The k-Colorable Unit Disk Cover Problem

Monith S. Reyunuru[1], Kriti Jethlia[2], and Manjanna Basappa[2(✉)]

[1] Amazon Development Center, Hyderabad, India
f20160006@hyderabad.bits-pilani.ac.in
[2] CSIS Department, Birla Institute of Technology and Science Pilani, Hyderabad Campus, Hyderabad, India
{f20180223,manjanna}@hyderabad.bits-pilani.ac.in

Abstract. We consider the k-Colorable Discrete Unit Disk Cover (k-*CDUDC*) problem as follows. Given a parameter k, a set P of n points, and a set D of m unit disks, both sets lying in the plane, the objective is to compute a set $D' \subseteq D$ such that every point in P is covered by at least one disk in D' and there exists a function $\chi : D' \rightarrow C$ that assigns colors to disks in D' such that for any d and d' in D' if $d \cap d' \neq \emptyset$, then $\chi(d) \neq \chi(d')$, where C denotes a set containing k distinct colors.

For the k-*CDUDC* problem, our proposed algorithms approximate the number of colors used in the coloring if there exists a k-colorable cover. We first propose a 4-approximation algorithm in $O(m^{7k} n \log k)$ time for this problem, where k is a positive integer. The previous best known result for the problem when $k = 3$ is due to the recent work of Biedl et al. [Computational Geometry: Theory & Applications, 2021], who proposed a 2-approximation algorithm in $O(m^{25}n)$ time. For $k = 3$, our algorithm runs in $O(m^{21}n)$ time, faster than the previous best algorithm, but gives a 4-approximate result. We then generalize the above approach to yield a family of ρ-approximation algorithms in $O(m^{\alpha k} n \log k)$ time, where $(\rho, \alpha) \in \{(4, 7), (6, 5), (7, 5), (9, 4)\}$. We also extend our algorithm to solve the k-*Colorable Line Segment Disk Cover (k-CLSDC)* and k-*Colorable Rectangular Region Cover (k-CRRC)* problems, in which instead of the set P of n points, we are given a set S of n line segments, and a rectangular region $\mathcal{R}$, respectively.

Keywords: Colorable unit disk cover · Approximation algorithm · Grid-partitioning

1 Introduction

Our motivation for studying the problem arises from practical applications in the frequency/channel assignment problem in wireless/cellular networks. In ad-hoc mobile networks, each host(station/tower) is equipped with a Radio-Frequency (RF) transceiver to provide reliable transmission inside a circular range, represented by a disk, within some distance. Each wireless client is equipped with corresponding receivers. The clients themselves are represented by a set of points

© Springer Nature Switzerland AG 2021
O. Gervasi et al. (Eds.): ICCSA 2021, LNCS 12949, pp. 402–417, 2021.
https://doi.org/10.1007/978-3-030-86653-2_30

P in a plane. The disks representing the range (which is presumably the same for all stations) of each potential host is represented by the set D. In the spirit of reducing interference in broadcast and other energy-saving measures, we aim to limit or reduce the number of different frequencies(channels) assigned to each, represented by coloring. Typically, (Wi-Fi) networks are built with 3 independent channels [4], hence the motivation for a study on the *3-CDUDC* problem. In the same spirit, we generalize the *3-CDUDC* to the *k-CDUDC* problem, where $k > 0$ is an integer. We further generalize the problem by considering line segments and a continuous rectangular region as representing potential wireless clients (resp. the *k-CLSDC* and *k-CRRC* problems), instead of points.

1.1 Related Work

The *3-CDUDC* problem, to the best of our knowledge, was first studied by Biedl et al., [3]. They gave a 2-approximation algorithm in $O(nm^{25})$ time for the *3-CDUDC* problem. Their approach first partitions the plane into horizontal strips, solves the problem for every strip optimally, then returns the union of solutions of all strips. To solve the problem for any strip they show that at most a constant number of disks of an optimal solution intersect any vertical line. Based on this, they define a directed acyclic graph such that there exists a path from source to a destination corresponding to this optimal solution. In this paper, we attempt to improve upon this impractical $O(nm^{25})$ running time. Our approach, however, focuses on the specific geometric properties that arise from the dual conditionals of the problem statement. Although both of the approaches, initially, begin by dividing the plane, we recognize a unique bound that exists in our need to bound the colorability and provide a novel solution in the same regard.

A notion of *conflict-free coloring (CF-coloring)* was introduced by Even et al. [8]. and Smorodinsky [13]. In the *CF-coloring* problem we are given a set of points (representing client locations) and a set of base stations, the objective is to assign colors (representing frequencies) to the base stations such that any client lying within the range of at least one base station is covered by the base station whose color is different from the colors of the other base stations covering the client, and the number of colors used should be as minimum as possible. Here, the range of base stations is modeled as regions e.g., disks or other geometric objects. Even et al., [8] proved that $O(\log n)$ colors are always sufficient to *CF-color* a set of disks in the plane, and in the worst case, $\Omega(\log n)$ colors are required. Note that this *CF-coloring* of disks is different from our notion of k-colorable disk cover of points. In the former overlapping disks may be given the same color if they don't share a client, whereas in the *k-CDUDC* overlapping disks must be colored with distinct colors. A generalization of *CF-coloring* is called a *k-fault-tolerant CF-coloring*. Cheilaris et al. [7] presented a polynomial-time $(5 - \frac{2}{k})$-approximation algorithm for the *k-fault-tolerant CF-coloring* in 1-dimensional space. Horev et al. [11] proved that $O(k \log n)$ colors are sufficient for any set of n disks in the plane. For *dynamic CF-coloring* and results on *CF-coloring* of other geometric objects, we refer to [5] and references therein.

A related problem [2] to the *k-CDUDC* problem in the literature is the *Discrete Unit Disk Cover* (*DUDC*) problem. In the *DUDC* problem, we are given a set P of n points and a set D of m unit disks, our goal is to select as the smallest number of disks from D as possible such that the union of these selected disks covers all points in P. As in the *k-CDUDC*, here also, the sets P and D can be considered as representing a set of wireless clients and a set of base stations or towers, respectively. The *DUDC* problem is NP-hard and is a very well studied one. There is a polynomial time approximation scheme (PTAS) with impractical running time for this problem [12]. The current best approximation algorithm with reasonable running time is $(9 + \epsilon)$ for any $\epsilon > 0$ [2]. However, a series of approximation algorithms have been proposed for this problem by various authors over the past two decades, and a complete survey on this can be found in [10]. When a line segment is used to represent a potential wireless client, the *DUDC* problem becomes a *Line Segment Disk Cover* (*LSDC*) problem. In a similar line, there is another variant of the *DUDC* problem, a *Rectangular Region Cover (RRC)* problem, in which all the continuous set of points lying in a rectangular region represent wireless clients. All the available results for the *DUDC* problem also extend to the *LSDC* and *RRC* problems [1], with slightly different running time. We also extend our results for the *k-CDUDC* problem to solve the colorable variants of the *LSDC* and *RRC* problems, namely, the *k-CLSDC* and *k-CRRC* problems.

