at most the size of a smallest (r,b)-covering of $(P,\mathcal{B})$. Condition **(1)** is clearly true. Indeed, for all $x, y \in \{0,\dots,h-1\}$, $\Gamma^{(x,y)}[r,b]$ is the size of the smallest (r,b)-covering of $(P, \bigcup_{(i,j)\in I_{x,y}} \mathcal{B}_{i,j})$ and hence is at least the size of a smallest (r,b)-covering of $(P,\mathcal{B})$. Next, we verify condition **(2)**. Let $\mathcal{B}' \subseteq \mathcal{B}$ be a smallest (r,b)-covering of $(P,\mathcal{B})$ and $P' \subseteq P$ be the points covered by the disks in $\mathcal{B}'$ (hence P' consists of r red points and b blue points). By Lemma 10, there exist $x, y \in \{0,\dots,h-1\}$ such that the number of red (resp., blue) points in P' conflicting with (x,y) is at most ηr (resp., ηb). Let $\mathcal{B}'' = \mathcal{B}' \cap (\bigcup_{(i,j)\in I_{x,y}} \mathcal{B}_{i,j})$ and $P'' \subseteq P'$ be the points covered by the disks in $\mathcal{B}''$. Suppose P'' consists of r^* red points and b^* blue points. Note that any disk in $\mathcal{B}' \backslash \mathcal{B}''$ can only cover the points in P that conflict with (x,y). Therefore, any point in P' that does not conflict with (x,y) must be contained in P'', which implies that $r^* \in [(1-\eta)r, r]$ and $b^* \in [(1-\eta)b, b]$. Since $\Gamma^{(x,y)}$ is a 0-ACT for $(P, \bigcup_{(i,j)\in I_{x,y}} \mathcal{B}_{i,j})$, we have $\Gamma^{(x,y)}[r^*, b^*] \leqslant |\mathcal{B}''| \leqslant |\mathcal{B}'|$. It follows that condition **(2)** is also true. $\square$

We set h to be the smallest integer such that $\frac{12h-12}{h^2} \leqslant \varepsilon$; clearly, $h = O(1/\varepsilon)$. Then by the above lemma, Γ is an ε-ACT for $(P,\mathcal{B})$. In this way, we obtain a PTAS for the fair covering problem in $\mathbb{R}^2$.

Theorem 12 *There exists a $(1-\varepsilon)$-approximation algorithm for the fair covering problem in $\mathbb{R}^2$ which runs in $n^{O(1)} m^{O(1/\varepsilon^2)}$ time.*

Proof. In our algorithm, the most time-consuming work is the computation of each $\Gamma_{i,j}$ for $(i,j) \in I$, which takes $n_{i,j}^{O(1)} m_{i,j}^{O(h^2)}$ time as claimed before. All the other work can be done in time polynomial in h, n, m. Since $I = O(h^2 n)$, the overall time complexity of our algorithm is $(n+m)^{O(h^2)}$, i.e., $n^{O(1)} m^{O(1/\varepsilon^2)}$. $\square$

The algorithm can be straightforwardly generalized to higher dimensions and the case $t > 2$, resulting in the following theorem.

Theorem 13 *There exists a $(1-\varepsilon)$-approximation algorithm for the t-color fair covering problem in $\mathbb{R}^d$ which runs in $n^{O(t)} m^{O(1/\varepsilon^d)}$ time.*

3.2 Computing the 0-ACTs $\Gamma_{i,j}$

We now discuss the only missing piece in our algorithm above: the computation of the tables $\Gamma_{i,j}$. Recall that $\Gamma_{i,j}$ is a 0-ACT for $(P_{i,j}, \mathcal{B}_{i,j})$. We show that each $\Gamma_{i,j}$ can be computed in $n_{i,j}^{O(1)} m_{i,j}^{O(h^2)}$ time where $n_{i,j} = |P_{i,j}|$ and $m_{i,j} = |\mathcal{B}_{i,j}|$. The key observation is the following.

Lemma 14 *For $r, b \in \{1,\dots,n_{i,j}\}$, an (r,b)-covering of $(P_{i,j}, \mathcal{B}_{i,j})$ is of size at most $\lfloor h^2/\pi \rfloor$.*

Proof. Recall that an (r,b)-covering of $(P_{i,j}, \mathcal{B}_{i,j})$ consists of *disjoint* disks in $\mathcal{B}_{i,j}$. All disks in $\mathcal{B}_{i,j}$ are contained in the $h \times h$ square $\square_{i,j}$. The area of $\square_{i,j}$ is h^2 and the area of a unit-disk is π. Therefore, any subset of disjoint disks in $\square_{i,j}$ is of size at most $\lfloor h^2/\pi \rfloor$. $\square$

With the above observation, we can compute $\Gamma_{i,j}$ as follows. We enumerate all subsets of $\mathcal{B}_{i,j}$ of size at most $\lfloor h^2/\pi \rfloor$, and keep the ones that consist of disjoint disks. In this way, we obtain all (r,b)-coverings of $(P_{i,j}, \mathcal{B}_{i,j})$ for all $r, b \in \{1,\dots,n_{i,j}\}$. By checking these coverings one by one, we can find the smallest (r,b)-covering for all $r, b \in \{1,\dots,n_{i,j}\}$, and hence compute $\Gamma_{i,j}$. The total time cost is $n_{i,j}^{O(1)} m_{i,j}^{O(h^2)}$.

