Geometric Cover with Outliers Removal

Zhengyang Guo
School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore

Yi Li
School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore

Abstract

We study the problem of partial geometric cover, which asks to find the minimum number of geometric objects (unit squares and unit disks in this work) that cover at least $(n-t)$ of n given planar points, where $0 \le t \le n/2$. When $t = 0$, the problem is the classical geometric cover problem, for which many existing works adopt a general framework called the shifting strategy. The shifting strategy is a divide and conquer paradigm which partitions the plane into equal-width strips, applies a local algorithm on each strip and then merges the local solutions with only a small loss on the overall approximation ratio. A challenge to extend the shifting strategy to the case of outliers is to determine the number of outliers in each strip. We develop a shifting strategy incorporating the outlier distribution, which runs in $O(tn \log n)$ time. We also develop local algorithms on strips for the outliers case, improving the running time over previous algorithms, and consequently obtain approximation algorithms to the partial geometric cover.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Geometric Cover, Unit Square Cover, Unit Disk Cover, Shifting Strategy, Outliers Detection, Computational Geometry

Digital Object Identifier 10.4230/LIPIcs.STACS.2021.39

1 Introduction

The geometric cover with outlier is a generalization of the classic geometric cover problems to the case where there are outliers. As the real-world data usually contain outliers, which can dramatically affect the output of a general geometric cover algorithm, we hope to exclude the outliers along with finding an optimal covering. Given n points in the plane and an integer $0 \le t \le n/2$, the geometric cover with outliers asks to find the minimum number of geometric objects of a given type that cover at least $(n-t)$ points. The uncovered points are referred to as outliers. The geometric objects we consider in this work are the most two common ones, namely, unit squares (side length 1) and unit disks (radius 1). Correspondingly, we call the two problems partial square cover and partial disk cover, denoted by PSC and PDC, respectively. The problems are formally defined as follows.

▶ **Problem 1** (Partial Geometric Covers). *Given a planar point set X of size n and an integer $0 \le t \le n/2$, we define three problems as follows.*

- *Partial Unit Square Cover (PSC) find a minimum number of unit squares that cover at least $(n-t)$ points of X.*
- *Partial Unit Disk Cover (PDC): find a minimum number of unit disks that cover at least $(n-t)$ points of X.*
- *Restricted Partial Unit Disk Cover (RPDC): find a minimum subset of X such that the unit disks centered in the subset cover at least $(n-t)$ points of X.*

The special case of PSC and PDC when $t = 0$ are the classical square cover and disk cover problems, which were motivated by the applications in image processing [25] and wireless networking [23]. Both of them are known to be strongly NP-hard and hence no FPTAS

38th International Symposium on Theoretical Aspects of Computer Science (STACS 2021).
Editors: Markus Bläser and Benjamin Monmege; Article No. 39; pp. 39:1–39:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

exists [18]. Even when the square or disk positions are restricted to the given point set, the problem (which becomes RPDC in the disk case) remains NP-hard [7]. Therefore the research has been considering developing polynomial-time algorithms with a small approximation ratio and time complexity.

For PSC, Gonzalez gives a 2-approximation with time complexity $O(n \log n)$ and a $(1+\epsilon)$-approximation algorithm with time complexity $O(\epsilon^{-1} n^{1/\epsilon})$ in [14]. The techniques can be adapted to PDC, producing a $(2+\epsilon)$-approximation solution in $O(\epsilon^{-2} n^7)$ time [14, 3] and also a $(1+\epsilon)$-approximation in $O(\epsilon^{-2} n^{6\lceil\sqrt{2}/\epsilon\rceil-1})$ time [20]. They suffice for a PTAS but for a constant-factor approximation it could incur a high runtime. Furthermore, an $O(n \log n)$-time 4-approximation algorithm was given in [3] and there also exists a 2.8334-approximation with runtime $O(n(\log n \log\log n)^2)$ in [10].

These algorithms are all for the case of no outliers. To the best of out knowledge, we only found three papers studying the outlier case of the covering problems as listed in Problem 1. In [11], Gandhi et al. gave a $(1+\epsilon)^2$-approximation algorithm for PDC which runs in runtime $O(\epsilon^{-1} n^{2\lceil\sqrt{2}/\epsilon\rceil^2+1})$. Later in [12], Ghasemalizadeh and Razzazi gave a $(1+\epsilon)$-approximation to PSC with outliers in runtime $O(\epsilon^{-1} n^{4/\epsilon+2})$ and a $(1+\epsilon)$-approximation to PDC with outliers in time[1] $O(\epsilon^{-1} n^{6\lceil\sqrt{2}/\epsilon\rceil+2})$. Note that their runtimes do not depend on t, the number of outliers, as their algorithms actually compute the solutions for all $t = 0, \ldots, n/2$. Additionally for PDC, Inamdar studies the local search methods in [19] and gives for $0 < \epsilon \leq 1/2$ an $(1+4\epsilon)$-approximation algorithm that runs in time at least $n^{O(1/\epsilon^2)}$ time. The best existing results for both the outlier and the non-outlier cases are summarized in Table 1, where $\ell = 1/\epsilon$ is the approximation parameter.

Although there are only limited works studying the partial geometric over, detecting outliers along with shape fitting tasks is of special significance in computational geometry and has thus become an enduringly popular research topic. Examples include k-means/medians/centers clustering with outliers [15], subsets of size $(n-t)$ with the minimum diameter [6], rectangles of the minimum area that covers at least $(n-t)$ points [24], convex hulls with outliers [2] and projective clustering with outliers [21].

A particularly important variant of the unit disk cover problem is when the given points lie within a vertical strip and we define the strip variants of Problem 1 as follows.

▶ **Problem 2** (Within-Strip Partial Geometric Covers). *Given a planar point set X contained in a vertical strip of width W and an integer $0 \leq t \leq n/2$ where $n = |X|$. We define three problems as follows.*

- *Within-strip Partial Unit Square Cover (StripPSC) find a minimum number of unit squares that cover at least $(n-t)$ points of X.*
- *Within-strip Partial Unit Disk Cover (StripPDC): find a minimum number of unit disks that cover at least $(n-t)$ points of X.*
- *Within-strip Restricted Partial Unit Disk Cover (StripRPDC): find a minimum subset of X such that the unit disks centered in the subset cover at least $(n-t)$ points of X.*

[1] We note an omission in the time complexity for PDC claimed in [12] and corrected it in our claim. In the last paragraph in Section 3 of [12, p553–554], which discusses the adaptation of the strip partial covering algorithm [12, p551] to disks. It divides the plane into strips of width 1 and group ℓ consecutive strips. However, when $\ell = 1$, the claim that "there can be no disk in the set OPT that covers points in two adjacent strips in more than one shift partition" at the bottom of [18, p132] would not hold and the approximation ratio of the original shifting strategy would not continue to hold. To apply the standard shifting strategy [18] when $\ell = 1$, the plane should be divided into strips of width 2 instead of 1. This leads to a runtime of $O(\epsilon^{-1} n^{6\lceil\sqrt{2}/\epsilon\rceil+2})$ instead of the claimed $O(\epsilon^{-1} n^{4\sqrt{2}/\epsilon+2})$.