2 The *k*-CDUDC Problem

In this section we consider the following problem.

– *k-Colorable Discrete Unit Disk Cover (k-CDUDC)*: Given a set P of n points, and a set D of m unit disks (of radius=1), both lying in the plane, and a parameter k, the objective is to compute a set $D' \subseteq D$ that covers all points in P such that the set D' can be partitioned into $\{D'_1, D'_2, \ldots, D'_k\}$, where for each $a \in \{1, 2, \ldots, k\}$ the disks in D'_a are pairwise disjoint, i.e., the disks in D' can be colored with at most k colors such that the overlapping disks receive distinct colors and every point in P is covered by a disk in D'.

As it was pointed out in [3] that there is a related problem, namely, *Unit Disk Chromatic Number (UDCN)* problem, that aims to color all nodes in a given unit disk graph with at most k colors. The *UDCN* problem is NP-hard for any $k \geq 3$ [6]. Similar to Biedl et al. [3], we can center a set D of m unit disks in the plane such that there are at least $k+1$ pairwise non-disjoint disks that have a common intersection region and a unit disk graph $G_D = (V_D, E_D)$ induced by D is connected. Let us then place a set P of n points in this intersection region. Now observe that the set P has a cover which is at most k-colorable, whereas the graph G_D is at least $(k+1)$-colorable. Hence, the *k-CDUDC* problem is different from the *UDCN* problem. Biedl et al. [3] showed that the *3-CDUDC* problem is NP-hard by carefully incorporating a set P of n points in the NP-hard proof of the *UDCN* problem with $k = 3$ in [6]. This directly implies that the *k-CDUDC*

is NP-hard since the k-*CDUDC* is a generalization of *3-CDUDC*. It is also easy to see that the k-*CDUDC* problem belongs to the class NP, as follows: Here, the certificate for any Yes instance of k-*CDUDC* is a set of k distinct colors identified by non-negative integers $1, 2, \ldots, k$, and a mapping $\chi : D' \rightarrow \{1, 2, \ldots, k\}$, where $D' \subseteq D$. A polynomial time verifier checks if every point in P is covered by a disk in D' and for every pair of disks $d, d' \in D'$ if $d \cap d' \neq \emptyset$, whether it is the case that $\chi(d) \neq \chi(d')$.

2.1 A 4-Approximate Algorithm

Here, our algorithm is based on partitioning the plane containing points into a grid and then determining bound on the number of unit disks that can participate in any k-colorable covering of points lying within any square of the grid. We first define a grid of width two units that partitions the plane into squared regions. Each of these squared regions is a grid cell with a size 2×2. For simplicity assume no point of P lies on the boundary of these grid cells. Let us associate a unique ID $id_{\mathcal{C}}$ to each grid cell $\mathcal{C}$ as follows; let $p = (x_p, y_p)$ be a point in $\mathcal{C}$ and τ be the grid width, then $id_{\mathcal{C}} = (\lfloor \frac{x_p}{\tau} \rfloor, \lfloor \frac{y_p}{\tau} \rfloor)$, (see Fig. 1). Note that each grid cell has a unique ID associated with it but multiple points can be associated with the same ID (if they lie within the corresponding grid cell). Let $id_{\mathcal{C}_1}$ and $id_{\mathcal{C}_2}$ be any two arbitrary grid cells with base points (x_1, y_1) and (x_2, y_2) respectively. We define the greater than operator for an ID as follows: $(id_{\mathcal{C}_1} > id_{\mathcal{C}_2}) \iff ((x_1 > x_2) \wedge (y_1 > y_2)) \vee ((x_1 = x_2) \wedge (y_1 > y_2)) \vee ((x_1 > x_2) \wedge (y_1 = y_2))$. Note that our definition of $id_{\mathcal{C}}$ implies that the grid cells are indexed from bottom-left to top-right, this operator simply indicates the order of iteration that is followed by our algorithm. We move left to right row-wise. A pre-defined order is essential to our handing-over logic.

C_1 $[0, 2]$	C_2 $[1, 2]$	C_1 $[2, 2]$
C_3 $[0, 1]$	C_4 $[1, 1]$	C_3 $[2, 1]$
C_1 $[0, 0]$	C_2 $[1, 0]$	C_1 $[2, 0]$

Fig. 1. Assignment of unique ID's $id_{\mathcal{C}}$ and color sets for a grid $\mathcal{G}$ with $\tau = 2$

Given a grid cell $\mathcal{C}'$, the following lemma provides a bound on the cardinality of the k-colorable unit disks covering all points lying within the grid cell $\mathcal{C}'$. Let $D_{\mathcal{C}'} \subseteq D$ be the set of k-colorable unit disks covering all points of P lying within the grid cell $\mathcal{C}'$. The proof of the lemma is based on the observation that determining this bound is the same as determining a maximum number of disjoint unit disks that could potentially intersect $\mathcal{C}'$.

Observation 1. *If $\mathcal{C}'$ is a grid cell of size $\tau \times \tau$, then the maximum number of pairwise disjoint unit disks that could potentially intersect $\mathcal{C}'$ is at most $2\tau + 2 + (\frac{\tau}{2})^2$ if τ is even, and is atmost $4 \times \lceil \frac{\tau}{2} \rceil + 4 + (\frac{\tau}{2})^2$ if τ is odd.*

Proof. We will provide an upper bound to the number of pairwise disjoint unit disks that can cover a square $\mathcal{C}'$ of side length τ. Let us prove this by considering the two cases: τ being even and odd.