4 Conclusion

In this paper, we introduced a new fair-covering problem, which is motivated by fair representation of multiple demographics in a geometric facility location setting. We proved that the problem is NP-hard even in one dimension when the number of color groups is large. When the number of colors is fixed, we presented a polynomial time exact algorithm in one dimension, and a PTAS in any fixed dimension. Many open problems remain, including whether one can achieve a constant factor approximation significantly faster than our PTAS, and whether the PTAS can be achieved for covering by non-disjoint balls.

References

[1] S. Ahmadian, A. Epasto, R. Kumar, and M. Mahdian. Clustering without over-representation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 267–275, 2019.

[2] G. Anegg, H. Angelidakis, A. Kurpisz, and R. Zenklusen. A Technique for Obtaining True Approximations for k-Center with Covering Constraints. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 52–65. Springer, 2020.

[3] E. M. Arkin and R. Hassin. Approximation algorithms for the geometric covering salesman problem. *Discrete Applied Mathematics*, 55(3):197–218, 1994.

[4] A. Backurs, P. Indyk, K. Onak, B. Schieber, A. Vakilian, and T. Wagner. Scalable fair clustering. *arXiv preprint arXiv:1902.03519*, 2019.

[5] S. Bandyapadhyay, T. Inamdar, S. Pai, and K. Varadarajan. A Constant Approximation for Colorful k-Center. In *27th Annual European Symposium on Algorithms (ESA 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[6] S. Bera, D. Chakrabarty, N. Flores, and M. Negahbani. Fair algorithms for clustering. In *Advances in Neural Information Processing Systems*, pages 4955–4966, 2019.

[7] L. E. Celis, V. Keswani, D. Straszak, A. Deshpande, T. Kathuria, and N. K. Vishnoi. Fair and diverse DPP-based data summarization. *arXiv preprint arXiv:1802.04023*, 2018.

[8] T. M. Chan and E. Grant. Exact algorithms and APX-hardness results for geometric packing and covering problems. *Computational Geometry*, 47(2):112–124, 2014.

[9] X. Chen, B. Fain, L. Lyu, and K. Munagala. Proportionally Fair Clustering. In *International Conference on Machine Learning*, pages 1032–1041, 2019.

[10] F. Chierichetti, R. Kumar, S. Lattanzi, and S. Vassilvitskii. Fair clustering through fairlets. In *Advances in Neural Information Processing Systems*, pages 5029–5037, 2017.

[11] K. L. Clarkson and K. Varadarajan. Improved approximation algorithms for geometric set cover. *Discrete & Computational Geometry*, 37(1):43–58, 2007.

[12] R. G. Downey and M. R. Fellows. Fixed-parameter tractability and completeness ii: On completeness for W[1]. *Theoretical Computer Science*, 141(1-2):109–131, 1995.

[13] R. J. Fowler, M. S. Paterson, and S. L. Tanimoto. Optimal packing and covering in the plane are NP-complete. *Information processing letters*, pages 133–137, 1981.

[14] D. S. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *Journal of the ACM (JACM)*, pages 130–136, 1985.

[15] X. Jia, K. Sheth, and O. Svensson. Fair Colorful k-Center Clustering. In *Integer Programming and Combinatorial Optimization*, pages 209–222. Springer, 2020.

[16] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[17] J. Kleinberg, H. Lakkaraju, J. Leskovec, J. Ludwig, and S. Mullainathan. Human decisions and machine predictions. *The quarterly journal of economics*, 133(1):237–293, 2018.

[18] M. Kleindessner, P. Awasthi, and J. Morgenstern. Fair k-Center Clustering for Data Summarization. In *International Conference on Machine Learning*, pages 3448–3457, 2019.

[19] D. Marx. Efficient approximation schemes for geometric problems? In *European Symposium on Algorithms*, pages 448–459, 2005.

[20] N. Megiddo and K. J. Supowit. On the complexity of some common geometric location problems. *SIAM journal on computing*, 13(1):182–196, 1984.

[21] C. Rösner and M. Schmidt. Privacy Preserving Clustering with Constraints. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[22] M. Schmidt, C. Schwiegelshohn, and C. Sohler. Fair coresets and streaming algorithms for fair k-means. In *International Workshop on Approximation and Online Algorithms*, pages 232–251. Springer, 2019.

[23] C. Toregas, R. Swain, C. ReVelle, and L. Bergman. The location of emergency service facilities. *Operations research*, 19(6):1363–1373, 1971.

[24] D. Yang, S. Misra, X. Fang, G. Xue, and J. Zhang. Two-tiered constrained relay node placement in wireless sensor networks: Computational complexity and efficient approximations. *IEEE Transactions on Mobile Computing*, 11(8):1399–1411, 2011.

[25] M. B. Zafar, I. Valera, M. Rodriguez, K. Gummadi, and A. Weller. From parity to preference-based notions of fairness in classification. In *Advances in Neural Information Processing Systems*, pages 229–239, 2017.