Table 1 Summary of the best existing results (including the non-outlier case) and our main results. The runtime column suppresses the $O(\cdot)$ notation. The approximation parameter ℓ is always a positive integer. Some of our results are $(\alpha, 1+\delta)$-bicriteria approximations, i.e., they achieve an α-approximation by removing at most $(1+\delta)t$ outliers, where $\delta > 0$ is arbitrary.

	Paper	Problem	Approximation Ratio	Runtime
Unit Square Cover	[14]	PSC, $t=0$ PSC, $t=0$	2 $1+\frac{1}{\ell}$	$n\log n$ $\ell^2 n^{4\ell-1}$
	[12]	PSC	$1+\frac{1}{\ell}$	$\ell n^{4\ell+2}$
	This work	PSC	$(2, 1+\delta)$	$\delta^{-1}nt\log n$
Unit Disk Cover	[14]	StripPDC, $t=0$, $W=2\ell$	1	$n^{4\lceil\sqrt{2}\ell\rceil+1}$
	[9]	StripPDC, $t=0$, $W\le 4/5$ StripRPDC, $t=0$, $W\le 4/5$	1 1	n^{13} n^7
	[10]	PDC, $t=0$	2.8334	$n(\log n\log\log n)^2$
	[3]	PDC, $t=0$	4	$n\log n$
	[5]	PDC, $t=0$	7	n
	[11]	PDC	$(1+\frac{1}{\ell})^2$	$\ell n^{2\lceil\sqrt{2}\ell\rceil^2+1}$
	[12]	PDC	$1+\frac{1}{\ell}$	$\ell n^{6\lceil\sqrt{2}\ell\rceil+2}$
	This work	PDC StripPDC, $W\le 4/5$ RPDC StripRPDC, $W\le\sqrt{5}/3$	$(\frac{7}{2}, 1+\delta)$ 1 $(1+\frac{6}{\sqrt{5}}+\frac{1}{\ell}, 1+\delta)$ 1	$n^7t+\delta^{-1}nt\log n$ n^7t $\ell n^4t+\delta^{-1}\ell nt\log n$ n^4t

The strip variants are motivated with the hope to obtain an algorithm of a better approximation ratio when some restriction is imposed on the input set, which is possible since the VC dimension might be smaller [16]. Once a local algorithm for the strip version is obtained, a natural idea for solving the full problem is to first partition the plane into strips and then merge the local within-strip solutions by the shifting strategy introduced in [18].

A challenge is that we need to determine the number of outliers on each strip in order to run the local algorithm. In Section 3, we introduce a new shifting strategy to overcome such challenge. We also prove in Theorem 1 that merging the local solutions will only incur a small multiplicative loss on the approximation ratio.

For notational convenience, we denote a solution to PSC, PDC, RPDC, StripPSC, StripPDC and StripRPDC by $sol_S(X,t)$, $sol_D(X,t)$, $sol_R(X,t)$, $sol_S(X,W,t)$, $sol_D(X,W,t)$ and $sol_R(X,W,t)$, respectively, and the optimal solution by $opt_S(X,t)$, $opt_D(X,t)$, $opt_R(X,t)$, $opt_S(X,W,t)$, $opt_D(X,W,t)$ and $opt_R(X,W,t)$, respectively.

Our Results. We summarize the existing results and our main results in Table 1.

We develop a shifting strategy that approximates the optimal number of outliers on each strip in Section 3. With the new shifting strategy, we give an $O(\delta^{-1}nt\log n)$-time bicriteria approximation algorithm to PSC, which outputs at most $2\cdot opt(X,t)$ unit squares covering at least $n-(1+\delta)t$ of the given points. This can be viewed as an extension of the 2-approximation $O(n\log n)$-time result in [14] to the outlier case without compromising the time complexity for small t and constant δ.

For the strip variants of the disk cover, we give an $O(n^7t)$-time exact algorithm for StripPDC when $W\le 4/5$, improving on the best known non-outlier result of $O(n^{13})$ time, and an $O(n^4t)$-time exact algorithm for StripRPDC when $W\le\sqrt{5}/3$, also improving on the best known runtime of $O(n^7)$. These improvements are close to quadratic for small t.

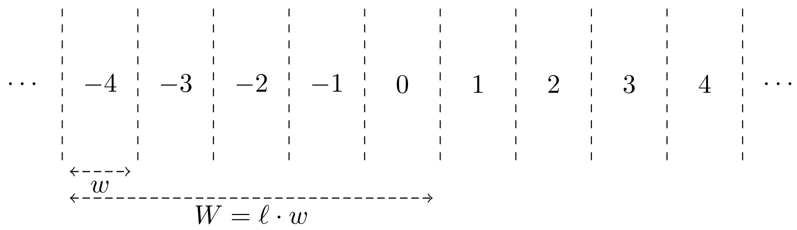

Figure 1 The plane is divided into vertical strips of width w, indexed by integers, and ℓ strips are grouped into a wider one of width W. The figure shows an example of $\ell = 5$.

For the original problem of PDC, based on our new results for the strip variant and the new shifting strategy, we show a 3.5-approximation algorithm with runtime $O(n^7 t)$. This is a new trade-off between the approximation ratio and the running time, and is so far the best running time for an approximation ratio less than 4 for PDC. In the same spirit, we show a $(1 + 6/\sqrt{5} + \epsilon)$-approximation algorithm for RPDC with a runtime of $O(n^4 t/\epsilon)$, where the polynomial dependence on n has a constant exponent, independent of ϵ.

We also extend the previous 4-approximation algorithm [3] for the unit disc cover problem to the outlier case with the same $O(n \log n)$ running time. See Appendix C.

2 Organization of the Paper

The high-level approach to solve PSC, PDC and RPDC follows a divide-and-conquer paradigm known as the shifting strategy [18]. We divide the plane into strips of equal width, run a local algorithm to solve the subproblem on every strip and then merge the local solutions. The main challenge is to determine the number of outliers within each strip. Inspired by [15], we develop a new shifting strategy in the presence of outliers that can in $O(tn \log n)$ time approximate the number of outliers on each strip. We present this new shifting strategy in Section 3 and then derive a 2-approximation to PSC in Section 4. The disk cover problems are discussed in Section 5. We state several new geometric observations in Section 5.1, upon which we design polynomial-time local algorithms that output exact solutions to StripPDC and StripRPDC in Section 5.2. Finally in Section 5.3, with the local algorithms and the new shifting strategy, we obtain one bicriteria algorithm for PDC and one for RPDC.

3 A Shifting Strategy Compatible with Outliers

The shifting strategy introduced by Hochbaum and Maass in [18] has been widely employed in the problem of geometric cover [14, 4, 8, 22, 1, 10, 20, 13]. The strategy requires a partition of the plane and a local algorithm for each single part of the partition. It runs the local algorithm for each part and merges the local solutions with only a small loss on the final approximation ratio. However, in the presence of outliers, we have to determine the number of outliers distributed to each part. Therefore, in this section, we develop a new shifting strategy that can approximate the number of outliers on each strip with provable guarantees.