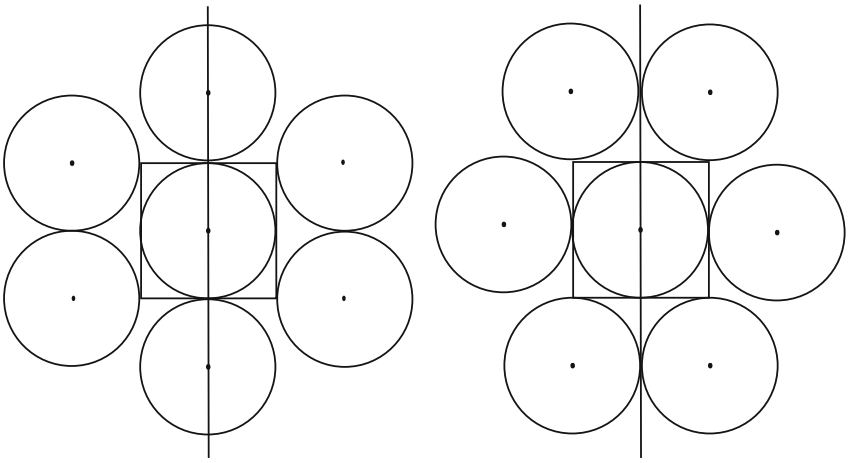
Since we aim at bringing an upper bound to the number of disks that have a common intersection point with $\mathcal{C}'$, we divide the region of $\mathcal{C}'$ into two parts; the inner part of the square and the outer edge on which these common intersection points can lie. To maximize the number of disks, it is intuitive to keep them as far as possible to increase the spacing between disks and thereby trying to increase the number of disks.

$\mathcal{C}'$ with Even Side Length: When τ is a multiple of 2, it is quite intutive that a symmetric pattern is likely to give the best results. So we attempt two types of symmetric pattern.

Case 1: Considering the square $\mathcal{C}'$ to be symmetric along the vertical axis, we arrange the disks in two possible cases: either a disk is arranged with edge of $\mathcal{C}'$ as tangent such that the center of $\mathcal{C}'$ lies vertically above/below the disk (see Fig. 2a), or the vertical partition is tangent to some of the disks (see Fig. 2b). In the first case the maximum number of disks along horizontal part of outer edge would be $2 \times \frac{\tau}{2}$, since the diameter of disk is 2. In the second case it can be shown that the maximum number of disks along the horizontal part would be $2 \times (\frac{\tau}{2} + 1)$.

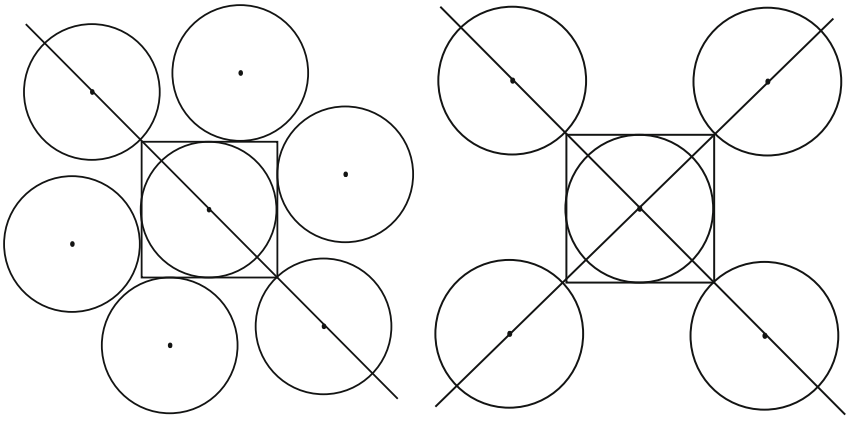
However, as we see for the case $\tau = 2$ (shown in Fig. 2a and Fig. 2b), both the cases break down to the same case since if for a given pair of parallel edges of $\mathcal{C}'$ if one of the above case is true, then for the other pair the other case stands true. By induction we can prove that this stands true for all the even values of $\tau > 2$. Through this we get an upper limit in the number of disks along the edges. For the inner area of $\mathcal{C}'$ the maximum number of disjoint disks are $(\frac{\tau}{2})^2$. So the maximum number of disjoint disks are $(\tau/2)^2 + 2 \times \tau/2 + 2 \times (\tau/2 + 1)$, which is $2\tau + 2 + (\tau/2)^2$.

Case 2: Considering $\mathcal{C}'$ to be diagonally symmetric, here we consider one of its diagonals as the symmetry line and place the disk such that their center is on the diagonal and intersects the square at its corner. As the disk covers a part of the edge of the square, say δ where $\delta < 1$ (i.e., radius of disk) and $\delta > 0$, apart from the case where the disks are arranged at the diagonals again (see Fig. 3a), the maximum number of disks on an edge is still equivalent to $\frac{\tau}{2}$. The number of



(a) Center of disk along the vertical axis (b) Vertical line dividing the square as tangent

Fig. 2. Possible symmetric arrangement along vertical axis for even values of τ.

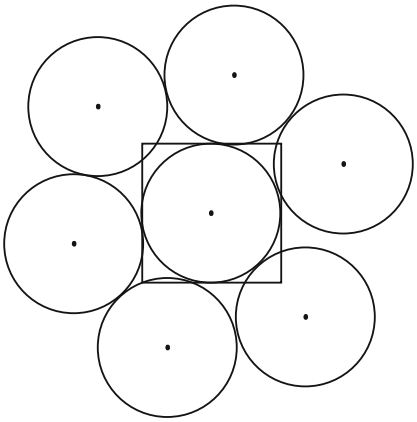


(a) Disks arranged along one of the diagonal (b) Disks arranged are along both the diagonals

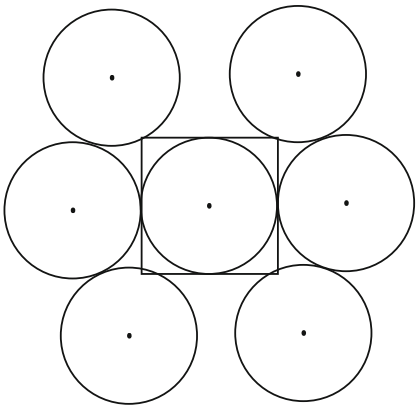
Fig. 3. Possible symmetric arrangement along diagonal for even values of τ.

disks within the interiors of $\mathcal{C}'$ as stated above in the previous case will remain same. Therefore, the total number of maximum possible pairwise disjoint disks are $2 + 4\times(\frac{\tau}{2}) + (\frac{\tau}{2})^2$, which is again equivalent to $2\tau + 2 + (\frac{\tau}{2})^2$ (see Fig. 3a, Fig. 3b, Fig. 4a, Fig. 4b, and Fig. 4c for illustration of the case $\tau = 2$).

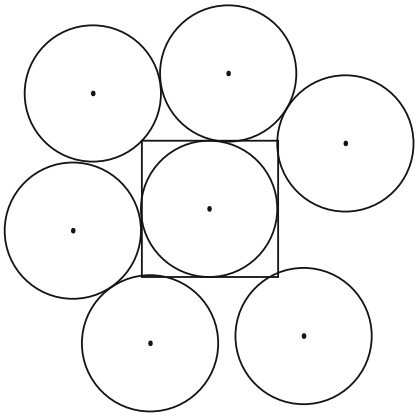
$\mathcal{C}'$ with Odd Side Length: When τ is not a multiple of 2, again it is quite intuitive that a symmetric pattern is likely to give the best results. So we attempt two types of symmetric pattern.



(a) Complete coverage as much closely as possible



(b) Symmetric coverage

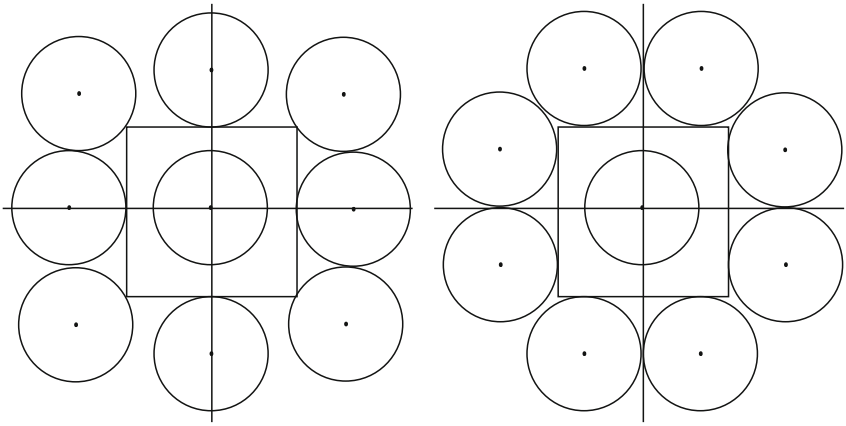


(c) Asymmetric coverage

Fig. 4. Various possibilities by trial and error for even values of τ.

Case 1: Like the case for even values of τ we consider the symmetric distribution along the horizontal and vertical axes. Again we have two possibilities either the center of disk lying along the axes or symmetric about the axes for both the pairs of edges. It can be shown that for both the pair of edges we would have only one amongst the two configurations at a time for getting the minimum number of disks. In the first case, $\lceil \frac{\tau}{2} \rceil$ disks can completely be accommodated on one edge and one disk as a common disk between two adjacent edges. So, there will be $4 \times \lceil \frac{\tau}{2} \rceil + 4$ disks in the exterior part for this case. The number of disks in the interior part as in the even case would be $(\frac{\tau}{2})^2$. So, the total becomes $4 \times \lceil \frac{\tau}{2} \rceil + 4 + (\frac{\tau}{2})^2$ (see Fig. 5a and Fig. 5b).

Case 2: When diagonally symmetric, the case is quite similar to the previous case with 4 disks at the corners and $\lceil \frac{\tau}{2} \rceil$ among the four edges of $\mathcal{C}'$. The total again is calculating to $4 \times \lceil \frac{\tau}{2} \rceil + 4 + (\frac{\tau}{2})^2$. □



(a) Center of disks along the axes (b) Axes as tangent to the disks

Fig. 5. Possible symmetric arrangement for odd values of τ.

Lemma 1. *If $\mathcal{C}'$ is a grid cell of size 2×2 and $D_{\mathcal{C}'} \subseteq D$ is a k-colorable solution for $P \cap \mathcal{C}'$, then $|D_{\mathcal{C}'}| \leq 7k$.*

Proof. From Observation 1 the cardinality of any set $S_{\mathcal{C}'}$ of pairwise disjoint unit disks intersecting with a grid cell $\mathcal{C}'$ is at most 7 as $\tau = 2$. Therefore, $|S_{C'}| \leq 7$. Now consider another $k-1$ sets $S^1_{\mathcal{C}'}$, $S^2_{\mathcal{C}'}$, ..., $S^{k-1}_{\mathcal{C}'}$, each of which can either be a replica of the same collection of disks in $S_{\mathcal{C}'}$ or a rotation or transformation of $S_{\mathcal{C}'}$ such that disks within each set remain pairwise disjoint and intersect $\mathcal{C}'$. Hence, any k-colorable solution $D_{\mathcal{C}'} \subseteq S_{\mathcal{C}'} \cup S^1_{\mathcal{C}'} \cup S^2_{\mathcal{C}'} \cup \ldots \cup S^{k-1}_{\mathcal{C}'}$. Thus, the lemma follows. □

The outline of our algorithm (Algorithm 1) for computing a cover $D' \subseteq D$ of the points P is as follows. We first partition the rectangular region containing the objects in D and P into individual grid cells of size $\tau \times \tau$. By utilizing the bound obtained in Observation 1 (for e.g., for $\tau = 2$ the actual bound is in Lemma 1) we compute a k-colorable cover of the points lying in each grid cell, in an exhaustive manner. To ensure that there is no conflict in the overall aggregate solution, we use a handing-over logic. Only disks of any particular grid cell cover centered within the same grid cell are colored with the associated color set. If a disk is required to be a part of this grid cell cover, but is centered in another grid cell, it is handed-over to that grid cell. Based on the grid width τ and the diameter of the disk, we then define a coloring scheme χ that assigns a color to each disk in the union D' of all the individual grid cell covers computed.

Finally, we return the pair (D', χ). Since the diameter of the disks is fixed to be two units, the approximation factor of the algorithm is implied by the choice of the value τ. If the value of τ is 2, then a unit disk can participate in the k-colorable covers of points lying in four adjacent grid cells. Hence, we prove that Algorithm 1 is a 4-approximate algorithm (see Theorem 1). Later, we show that by varying the grid width τ, which results in a unit disk participating in more than four individual grid cell covers, we can obtain a family of algorithms with approximation factors corresponding to the choice of the value of τ (see Subsect. 2.2).