We now illustrate the shifting strategy for PSC, PDC and RPDC with t outliers. Suppose that the plane is divided into (infinitely many) vertical strips of width w ($w \leq 1$), indexed by integers, say, $\dots, -2, -1, 0, 1, 2, \dots$ from left to right. There are in total ℓ ways $G_1, G_2, \dots, G_\ell$ to group ℓ consecutive strips, where $G_i = \{[k \cdot \ell + i,\ k \cdot \ell + \ell + i - 1] \mid k \in \mathbb{Z}\}$, $i \in \{0, 1, 2, \dots, \ell - 1\}$, resulting in the plane's being divided into wider strips of width $\ell \cdot w$. See Figure 1 for an illustration. To determine the number of outliers in each wider strip, we combine the shifting strategy [18] and the idea from [15]. We use i for the grouping index and j for the index for non-empty strip from left to right in a grouping G_i.

▶ **Theorem 1.** *With a local algorithm $\mathcal{A}$ to a strip of width $\ell \cdot w \leq 1$, we can find a solution*

1. $sol_S(X, \lfloor (1+\delta)t \rfloor) \leq \tau \cdot \left(1 + \frac{\lceil 1/w \rceil}{\ell}\right) \cdot opt_S(X,t)$ *for PSC;*
2. $sol_D(X, \lfloor (1+\delta)t \rfloor) \leq \tau \cdot \left(1 + \frac{\lceil 2/w \rceil}{\ell}\right) \cdot opt_D(X,t)$ *for PDC;*
3. $sol_R(X, \lfloor (1+\delta)t \rfloor) \leq \tau \cdot \left(1 + \frac{\lceil 2/w \rceil}{\ell}\right) \cdot opt_R(X,t)$ *for RPDC;*

where τ denotes the approximation ratio of $\mathcal{A}$ and $\delta > 0$ is arbitrary.

Proof. Let s_i ($s_i \leq n$, $i = 1, 2, \ldots, \ell$) denote the number of non-empty strips of width $\ell \cdot w$ in G_i. These strips in G_i are denoted as $S_{i,j}$, $j = 1, 2, \ldots, s_i$ from left to right. Define $X_{i,j} = X \cap S_{i,j}$. We apply the local algorithm $\mathcal{A}$ to obtain a solution $sol(X_{i,j}, q)$ for each $q \in I$ where $I = \{\lfloor (1+\delta)^r \rfloor | r = 0, 1, 2, \ldots, \lfloor \log_{1+\delta} t \rfloor\} \cup \{0, \lfloor (1+\delta)t \rfloor\}$. We also define a function $f_{i,j}$ on $\{0, 1, 2, \ldots, \lfloor (1+\delta)t \rfloor\}$, where $f_{i,j}(q)$ ($q \in \{0, 1, 2, \ldots, \lfloor (1+\delta)t \rfloor\}$) is defined to be the value of the lower convex hull of $\{(q, sol(X_{i,j}, q)) \,|\, q \in I\}$. The summation $F_i(q_{i,1}, q_{i,2}, \ldots, q_{i,s_i}) = \sum_{j=1}^{s_i} f_{i,j}(q_{i,j})$, $\sum_{j=1}^{s_i} q_{i,j} \leq \lfloor (1+\delta)t \rfloor$ is a convex function.

The minimum point of F_i can be found by going down along the edges of the convex polygonal surface whose vertices are $(q_{i,1}, q_{i,2}, \ldots, q_{i,s_i}, F_i(q_{i,1}, q_{i,2}, \ldots, q_{i,s_i}))$, $\sum_{j=1}^{s_i} q_{i,j} \leq \lfloor (1+\delta)t \rfloor$. The details are presented in Algorithm 1. We denote the minimum point by $(t_{i,1}, t_{i,2}, \ldots, t_{i,s_i})$. We also let $t^*_{i,j}$ denote the number of outliers in $X_{i,j}$ for the optimal solution $opt(X,t)$, and $t'_{i,j}$ be the power of $1+\delta$ between $t^*_{i,j}$ and $\lfloor (1+\delta)t^*_{i,j} \rfloor$. The solutions on each strip are put together to get

$$sol_i(X, \lfloor (1+\delta)t \rfloor) := \sum_{j=1}^{s_i} sol(X_{i,j}, t_{i,j}).$$

As $\sum_{j=1}^{s_i} t'_{i,j} \leq \sum_{j=1}^{s_i} \lfloor (1+\delta)t^*_{i,j} \rfloor \leq \lfloor (1+\delta)t \rfloor$, we then have

$$\sum_{j=1}^{s_i} sol(X_{i,j}, t_{i,j}) = F_i(t_{i,1}, t_{i,2}, \ldots, t_{i,s}) \leq F_i(t'_{i,1}, t'_{i,2}, \ldots, t'_{i,s_i}) \leq \sum_{j=1}^{s_i} sol(X_{i,j}, t'_{i,j}).$$

Moreover $opt(X_{i,j}, t'_{i,j}) \leq opt(X_{i,j}, t^*_{i,j})$ as $opt(X_{i,j}, q)$ is a decreasing function on q. We thus have

$$sol_i(X, \lfloor (1+\delta)t \rfloor) \leq \sum_{j=1}^{s_i} sol(X_{i,j}, t'_{i,j}) \leq \tau \cdot \sum_{j=1}^{s_i} opt(X_{i,j}, t'_{i,j}) \leq \tau \cdot \sum_{j=1}^{s_i} opt(X_{i,j}, t^*_{i,j}).$$

We claim that a unit square can cross at most $\lceil 1/w \rceil + \ell$ strips of width $\ell \cdot w$ in $G_1 \cup \cdots \cup G_\ell$. The proof is deferred to Lemma 5. This indicates that for unit square

$$\sum_{i=1}^{\ell} sol_i(X, \lfloor (1+\delta)t \rfloor) \leq \tau \cdot \sum_{i=1}^{\ell} \sum_{j=1}^{s_i} opt(X_{i,j}, t^*_{i,j}) \leq \tau \cdot (\lceil 1/w \rceil + \ell) \cdot opt(X,t).$$

We finally get a solution

$$sol(X, \lfloor (1+\delta)t \rfloor) = \min_{i=1,\ldots,\ell} sol_i(X, (1+\delta)t) \leq \tau \cdot \left(1 + \frac{\lceil 1/w \rceil}{\ell}\right) \cdot opt(X,t).$$

For unit disks, we shall prove in Lemma 5 that a unit disk can cross at most $\lceil 2/w \rceil + \ell$ strips in $G_1 \cup \cdots \cup G_\ell$ and a similar argument as above yields a solution with an approximation factor of $\tau \cdot \left(1 + \frac{\lceil 2/w \rceil}{\ell}\right)$. ◀