We now define any coloring function that assigns colors to disks to be conflict-free if for any pair of non-disjoint disks (i.e., overlapping disks) the colors assigned to them are different.

Lemma 2. *The coloring χ defined by Algorithm 1 is conflict-free.*

Proof. For the sake of contradiction, let us assume that there are two disks $d, d' \in D'$ such that $d \cap d' \neq \emptyset$, and $\chi(d) = \chi(d')$, where D' along with χ is the output of Algorithm 1. Since $d \cap d' \neq \emptyset$, the distance between the centers of d and d' is at most 2. Let the centers of d and d' be lying in the grid cells $\mathcal{C}$ and $\mathcal{C}'$, respectively. Observe that $\mathcal{C}$ and $\mathcal{C}'$ are either linearly or diagonally adjacent. If d and d' are chosen to cover points lying only in the respective grid cells, then d and d' are assigned colors from different color sets because the row and column numbers mod 2 in their ID's are not the same for both (see for-loop at Line 17) (contradicting that $\chi(d) = \chi(d')$). Therefore, the only possibility for color-conflict to arise between d and d' is that when both d and d' are centered in the same grid cell $\mathcal{C}$, where d covers a point lying in the cell above $\mathcal{C}$ and d' covers a point lying in the cell below $\mathcal{C}$ and each disk is initially chosen by the respective grid cell by means of the algorithm (Note that a similar case can be studied for horizontally and diagonally opposite grid cells). As per our color scheme (Line 16–28), these grid cells are the nearest to have the same color set (say C_1) associated with them (see Fig. 1). Step 11 in the algorithm solves the conflict that arises in this case as follows. By means of the grid cell ID condition, disk d' is handed over to grid cell $\mathcal{C}$ as its ID is greater and will receive the color set associated with that cell (C_3 in this case, see Fig. 1). Disk d however will not be handed over, but retains a color from the color set C_1 (contradicting that $\chi(d) = \chi(d')$). Thus, the lemma follows. □

Theorem 1. *Algorithm 1 is a 4-approximation algorithm that runs in $O(m^{7k} n \log k)$ time for the k-CDUDC problem.*

Proof. In line 2 of Algorithm 1, the rectangular region $\mathcal{R}$ is partitioned into $O(n)$ grid cells. For each grid cell $\mathcal{C}$, let $n_{\mathcal{C}}$ denote the number of points of P lying in $\mathcal{C}$, i.e., $n_{\mathcal{C}} = |P \cap \mathcal{C}|$. For each grid cell $\mathcal{C}$ we then enumerate all $O(m^7)$ subsets of D such that each such subset contains at most 7 pairwise disjoint disks intersecting with the cell $\mathcal{C}$. In each iteration of the for-loop at line 3, among $\binom{cm^7}{k}$ subsets of at most $7k$ disks for some constant c, we compute a set $D_{\mathcal{C}}$ that covers all $n_{\mathcal{C}}$

Algorithm 1. K_Colorable_Cover(P, D, k)

Input: A set P of n points, a set D of m unit disks in the plane, and an integer $k(> 0)$ such that $P \subset \cup_{d \in D} d$ and D can provide a k-colorable cover of the points in P.

Output: A k-colorable set $D' \subseteq D$ that covers all the points in $\mathcal{P}$ and a color mapping $\chi : D' \rightarrow \kappa$, where κ denotes the color set of distinct colors, and $|\kappa| \leq 4k$

Let the points in P and disks in D be lying entirely within the first quadrant of the coordinate system, and $\mathcal{R}$ be an axis-aligned rectangular region containing P and D, whose left and bottom boundary lines coincide with the y- and x-axes of the coordinate system, respectively.

Define a grid $\mathcal{G}$ that partitions $\mathcal{R}$ such that each grid cell is of size 2×2 and for each point $p = (x_p, y_p)$ lying in such a cell $\mathcal{C}$, let the unique id associated with $\mathcal{C}$ be $id_{\mathcal{C}} = [\lfloor \frac{x_p}{2} \rfloor, \lfloor \frac{y_p}{2} \rfloor]$. For each such cell $\mathcal{C}$, we also define a handover set $H_{\mathcal{C}} \leftarrow \emptyset$.

/* the grid cells in the following loop are considered in row-wise order from bottom-left to top-right, as defined in Subsection 2.1 */

for each grid cell $\mathcal{C}$ if $P \cap \mathcal{C} \neq \emptyset$ **do**

  if $H_{\mathcal{C}} = \emptyset$ **then**

    Let $D'' = \{d \in D \mid d \cap \mathcal{C} \neq \emptyset\}$

    Generate all subsets $D_1, D_2, \ldots, D_{O(m^7)} \subseteq D''$, each containing at most 7 pairwise disjoint disks, and among these, choose k subsets $S^1_{\mathcal{C}}, S^2_{\mathcal{C}}, \ldots, S^k_{\mathcal{C}}$, whose union covers all the points in $P \cap \mathcal{C}$.

  else

    Let $D'' = \{d \in D \mid d \cap \mathcal{C} \neq \emptyset, d \notin H_{\mathcal{C}}\}$.

    Generate all subsets $D_1, D_2, \ldots, D_{O(m^7)} \subseteq D''$, each containing at most 7 pairwise disjoint disks, and among these, choose k subsets $S^1_{\mathcal{C}}, S^2_{\mathcal{C}}, \ldots, S^k_{\mathcal{C}}$, whose union covers all the points in $P \cap \mathcal{C}$ but also contains all disks $d \in H_{\mathcal{C}}$.

  /* $|S^1_{\mathcal{C}} \cup S^2_{\mathcal{C}} \cup \ldots S^k_{\mathcal{C}} \cup H_{\mathcal{C}}| \leq 7k$ due to Lemma 1 */

  end if

  If any disk d in any subset $S^i_{\mathcal{C}}$ (for $i = 1, \ldots, k$) is centered in another grid cell $\mathcal{C}'$ whose ID $id_{\mathcal{C}'} > id_{\mathcal{C}}$, we remove that disk from $S^i_{\mathcal{C}}$ and add it to the handover set of that cell $H_{\mathcal{C}'}$.