Algorithm 1 Local algorithm for every non-empty strip in a grouping.

```
procedure SHIFTING(X, t, A, G_i)          ▷ X a set of n points, 0 ≤ t < n, G_i a grouping
    s ← number of nonempty strips in G_i                                          ▷ s ≤ n
    I ← {⌊(1+δ)^r⌋ | r = 0,1,2,...,⌊log_{1+δ} t⌋} ∪ {0, ⌊(1+δ)t⌋}
    for j = 1,2,...,s do
        X_j ← points of X that are on the j-th nonempty strip
        Compute the lower convex hull of {(q, A(X_j, q))|q ∈ I}  ▷ A is the local algorithm
        for t_j = 0,1,...,⌊(1+δ)t⌋ do
            f_j(t_j) ← the value of the lower convex hull at t_j, as defined in text
    q_1 ← 0, q_2 ← 0, ..., q_s ← 0
    T ← [(f_j(t_j) − f_j(t_j − 1), j, t_j) |1 ≤ j ≤ s, 1 ≤ t_j ≤ ⌊(1+δ)t⌋]
    Sort T according to the partial ordering ≼
    for k = 1,2,...,⌊(1+δ)t⌋ do
        j ← the index such that T[k] = (f_j(t_j) − f_j(t_j − 1), j, t_j)
        q_j ← q_j + 1
    return (q_1, q_2, ..., q_s)
```

An algorithm to find the minimum point of $F_i(q_{i,1}, q_{i,2}, \ldots, q_{i,s_i})$ subject to $\sum_{j=1}^{s_i} q_{i,j} \leq \lfloor(1+\delta)t\rfloor$ is given in [15], which we reproduce in Algorithm 1. In the remaining of this subsection, we focus on explaining the algorithm that outputs the minimum of F_i in one grouping and thus omit the grouping index i in all the subscripts. For example, F_i becomes F and $f_{i,j}$ becomes f_j. Also for convenience, we have the following definition.

▶ **Definition 2.** *Suppose $g_j(q) := f_j(q) - f_j(q-1)$, then we define a partial ordering such that $g_{j_1}(q_1) \preccurlyeq g_{j_2}(q_2)$ if one of the following conditions is satisfied*
1. $f_{j_1}(q_1) - f_{j_1}(q_1 - 1) < f_{j_2}(q_2) - f_{j_2}(q_2 - 1)$
2. $j_1 < j_2$ *and* $f_{j_1}(q_1) - f_{j_1}(q_1 - 1) = f_{j_2}(q_2) - f_{j_2}(q_2 - 1)$
3. $j_1 = j_2$ *and* $q_1 < q_2$ *and* $f_{j_1}(q_1) - f_{j_1}(q_1 - 1) = f_{j_2}(q_2) - f_{j_2}(q_2 - 1)$

With the partial ordering on $\{g_j(q)|1 \leq j \leq s, 1 \leq q \leq (1+\delta)t\}$, we restate the algorithm in [15] below. The algorithm can be regarded as a discrete version of gradient descent, at each step of which we go down along the steepest direction in which the function value decreases the most. The correctness of Algorithm 1 is ensured by the following two lemmata.

▶ **Lemma 3.** *At the beginning of the k-th iteration in Algorithm 1, we have $q_j = t_j - 1$ where $(f_j(t_j) - f_j(t_j - 1), j, t_j) = T[k]$.*

Proof. For any $t'_j < t_j$, we have $f_j(t'_j) - f_j(t'_j - 1) < f_j(t_j) - f_j(t_j - 1)$ by the convexity of f_j. Therefore $f_j(t'_j) - f_j(t'_j - 1)$ must be ahead of $f_j(t_j) - f_j(t_j - 1)$ under the partial ordering $\preccurlyeq$. This indicates that the update $q_j \leftarrow q_j + 1$ has been executed $(t_j - 1)$ times before. As the initial value of q_j is 0, then at the k-th iteration $(f_j(t_j) - f_j(t_j - 1), j, t_j) = T[k]$, it must be true that $q_j = t_j - 1$. ◀

We defer the proof of the following lemma to Appendix A for completeness.

▶ **Lemma 4** (Lemma 3.3 [15])**.** *Algorithm 1 outputs* $\min f(q_1, \ldots, q_s)$ *subject to* $\sum_{j=1}^{s} q_j \leq \lfloor(1+\delta)t\rfloor$.

We simply select the minimum among all $sol_i(X, \lfloor(1+\delta)t\rfloor)$, $1 \leq i \leq \ell$ as our final solution. To prove its approximation ratio, we need the following lemma.

▶ **Lemma 5.** *Suppose that $\ell \cdot w \leq 1$. If the strip boundary lines in the plane partition do not cross any point of X, then a unit square can cover points of X distributed in at most $\lceil 1/w \rceil + \ell$ different strips of width $\ell \cdot w$ in $G_1 \cup \cdots \cup G_\ell$, and a unit disk at most $\lceil 2/w \rceil + \ell$.*

Proof. Let S denote the leftmost and S' the rightmost strip of width $\ell \cdot w$ that intersects a unit square. Also let i_1 and i_2 denote the indices of the left boundary line of S and S' respectively. Then the index of the right boundary line of S is $i_1 + \ell$. Note that the distance between the right boundary line of S and the left boundary line of S' is $(i_2 - i_1 - \ell) \cdot w$ and must be smaller than 1. Therefore, we have $i_2 - i_1 < \ell + 1/w$. Since i_1, i_2 are integers, this implies that $i_2 - i_1 \leq \ell + \lceil 1/w - 1 \rceil$. We then conclude a unit square can intersect at most $\ell + \lceil 1/w - 1 \rceil + 1 = \ell + \lceil 1/w \rceil$ strips of width $\ell \cdot w$ in $G_1 \cup \cdots \cup G_\ell$. A similar argument works for unit disks. ◀

The algorithm to partition the plane into vertical strips of width w is presented in Appendix D. It guarantees that no boundary line crosses a point in X. It remains to develop local algorithms for the StripPSC, StripPDC and StripRPDC on strip of width $W = \ell \cdot w$.

4 Square Cover

In this section, we illustrate the application of Theorem 1 to the partial square cover problem. We first consider the local problem StripPSC with $W = 1$. For convenience, we define the notion of anchored squares as follows. Note that our definition is different from that in [18].

▶ **Definition 6** (Anchored Square). *For a strip of width $W = 1$ and a point set X within the strip, a square is anchored if its left and right sides coincide with the left and right boundary lines of the strip, respectively, and its upper side crosses a point of X.*

As a unit square L can be translated to an anchored one L' so that $L \cap X \subseteq L' \cap X$, we therefore only consider the anchored squares in StripPSC when $W = 1$. There are n of them. Without loss of generality, we can assume that no two points in X have the same y-coordinates, otherwise we simply rotate the plane. We sort X in the increasing order of the y-coordinates, say, $X_1, \ldots, X_n$. Let L_i denote the anchored square with X_i on its upper side and B_{ji} $(j < i)$ denote the number of points above the upper side of L_j and below the lower side of L_i. Let $N[i][k]$ denote the minimum size of a set of squares that covers X_i $(1 \leq i \leq n)$ and at least $(i - k)$ $(0 \leq k \leq t)$ points from X_1 to X_i, then we have the following recursive formula for $N[i][k]$.

$$N[i][k] = \min_{j<i} N[j][k - B_{ji}] + 1. \tag{1}$$

Note that we require X_i to be covered in an optimal cover of $N[i][k]$. Therefore it is only necessary to consider in (1) those j's such that X_j is not covered by L_i and $B_{ji} \leq k$. Let h_i denote the index of the highest point below the lower side of L_i, then only those $j \in [h_i - k, h_i]$ would be considered. As $k \leq t$, there are at most t such candidate j's. Computing all h_i takes $O(n)$ time and we can store these values. We also note that $B_{ji} = h_i - j$ for those $j \in [h_i - k, h_i]$. Therefore the recursive formula (1) can be rewritten as

$$N[i][k] = \min_{h_i - k \leq j \leq h_i} N[j][k - h_i + j] + 1.$$

This is the base of our local algorithm that outputs an exact solution to StripPSC when $W = 1$, which we present in Algorithm 2.

Algorithm 2 The algorithm that outputs an optimal solution to StripPSC.

procedure $\textsc{SquareLocal}(X,\, t)$ $\quad\triangleright\ 0 \le t \le n/2$
  $X \leftarrow$ set of planar points
  $n \leftarrow |X|$
  sort X so that the y-coordinates of the points are increasing
  $L_i \leftarrow$ the anchored square with X_i on its upper side
  $h_i \leftarrow$ the index of the highest point of X below L_i
  Initialize array $N[j][k] \leftarrow \infty$ for $1 \le j \le n$ and $0 \le k \le t$
  for $i = 1, 2, \dots, n$ **do**
    for $k = 0, 1, \dots, t$ **do**
      $N[i][k] \leftarrow \min_{j \in [h_i - k, h_i]} N[j][k - h_i + j] + 1$
  return $\min_{k \in [0,t]} N[n-k][t-k]$

▶ **Lemma 7.** *There exists an exact algorithm to StripPSC in time $O(nt \log n)$ when $W = 1$.*

Proof. It takes $O(n)$ time to compute all h_i and $O(nt)$ time to compute all B_{ij} for $j \in [h_i - t, h_i]$. For each $k = 0, \dots, t$, we maintain a data structure of the dynamic range minimum query (RMQ) for the one-dimensional array $N[1][k], \dots, N[n][k]$. The dynamic RMQ structure in [17] supports both update and query in $O(\log n)$ time. For each $i = 1, \dots, n$ and $k = 0, \dots, t$, we update the data structure once and query the data structure once. Therefore, filling the $N[i][k]$ array takes time $O(nt \log n)$ in total and the overall runtime is thus $O(nt \log n)$. ◀

▶ **Theorem 8.** *For PSC, there exists an $O(\delta^{-1} nt \log n)$-time algorithm which outputs $2 \cdot opt(X,t)$ unit squares that cover at least $n - (1+\delta)t$ of the given points.*

Proof. Note that in Algorithm 2, we have computed all $N[i][k]$ for $1 \le i \le n$ and $1 \le k \le t$, that is, we know the optimal solution to StripPSC that cover $n - k$ points of X for all $k = 0, \dots, t$ when $W = 1$. With Algorithm 2 as the local algorithm $\mathcal{A}$ and $\ell = 1$, we can run Algorithm 1 in $O(nt \log_{1+\delta} n) = O(\delta^{-1} nt \log n)$ time and obtain a solution of size $\tau \cdot \left(1 + \frac{\lceil 1/w \rceil}{\ell}\right) \cdot opt(X,t) = 2 \cdot opt(X,t)$ that covers at least $n - (1+\delta)t$ points of X. In total, the time complexity is $O(\delta^{-1} nt \log n)$. ◀

5 Disk Cover

In this section, we study the disk cover problems PDC and RPDC. We first introduce the notions and definitions in Section 5.1. We also prove a few lemmata which are critical for developing the local algorithms to StripPDC and StripRPDC. In Section 5.2 we describe the local algorithm in details. And finally in Section 5.3, we solve PDC and RPDC by the local algorithm in Section 5.2 and the shifting strategy we develop in Section 3 together.

5.1 Geometric Observations

When the disk centers are unrestricted, it is equivalent to considering only the unit disks whose boundary cross two points of X. Such unit disks are defined as the anchored disks in [11, 12]. In this work, we extend this notion to include the unit disks with a point in X as its highest or lowest point and give the formal definition below. We shall only consider the anchored disks for PDC and StripPDC in the rest of the paper.

▶ **Definition 9** (Anchored Unit Disks). *Given a point set X in the plane, a unit disk is anchored if either there are two points of X on its boundary, or the highest, or the lowest point of the unit disk is in X.*

▶ **Lemma 10.** *It is sufficient to consider only the anchored unit disks in PDC and StripPDC. There are at most $n^2 + n$ anchored disks.*

Proof. For any unit disk D, let D' be the lowest disk such that $D' \cap X \supseteq D \cap X$. We can prove by contradiction that there are two points of X on the boundary of D' or the highest point of D' is in X. If there is no point of X on the boundary of D', then we can translate D' downwards by a small value such that the translated disk still covers $D' \cap X \supseteq D \cap X$. If there is only one point of X on the boundary of D' and the point is not at the highest position of D', we can rotate D' around this point by a small angle so that the y-coordinate of the disk center decreases and the rotated disk still covers $D' \cap X \supseteq D \cap X$. Either of two cases would result in a contradiction. There are at most $\frac{(n-1)n}{2} \cdot 2 + n = n^2$ of such disks. We also include the disks whose lowest point is in X and it total there are $n^2 + n$ of them. ◀

Before presenting the observations related to strip geometric cover, we introduce some basic notions and lemmata which are helpful for the readers to understand the results. Definition 11 is inspired by the mutually spanning set in [9], in which one unit disk is supposed to cover a nonempty subset of X both above and below any other unit disk. However, the definition of mutually spanning set is too strong and unnecessary. We therefore revise it into the top spanning set.

▶ **Definition 11** (Top Spanning Set). *Suppose the points in X lie in a vertical strip of width $w < 1$. A set of unit disks is top spanning if the set is either a singleton, or each unit disk other than the highest one covers a nonempty subset of X above the highest disk.*

▶ **Definition 12** (Inscribed Rectangles [9]). *A unit disk centered in a strip of width $W < 1$ covers a strip segment of height at least $2\sqrt{1 - W^2}$. The strip segment is referred to as the inscribed rectangle of the unit disk.*

An illustration of the inscribed rectangle is shown in Figure 2.

▶ **Definition 13** (Vertical Span). *Given a strip and a set of unit disks $\{D_1, \ldots, D_m\}$, the vertical span is defined to be the height difference between the highest and lowest points of the strip covered by $\bigcup_{i=1}^{m} D_i$.*

The following lemma was proved for the mutually spanning set in [9] and it is still true for the top spanning set. We reproduce a proof in Appendix B.

▶ **Lemma 14** ([9]). *Consider unit disks D_1, D_2 whose centers o_1, o_2 are in a vertical strip of width $W < 1$. If $y(o_1) \geq y(o_2)$ and D_2 covers some point above D_1, then we conclude that $y(o_1) - y(o_2) \leq 1 - \sqrt{1 - W^2}$. Further, the span of a top spanning set is at most $3 - \sqrt{1 - W^2}$.*

We are now ready for proving a few geometric observations. In [9], the authors studied the within strip unit disk cover where the points are within a given strip, and the unit disks can only be selected from a finite set $\mathcal{D}$ where the disk centers are also witihn the strip. Let X' consist of points in X that are covered by the inscribed rectangles of the disks in $\mathcal{D}$. In [9, Lemma 5], it is proved that among all $C \subseteq \mathcal{D}$ that covers X', there exists one covering of the minimum size and the covering does not contain any mutually spanning set of more than 3 disks. In a similar approach, we can prove the following two lemmata.