  $D_{\mathcal{C}} \leftarrow S^1_{\mathcal{C}} \cup S^2_{\mathcal{C}} \cup \ldots \cup S^k_{\mathcal{C}}$

  For every point $p \in P$ that is covered by a disk $d \in D_{\mathcal{C}}$ we remove it from P.

end for

$D' \leftarrow \bigcup_{\mathcal{C}, P \cap \mathcal{C} \neq \emptyset} D_{\mathcal{C}}$

Let C_1, C_2, C_3, C_4 be four disjoint color sets, each containing k distinct colours.

for every grid cell $\mathcal{C}$ with $id_c = [i, j]$ **do**

  if $((i \bmod 2 = 0) \wedge (j \bmod 2 = 0))$ **then**

    Assign C_1 to $\mathcal{C}$.

  else if $((i \bmod 2 = 0) \wedge (j \bmod 2 \neq 0))$ **then**

    Assign C_2 to $\mathcal{C}$.

  else if $((i \bmod 2 \neq 0) \wedge (j \bmod 2 = 0))$ **then**

    Assign C_3 to $\mathcal{C}$.

  else if $((i \bmod 2 \neq 0) \wedge (j \bmod 2 \neq 0))$ **then**

    Assign C_4 to $\mathcal{C}$.

  end if

end for

For every grid cell $\mathcal{C}$ and its assigned color set C_i we allot one color each to the subsets $S^1_C, S^2_C, \ldots S^k_C$. Every disk centered within that subset will now be colored with the corresponding color from the color set.

For any disk $d \in D'$ let $\chi(d)$ represents the color assigned to the disk d in the above coloring assignment process.

return (D', χ)

points lying in $\mathcal{C}$. In order to do this, we first compute the voronoi diagram $VOD_{\mathcal{C}}$ on the center points of these $7k$ disks. We then do point location queries for each of these $n_{\mathcal{C}}$ points on $VOD_{\mathcal{C}}$ to determine the closest center point. We will then test whether the corresponding disk centered at the closest center point covers this point. This step will take $O(k \log k + n_{\mathcal{C}} \log k)$ time. Therefore, we invest $O(m^{7k}(k \log k + n_{\mathcal{C}} \log k))$ time to compute k-colorable cover of the points lying in the cell $\mathcal{C}$. The total time the for-loop takes over all the nonempty grid cells is $O(m^{7k} k \log k) + \sum_{\mathcal{C}} O(m^{7k} n_{\mathcal{C}} \log k) = O(m^{7k}(n+k) \log k)$. The remaining steps of the algorithm (including the preprocessing in steps 1 and 2) will take no more than $O(mn)$ time. Under the assumption that n is much larger than the number k of colors, the time complexity of the algorithm in total is $O(m^{7k} n \log k)$. In order to assign the colors at a later step of the algorithm, with each grid cell $\mathcal{C}$ we also associate and store the corresponding cover $D_{\mathcal{C}}$ computed in the for-loop at line 3. Thus, the additional space the algorithm requires is $O(nk)$.

The handover behaviour at line 11 of the algorithm and Lemma 2 ensure that any color-conflict is resolved in an elegant way. Since we use four disjoint sets of k distinct colors and for each grid cell we compute k-colorable unit disk cover (from Lemma 1), the approximation factor of the algorithm is 4. Thus, the theorem follows. □

The following corollary says that Theorem 1 yields a faster algorithm for 3-*CDUDC* than that of Biedl et al. [3], but at the cost of increase in approximation factor.

Corollary 1. *There exists a 4-approximate algorithm that runs in $O(m^{21}n)$ time for the 3-CDUDC problem.*

Proof. Follows from Theorem 1 by applying $k = 3$. □

2.2 Generalization

In this subsection, we generalize the results from the preceding subsection to a general case observations for possible values of the width τ of each cell in the grid partitioning approach. We begin by attempting to generalize the potential coloring schemes for each grid cell $\tau \times \tau$ similar to the observations presented in Lemma 1. We first define a parameter ρ that represents the factor indicating the number of additional color sets needed to satisfy a union of independent solution sets. As a result, we obtain a family of approximation algorithms for the k-*CDUDC* problem, depending upon a different possible values for ρ and τ.

Lemma 3. *The union of all independently optimal k-colorable solution sets for points lying in each grid cell $\mathcal{C}$ of size $\tau \times \tau$ is ρk colourable, where*

$$\rho = \begin{cases} 4 & \text{if } \tau \geq 2 \\ 6 & \text{if } \frac{8}{5} \leq \tau < 2 \\ 7 & \text{if } \sqrt{2} \leq \tau < \frac{8}{5} \\ 9 & \text{if } 1 \leq \tau < \sqrt{2} \end{cases}$$

Proof. We prove this by verifying the number of grid cells a unit disk can maximally intersect while considering each case. This determines the number of disjoint color sets, each consisting of at most k distinct colors. Clearly, this number is the same as ρ. Note that we are interested in finding the upper bound of such intersections.

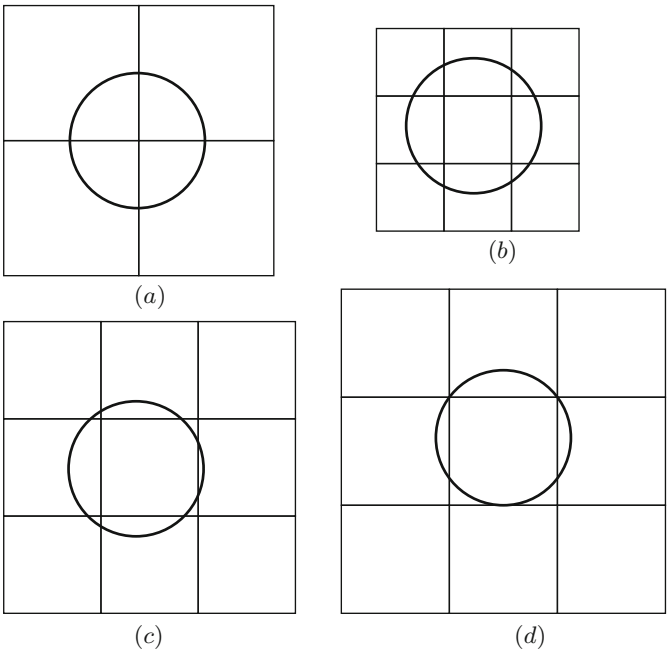


Fig. 6. Proof of Lemma 3

$\tau \geq 2$: Since the diameter of each disk is 2 and the width of grid cell is also ($\geq$)2, certainly, no disk can span more than 2 linearly adjacent disks (see Fig. 6(a)). Thus, maximal intersection count is achieved by placing the disk in any of the grid intersection corners. Here the disk will certainly intersect 4 grid cells regardless of the width.

$\frac{8}{5} \leq \tau < 2$: If the width of the grid cell is less than 2, surely, a unit disk can span 3 linearly adjacent grid cells. However, if the width is $\frac{8}{5}$, the disk cannot span more than 2 diagonally adjacent grid cells (see Fig. 6(d)). Hence, if the width is greater than or equal to $\frac{8}{5}$, then no matter where the disk is centered (in Fig. 6(d)), it can not intersect more than 6 grid cells simultaneously.

$\sqrt{2} \leq \tau < \frac{8}{5}$: If the width of the grid cell is $\sqrt{2}$, surely, a unit disk can span 3 linearly adjacent grid cells and the middle cell from the next adjacent column or row of the linearly adjacent grid cells (see Fig. 6(c)). Thus, we use the position shown in Fig. 6(c) to indicate maximal count possible.

$1 \leq \tau < \sqrt{2}$: For width ≥ 1, a unit disk could potentially intersect 3 linearly and diagonally adjacent grid cells (see Fig. 6(*b*)).

We do not consider the grid partitioning with grid cells of width $\tau < 1$ as it is inconsequential to our study and provides no substantial results (as the colorability increases substantially with no real improvement to the number of intersecting disks). □

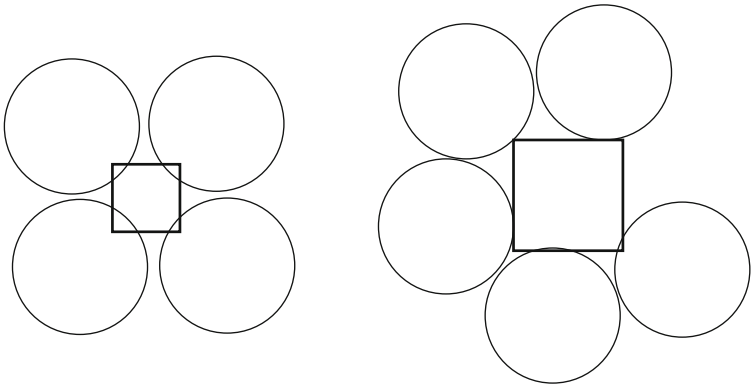
In the same spirit, we provide a generalization of the observation presented in Lemma 3 for the maximum number of pairwise disjoint unit disks that can intersect a square of size $\tau \times \tau$. We define a parameter α that is the count of mutually non-intersecting unit disks that can intersect a grid cell of width τ.

Lemma 4. *The bound on the cardinality of a set $S^i_{\mathcal{C}}$ of pairwise disjoint unit disks that can intersect a grid cell $\mathcal{C}$ of size $\tau \times \tau$ is α, for an integer $1 \leq i \leq k$, where*

$$\alpha = \begin{cases} 4 & \text{if } \tau = 1 \\ 5 & \text{if } \frac{8}{5} \geq \tau \geq \sqrt{2} \\ 7 & \text{if } \tau = 2 \\ 10 & \text{if } \tau = 3 \\ 14 & \text{if } \tau = 4 \\ 17 & \text{if } \tau = 5 \end{cases}$$

Proof. Consider a grid cell $\mathcal{C}$ of size 1×1. Imagine placing four unit disks, each centered farthest apart from one another, but outside the cell $\mathcal{C}$, and touching one of the four corners of the cell $\mathcal{C}$. Since the cell size is 1×1, we can not place any more disk that intersects the cell, but at the same time disjoint from each of these four disks. Hence, the bound $\alpha = 4$ for the case of width $\tau = 1$. Proof for each of the other cases can be done similar to the proofs provided for the cases $\tau = 1$ given above, and $\tau = 2$ given in Lemma 1 (see, for example, the cases $\tau = 1$ and $\tau = 8/5$ being illustrated in Fig. 7). We do not consider grid cells of width $\tau > 5$ as it provides no improvement to the running time while the approximation factor stays the same after $\tau = 2$ (as seen in Lemma 3). On the other hand, $\tau < 1$ is also not considered because the approximation factor ρ becomes arbitrarily very high for $\tau < 1$. One can observe here that the upper bound for $|S^i_{\mathcal{C}}|$ obtained in Observation 1 is tight in case of τ being even, whereas in case of odd, it is a loosely bound. □

Theorem 2. *There exists a ρ-approximation algorithm to solve the k-CDUDC problem, that has a running time of $O(m^{\alpha k} n \log k)$ for a given grid width $1 \leq \tau \leq 5$, where*



(a) $\tau = 1$ (b) $\tau = \frac{8}{5}$

Fig. 7. Proof of Lemma 4

$$\rho = \begin{cases} 4 & \text{if } \tau \geq 2 \\ 6 & \text{if } \frac{8}{5} \leq \tau < 2 \\ 7 & \text{if } \sqrt{2} \leq \tau < \frac{8}{5} \\ 9 & \text{if } 1 \leq \tau < \sqrt{2} \end{cases}, \quad \alpha = \begin{cases} 4 & \text{if } \tau = 1 \\ 5 & \text{if } \frac{8}{5} \geq \tau \geq \sqrt{2} \\ 7 & \text{if } \tau = 2 \\ 10 & \text{if } \tau = 3 \\ 14 & \text{if } \tau = 4 \\ 17 & \text{if } \tau = 5 \end{cases}$$

Proof. The input of Algorithm 1, in addition to a set P of n points, a set D of m unit disks, and an integer $k(> 0)$, also consists of a grid partitioning parameter τ. From Lemmata 3 and 4, it is clear that any reasonable value for the parameter τ will imply the values of ρ and α. Hence, we have a ρ-approximation algorithm in $O(m^{\alpha k} n \log k)$ time. □

3 Line Segment and Rectangular Region Cover

In the same spirit as the *k-CDUDC* problem is considered due to its practical application in frequency/channel assignment in wireless networks, we also define two problems, that generalize the locations of potential wireless clients from discrete set of points to line segments and from discrete set of points to a continuous rectangular region, namely, the *k-Colorable Line Segment Disk Cover (k-CLSDC)* and *k-Colorable Rectangular Region Cover (k-CRRC)* problems, respectively.