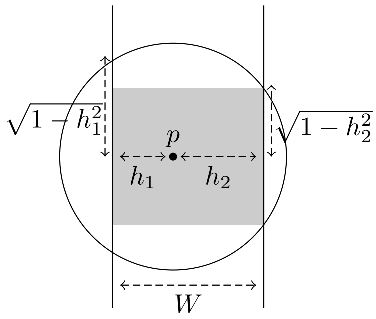

Figure 2 The figure illustrates the inscribed rectangle. The two vertical lines are the left and right boundary lines of a strip of width W. The center point p of the unit disk is inside the strip. The distance from p to the left boundary line is denoted by h_1, and the distance to the right boundary line is denoted by h_2. The shadowed area is the inscribed rectangle whose height is $2\min\{\sqrt{1-h_1^2}, \sqrt{1-h_2^2}\} \geq 2\sqrt{1-W^2}$.

▶ **Lemma 15.** *When $W \leq \frac{4}{5}$, there exists an optimal solution on* StripPDC *that contains no top spanning set of more than* 2 *disks.*

Proof. Assume that *opt* is an optimal solution and contains a top spanning set of 3 disks, say, D_1, D_2 and D_3, from bottom to top. Let p_1 be the lowest point of X covered by $D_1 \cup D_2 \cup D_3$ and p_2 the highest. By Lemma 14, we know the vertical span of $D_1 \cup D_2 \cup D_3$ is at most $3 - \sqrt{1-W^2}$. Let D_1' be the unit disk with p_1 as its lowest point and D_2' be the unit disk with p_2 as its highest point. Then D_1' covers a segment of length at least $2\sqrt{1-W^2}$ above p_1, and D_2' covers a segment of length at least $2\sqrt{1-W^2}$ below p_2. Since $2 \times 2\sqrt{1-W^2} \geq 3 - \sqrt{1-W^2}$ when $W \leq \frac{4}{5}$, we can replace $\{D_1, D_2, D_3\}$ with $\{D_1', D_2'\}$ and obtain a smaller solution, contradicting the optimality of *opt*. Therefore *opt* does not contain any top spanning set of 3 unit disks. ◀

▶ **Lemma 16.** *When $W \leq \frac{\sqrt{5}}{3}$, there is an optimal solution on* StripRPDC *that contains no top spanning set of more than* 2 *disks.*

Proof. Assume *opt* is an optimal cover and contains a top spanning set of 3 unit disks. Let D_1, D_2, D_3, p_1 and p_2 be as in the proof of Lemma 15. Besides, by o_1, o_2 and o_3 we denote the centers of D_1, D_2 and D_3, respectively. From Lemma 14 we know $y(o_3) - y(p_1) \leq y(o_3) - y(o_1) + 1 \leq 2 - \sqrt{1-W^2}$. Let p denote the highest point not covered by D_3 and we have $y(o_3) - y(p) \geq \sqrt{1-W^2}$. The unit disk D centered at p covers a strip segment of length $\sqrt{1-W^2}$ below p, and hence $D \cup D_3$ covers a strip segment of length $2\sqrt{1-W^2}$ below o_3. When $W \leq \frac{\sqrt{5}}{3}$, we have $2\sqrt{1-W^2} \geq 2 - \sqrt{1-W^2}$ and $(D_1 \cup D_2 \cup D_3) \cap X \subseteq (D \cup D_3) \cap X$. Replacing $D_1 \cup D_2 \cup D_3$ with $D \cup D_3$ would give us a smaller cover, which contradicts the optimality of *opt*. ◀

5.2 Exact Algorithms to StripPDC and StripRPDC

In this subsection, we develop an exact algorithm for StripPDC from Lemma 15. The detailed description is in Algorithm 3. In the same way, we can develop an exact algorithm for StripRPDC from Lemma 16. Let $\mathcal{D}$ denote the set of unit disks which are candidates in an optimal covering. For StripPDC they are the anchored disks as defined in Definition 9 and for StripRPDC they are the unit disks centered in the point set X. We first state our main theorem below.

Algorithm 3 The algorithm that outputs an optimal solution to StripPDC.

```
procedure DiskLocal(X, t)
    U[i][k] ← ∅ for i = 0, 1, ..., n and k = 0, 1, 2, ..., n − t
    G[i][k] ← ∅ for i = 1, 2, ..., n and k = 0, 1, 2, ..., n − t
    for i = 1, 2, ..., n do
        for k = max(i − t, 0), max(i − t, 0) + 1, ..., min(i − 1, n − t) do
            for T ∈ U[i − 1][k] do
                if the disks in T cover X_i then
                    G[i][k + 1] ← G[i][k + 1] ∪ {T}
                else
                    G[i][k] ← G[i][k] ∪ {T}
                    for D ∈ 𝒟 do
                        if D covers X_i then
                            G[i][k + 1] ← G[i][k + 1] ∪ {T ∪ {D}}
        for k = max(i − t, 0), max(i − t, 0) + 1, ..., min(i, n − t) do
            for T ∈ G[i][k] do
                if |T| ≤ 2 then
                    U[i][k] ← U[i][k] ∪ {T}
            for T_1 ∈ U[i][k] do
                for T_2 ∈ U[i][k] do
                    if sig(T_1) = sig(T_2) and |T_1| ≤ |T_2| then
                        U[i][k] ← U[i][k] − {T_2}
    return any T ∈ U[n][n − t]
```

▶ **Theorem 17.** *Algorithm 3 computes an exact solution to StripPDC in $O(n^7 t)$ time when $W \leq 4/5$. If the set of anchored disks is replaced by the set of unit disks centered in X, Algorithm 3 would output an exact solution to StripRPDC in $O(n^4 t)$ time.*

We follow the dynamic programming introduced in [12, Section 3] to develop a local algorithm. We assume no two points of X have the same y-coordinates, otherwise we can rotate X. We also sort X in an increasing order of their y-coordinates. Let X_1, X_2, $\ldots$, X_n denote the sorted points from bottom to top. A subcover is denoted by (k, i, T) where T is the set of the disks that cover at least k points between X_1 and X_i. The signature of (k, i, T), denoted by $sig(k, i, T)$, consists the highest disk D of T and those in T which cover at least one point of X above D.

Now we state the idea of the dynamic programming. Let $U[i][k]$ store all the subcovers that cover at least k points at the i-th step in the algorithm. We iterate over all $T \in U[i][k]$. If T already covers X_{i+1}, we simply add T into $U[i+1][k+1]$. Otherwise, we add T to $U[i+1][k]$ and all possible $T \cup \{D\}$ to $U[i+1][k+1]$, where $D \in \mathcal{D}$ is a unit disk that covers X_{i+1}. At the end of the i-th iteration, for two subcovers T_1 and T_2 in $U[i+1][k]$, if $sig(T_1) = sig(T_2)$ and $|T_1| \leq |T_2|$, we remove T_2 from $U[i+1][k]$. By Lemma 15, there is an optimal solution to StripPDC that contains no top spanning set of more than 2 disks. It is obvious that the signature is a top spanning set by Definition 11, we therefore remove T if $|sig(T)| > 2$.