We begin our approach using a fundamental combinatorial result involving unit disks that helps us to transform the above problems into our original *k-CDUDC* problem. Given a set D of m unit disks in the plane, a sector is the smallest region bordered by the boundary lines of disks and is covered by the same set of disks in D. Thus, the arrangement of all disks of D subdivides the

plane into many sectors. It is not hard to show that the worst-case complexity of the arrangement of any set of m unit disks is quadratic, as stated below.

Observation 2. (Funke et al. [9]). *The number of sectors created by intersection of m unit disks in D is $O(m^2)$.*

To develop approximation algorithms for the k-*CLSDC* problem, we transform every instance of k-*CLSDC* problem into an instance of k-*CDUDC* problem as follows. In an instance of k-*CLSDC* problem, we have a set D of m unit disks covering a finite union of n line segments of arbitrary length with arbitrary orientation, and an integer k, The objective, here, is to compute a k-colorable cover of all the line segments. We split each of these line segments into slices such that each such slice lies within some sector. Now, for each subset of slices lying within a single sector, we add one point into the same sector and remove all the slices. This collection of points is referred to as P'. Hence, from Observation 2 we have that $|P'| = O(m^2)$. This can be taken as an instance of the k-*CDUDC* problem, where P' is taken as the input set of points P.

Similarly, we can do a similar transformation for the k-*CRRC* problem. Here, we have a set D of m unit disks covering a continuous rectangular region $\mathcal{R}$. Our objective is to compute a k-colorable set of units disks such that $\mathcal{R}$ is covered by the union of these disks. As above, we split $\mathcal{R}$ into $O(m^2)$ sectors, as induced by the union of disks in D, and add one point into each sector.

For the k-*CLSDC* problem, the construction of the set P' of points can be done as follows. We first preprocess the given set D of m unit disks into any reasonable data structure, e.g., a doubly connected edge list (*DCEL*), in $O(m^2)$ time [14]. We can build the Voronoi diagram VOR_D on the center points of disks in D in $O(m \log m)$ time. We store cross pointers between Voronoi cells (or disks) in VOR_D and the faces (or sectors) of *DCEL* that are contributed by the corresponding disks. For each of the given n line segments, we do point location query on VOR_D for the left endpoint of the line segment to determine the disk in which it lies. We then follow the cross pointer to access the sector that contains it. Subsequently, we traverse the adjacent sectors of this sector in *DCEL*. As we do, we add points into those sectors (also, into P') that covers a portion of the line segment and mark the corresponding faces (or sectors) as processed in *DCEL*. This step will take $O(n \log m + m^2)$ time. In the case of the k-*CRRC* problem, the construction of P' takes $O(m^2)$ time as we have to test whether each sector is intersected by $\mathcal{R}$.

Therefore, we have the following results for the k-*CLSDC* and k-*CRRC* problems.

Theorem 3. *We have a ρ-approximation algorithm to solve the k-CLSDC problem, that has a runing time of $O(n \log m + m^{\alpha k+2} \log k + m^{\alpha k} k \log k)$ for a given grid width $1 \leq \tau \leq 5$, where ρ and α are defined as in Theorem 2.*

Proof. Follows from Theorem 2 by applying $|P| = |P'| = O(m^2)$, where $O(n \log m)$ is due to the preprocessing of the input before we run the algorithm of Theorem 2. □

Theorem 4. *We have a ρ-approximation algorithm to solve the k-CRRC problem, that has a runing time of $O(m^{\alpha k+2} \log k + m^{\alpha k} k \log k)$ for a given grid width $1 \leq \tau \leq 5$, where ρ and α are defined as in Theorem 2.*

Proof. Follows from Theorem 2 by applying $|P| = |P'| = O(m^2)$. □

4 Conclusion

In this paper, we have proposed constant-factor approximation algorithms for computing k-colorable unit disk covering of points, line segments, and a rectangular region. In the future work, we wish to improve the running time of the local algorithm for a grid cell instead of brute-forcing for optimal cover of points lying in the grid cell. This will improve the overall running time of the approximation algorithms.

References

1. Basappa, M.: Line segment disk cover. In: Panda, B.S., Goswami, P.P. (eds.) CALDAM 2018. LNCS, vol. 10743, pp. 81–92. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-74180-2_7
2. Basappa, M., Acharyya, R., Das, G.K.: Unit disk cover problem in 2D. J. Discrete Algorithms **33**, 193–201 (2015)
3. Biedl, T., Biniaz, A., Lubiw, A.: Minimum ply covering of points with disks and squares. Comput. Geom. **94**, 101712 (2021)
4. Brass, A., Hurtado, F., Lafreniere, B.J., Lubiw, A.: A lower bound on the area of a 3-coloured disk packing. Int. J. Comput. Geom. Appl. **20**(3), 341–360 (2010)
5. de Berg, M., Markovic, A.: Dynamic conflict-free colorings in the plane. Comput. Geom. **78**, 61–73 (2019)
6. Clark, B.N., Colbourn, C.J., Johnson, D.S.: Unit disk graphs. Discrete Math. **86**(1–3), 165–177 (1990)
7. Cheilaris, P., Gargano, L., Rescigno, A., Smorodinsky, S.: Strong conflict-free coloring for intervals. Algorithmica **70**(4), 732–749 (2014)
8. Even, G., Lotker, Z., Ron, D., Smorodinsky, S.: Conflict-free colorings of simple geometric regions with applications to frequency assignment in cellular networks. SIAM J. Comp. **33**(1), 94–136 (2003)
9. Funke, S., Kesselman, A., Kuhn, F., Lotker, Z., Segal, M.: Improved approximation algorithms for connected sensor cover. Wirel. Netw. **13**(2), 153–164 (2007)
10. Fraser, R., López-Ortiz, A.: The within-strip discrete unit disk cover problem. Theor. Comput. Sci. **674**, 99–115 (2017)
11. Horev, E., Krakovski, R., Smorodinsky, S.: Conflict-free coloring made stronger. In: Kaplan, H. (ed.) SWAT 2010. LNCS, vol. 6139, pp. 105–117. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13731-0_11
12. Mustafa, N.H., Ray, S.: Improved results on geometric hitting set problems. Discrete Comput. Geom. **44**(4), 883–895 (2010)
13. Smorodinsky, S.: Combinatorial problems in computational geometry. Ph.D. thesis. Tel-Aviv University (2003)
14. de Berg, M., Cheong, O., van Kreveld, M., Overmars, M.: Computational Geometry: Algorithms and Applications, 3rd edn. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-77974-2