Proof of Theorem 17. The correctness of Algorithm 3 is guaranteed by Lemmata 1 and 2 in [12]. For any $T \in U[i][k]$, we have $|sig(T)| \leq 2$ and therefore there are $O(n^4)$ different signatures. There are no two subcovers with the same signature in $U[i][k]$, so $|U[i][k]| = O(n^4)$.

Observe that $U[i][k]$ is nonempty for at most t values of k, it holds that $\sum_k |U[i][k]| = O(n^4 t)$. Furthermore, since the number of disks that cover X_i is $O(n^2)$, we have $|G[i][k]| = O(n^6)$ and further $\sum_k |G[i][k]| = O(n^6 t)$. After we construct $G[i][k]$, we only select some of them into $U[i][k]$. The would cost $O(n \cdot n^6 t)$ time. Besides, we also remove the larger covering of the sane signature in $U[i][k]$. The process can be done in linear time with respect to $|U[i][k]|$, as shown in [12] with the techniques from [14]. The overall time complexity is then $O(n \cdot t \cdot n^4 \cdot n^2 + n \cdot t \cdot n^6) = O(n^7 t)$.

The same algorithm can be applied to StripRPDC and the only difference is that we use the unit disks centered in X instead as the candidates in a covering. There are n such disks and by Lemma 16 there are $O(n^2)$ different signatures of size at most 2. Also we have $U[i][k] = O(n^2)$ and $G[i][k] = O(n^3)$ for StripRPDC. The overall time complexity would be $O(n \cdot t \cdot n^2 \cdot n + n \cdot t \cdot n^3) = O(n^4 t)$. ◀

5.3 Approximation Algorithms to PDC and RPDC

We apply Algorithm 3 as the local algorithm $\mathcal{A}$ for PDC and RPDC. Combining with Theorem 1, we obtain a global algorithm to PDC with approximation ratio $1 \cdot \left(1 + \frac{\lceil 2/w \rceil}{\ell}\right) = 3.5$ for $w = 0.4$ and $\ell = 2$, and a global algorithm to RPDC whose approximation factor is $1 \cdot \left(1 + \frac{\lceil 2/w \rceil}{\ell}\right) \leq 1 + \frac{2}{\ell \cdot w} + \frac{1}{\ell} = 1 + \frac{6}{\sqrt{5}} + \frac{1}{\ell} \approx 3.68 + \frac{1}{\ell}$.

▶ **Theorem 18.** *There exist a $(3.5, 1+\delta)$-bicriteria algorithm for PDC which runs in time $O(n^7 t + \delta^{-1} nt \log n)$, and a $(1 + \frac{6}{\sqrt{5}} + \frac{1}{\ell}, 1+\delta)$-bicriteria algorithm for RPDC which runs in time $O(\ell n^4 t + \delta^{-1} \ell nt \log n)$.*

Proof. Let $n_j = |X_j|$, the number of points in the j-th nonempty strip of the grouping G_i.

For PDC, we apply Algorithm 3 as the local algorithm $\mathcal{A}$ in Algorithm 1. Note that in Algorithm 3, all $U[n][n-k]$ $(0 \leq k \leq t)$ are computed. Therefore it takes $O(n_j^7 t)$ time to output $sol(X_j, t_j)$ for all $t_j \in \{\lfloor (1+\delta)^r \rfloor \mid r = 0, 1, 2, \ldots, \lfloor \log_{1+\delta} t \rfloor\} \cup \{0, \lfloor (1+\delta)t \rfloor\}$. On all the s_i strips, this would cost $O\left(n_1^7 t + \cdots + n_{s_i}^7 t\right) = O(n^7 t)$ time. Sorting all the values $f_i(t_i) - f_i(t_i - 1)$ in Algorithm 1 takes $O\left(nt \log_{1+\delta} nt\right) = O\left(\delta^{-1} nt \log n\right)$ time. As there are ℓ groupings, the total complexity is $O(\ell n^7 t + \delta^{-1} \ell nt \log n)$. Letting $\ell = 2$ and $w = 2/5$ yields a 3.5-approximation with time complexity $O(n^7 t + \delta^{-1} nt \log n)$.

For RPDC, we prove in the same way that the overall time complexity is $O(\ell n^4 t + \delta^{-1} \ell nt \log n)$. ◀

References

1. Christoph Ambühl, Thomas Erlebach, Matúš Mihalák, and Marc Nunkesser. Constant-factor approximation for minimum-weight (connected) dominating sets in unit disk graphs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 3–14. Springer, 2006.
2. Rossen Atanassov, Prosenjit Bose, Mathieu Couture, Anil Maheshwari, Pat Morin, Michel Paquette, Michiel Smid, and Stefanie Wuhrer. Algorithms for optimal outlier removal. *Journal of discrete algorithms*, 7(2):239–248, 2009.
3. Ahmad Biniaz, Paul Liu, Anil Maheshwari, and Michiel Smid. Approximation algorithms for the unit disk cover problem in 2D and 3D. *Computational Geometry*, 60:8–18, 2017.
4. Moses Charikar, Chandra Chekuri, Tomás Feder, and Rajeev Motwani. Incremental clustering and dynamic information retrieval. *SIAM Journal on Computing*, 33(6):1417–1440, 2004.
5. Adrian Dumitrescu, Anirban Ghosh, and Csaba D Tóth. Online unit covering in euclidean space. *Theoretical Computer Science*, 809:218–230, 2020.

6 David Eppstein and Jeff Erickson. Iterated nearest neighbors and finding minimal polytopes. *Discrete & Computational Geometry*, 11(3):321–350, 1994.
7 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.
8 Massimo Franceschetti, Matthew Cook, and Jehoshua Bruck. A geometric theorem for approximate disk covering algorithms. Technical Report ETR035, California Institute of Technology, 2001.
9 Robert Fraser and Alejandro López-Ortiz. The within-strip discrete unit disk cover problem. *Theoretical Computer Science*, 674:99–115, 2017.
10 Bin Fu, Zhixiang Chen, and Mahdi Abdelguerfi. An almost linear time 2.8334-approximation algorithm for the disc covering problem. In *International Conference on Algorithmic Applications in Management*, pages 317–326. Springer, 2007.
11 Rajiv Gandhi, Samir Khuller, and Aravind Srinivasan. Approximation algorithms for partial covering problems. *Journal of Algorithms*, 53(1):55–84, 2004.
12 Hossein Ghasemalizadeh and Mohammadreza Razzazi. An improved approximation algorithm for the most points covering problem. *Theory of Computing Systems*, 50(3):545–558, 2012.
13 Anirban Ghosh, Brian Hicks, and Ronald Shevchenko. Unit disk cover for massive point sets. In *International Symposium on Experimental Algorithms*, pages 142–157. Springer, 2019.
14 Teofilo F Gonzalez. Covering a set of points in multidimensional space. *Information processing letters*, 40(4):181–188, 1991.
15 Sudipto Guha, Yi Li, and Qin Zhang. Distributed partial clustering. *ACM Transactions on Parallel Computing (TOPC)*, 6(3):1–20, 2019.
16 Sariel Har-Peled. On complexity, sampling, and ε-nets and ε-samples. *Approximation Algorithm in Geometry*, 2010.
17 Alice Héliou, Martine Léonard, Laurent Mouchard, and Mikael Salson. Efficient dynamic range minimum query. *Theoretical Computer Science*, 656:108–117, 2016.
18 Dorit S Hochbaum and Wolfgang Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *Journal of the ACM (JACM)*, 32(1):130–136, 1985.
19 Tanmay Inamdar. Local search for geometric partial covering problems. In *Proceedings of the 31st Canadian Conference on Computational Geometry*, pages 242–249, 2019.
20 Paul Liu and Daniel Lu. A fast 25/6-approximation for the minimum unit disk cover problem, 2014. `arXiv:1406.3838`.
21 Nina Mishra, Rajeev Motwani, and Sergei Vassilvitskii. Sublinear projective clustering with outliers. In *15th Annual Fall Workshop on Computational Geometry and Visualization*, page 45. Citeseer, 2005.
22 Sada Narayanappa and Petr Vojtechovskỳ. An improved approximation factor for the unit disk covering problem. In *Proceedings of the 18th Canadian Conference on Computational Geometry*, pages 15–18, 2006.
23 Kaveh Pahlavan and Allen H Levesque. Wireless data communications. *Proceedings of the IEEE*, 82(9):1398–1430, 1994.
24 Michael Segal and Klara Kedem. Enclosing k points in the smallest axis parallel rectangle. *Information Processing Letters*, 65(2):95–99, 1998.
25 Steven L Tanimoto and Robert J Fowler. Covering image subsets with patches. In *Proceedings of the fifty-first International Conference on Pattern Recognition*, pages 835–839, 1980.

A Proof of Lemma 4

Proof. Let $(t_1^*, \ldots, t_s^*)$ denote a minimum point of f and it is obvious that $\sum_{j=1}^s t_j^* = \lfloor (1+\epsilon)t \rfloor$. We also let $(q_1, \ldots, q_s)$ denote the output of Algorithm 1 and $\sum_{j=1}^s q_j = \lfloor (1+\epsilon)t \rfloor$. If $(q_1, \ldots, q_s) \neq (t_1^*, \ldots, t_s^*)$, then there must be some $q_{j_1} < t_{j_1}^*$ and $q_{j_2} > t_{j_2}^*$. As q_{j_1} is the final output value of Algorithm 1, $f_{j_1}(t_{j_1}^*) - f_{j_1}(t_{j_1}^* - 1)$ cannot be in the first $\lfloor (1+\epsilon)t \rfloor$ elements of S. Besides, we have $f_{j_2}(t_{j_2}^* + 1) - f_{j_2}(t_{j_2}^*) \leq f_{j_2}(q_{j_2}) - f_{j_2}(q_{j_2} - 1)$ by the convexity of f_{j_2} and

$f_{j_2}(t^*_{j_2}+1)-f_{j_2}(t^*_{j_2})$ must in the first $\lfloor(1+\epsilon)t\rfloor$ elements of S. Therefore $f_{j_2}(t^*_{j_2}+1)-f_{j_2}(t^*_{j_2}) \leq f_{j_1}(t^*_{j_1}) - f_{j_1}(t^*_{j_1}-1)$. This indicates that $(\ldots, t^*_{j_1}-1, \ldots, t^*_{j_2}+1, \ldots)$ is also a minimum point of f and its L_1 distance to $(q_1, \ldots, q_s)$ is less than that of $(t^*_1, \ldots, t^*_s)$. Repeat this process, we can finally prove that the output of Algorithm 1 is a global minimum. ◀

B Proof of Lemma 14

Proof. Let D_1 and D_2 denote the two disks. Without loss of generality, we assume the center of D_1 is higher than that of D_2. The vertical distance between their centers can not exceed $\left(1-\sqrt{1-W^2}\right)$. Otherwise, D_2 would be disjoint from the upper edge of the inscribed rectangle of D_1, and thus cannot cover any point above D_1. The total span of $D_1 \cup D_2$ is therefore at most $1-\sqrt{1-W^2}+1+1 = 3-\sqrt{1-W^2}$. ◀

C 4-Approximation to PDC

In this section, we present a simple 4-approximation to PDC by generalizing the maximal independent set to the outlier case. The definition of partial maximal independent set is presented below.

▶ **Definition 19** (Partial Maximal Independent Set). *Given a point set X of size n and an integer $0 \leq t < n$, a subset $S \subseteq X$ is called a partial maximal independent set if for any distinct points $p, q \in S$, it holds that $d(p,q) > 2$ and $\left|\bigcup_{p \in S} B(p,2) \cap X\right| \geq n-t$.*

The greedy algorithm with time complexity $O(n \log n)$ in [3] can be slightly modified to a 4-approximation algorithm for PDC, which we present in Algorithm 4.

Algorithm 4 Greedy algorithm which outputs a 4-approximation to PDC.

Require: A set X of n planar points and an integer $0 \leq t < n$.
  $Y \leftarrow X$, $S \leftarrow \emptyset$
  Sort Y by x-coordinate
  while $|Y| > t$ **do**
    Find the leftmost uncovered point p and $S \leftarrow S \cup \{p\}$
    Place a right semicircle of radius 2 at p
    Remove the points covered by the semicircle from Y
  return S

▶ **Lemma 20.** *Algorithm 4 returns a 4-approximation solution to PDC in time $O(n \log n)$.*

Proof. It is easy to verify that S is a partial maximal independent set. Furthermore, for any $p, q \in S$, $p \neq q$, since $d(p,q) > 2$, there is no unit disk that can cover both p and q. Therefore a distinct unit disk is needed to cover each point in S, which implies that $|S| \leq opt(X,t)$. Note that four unit disks are sufficient to cover a semicircle with radius 2. Together with $|\bigcup_{p \in S} B(p,2) \cap X| \geq n-t$, we can obtain $4|S|$ unit disks that cover at least $(n-t)$ points of X. Note that $4|S| \leq 4 \cdot opt(X,t)$, we see that $4|S|$ unit disks make up to a 4-approximation. ◀

Although the algorithm is simple, there is a fatal drawback when applying the algorithm from left to right, it can only detect outliers X_i where $i \geq n-t$ and cannot detect the others. The bad case is that some outliers are far away from the other points, and at the same time, their x coordinates are around the median of $\{X_1, X_2, \ldots, X_n\}$. A unit disk covering such an outlier usually covers few or no other points, and not removing such isolated outliers could greatly increase the number of disks in the solution.

D Partitioning the Plane

Algorithm 5 Partitioning the plane into strips such that their boundary lines do not intersect X.

---

procedure PARTITION(X) ▷ X is a finite set of planar points
    $S \leftarrow \emptyset$
    $R \leftarrow \emptyset$
    for $p \in X$ **do**
        $S \leftarrow S \cup \{\lfloor x(p)/w \rfloor\}$
        $R \leftarrow R \cup \{x(p)/w - \lfloor x(p)/w \rfloor\}$
    if $0 \in R$ **then**
        $S \leftarrow \emptyset$
        $\tau \leftarrow \frac{1}{2} \min\{r \in R : r > 0\}$
        for $p \in X$ **do**
            $S \leftarrow S \cup \{\lfloor x(p)/w - \tau \rfloor\}$
    return S
