On the Discrete Unit Disk Cover Problem

Gautam K. Das[1], Robert Fraser[2], Alejandro Lopez-Ortiz[2], and Bradford G. Nickerson[1]

[1] Faculty of CS, University of New Brunswick, Fredericton, NB - E3B 5A3, Canada
[2] DRC School of CS, University of Waterloo, Waterloo, Ontario - N2L 3G1, Canada

Abstract. Given a set $\mathcal{P}$ of n points and a set $\mathcal{D}$ of m unit disks on a 2-dimensional plane, the *discrete unit disk cover (DUDC)* problem is (i) to check whether each point in $\mathcal{P}$ is covered by at least one disk in $\mathcal{D}$ or not and (ii) if so, then find a minimum cardinality subset $\mathcal{D}^* \subseteq \mathcal{D}$ such that unit disks in $\mathcal{D}^*$ cover all the points in $\mathcal{P}$. The discrete unit disk cover problem is a geometric version of the general set cover problem which is NP-hard [14]. The general set cover problem is not approximable within $c \log |\mathcal{P}|$, for some constant c, but the DUDC problem was shown to admit a constant factor approximation. In this paper, we provide an algorithm with constant approximation factor 18. The running time of the proposed algorithm is $O(n \log n + m \log m + mn)$. The previous best known tractable solution for the same problem was a 22-factor approximation algorithm with running time $O(m^2 n^4)$.

1 Introduction

Research on geometric set cover problems is often motivated by applications in wireless networking or facility location problems. Our interest in the problem arose from data management problems upon terrains, in bathymetric applications in particular. Suppose that we have a survey of a terrain with data represented as points in the xy-plane, and the height data is stored as an attribute of each point. Call this point set $\mathcal{P}$. Given a new survey of the same area, we obtain a new point set $\mathcal{Q}$. We wish to update our data set by treating the new data set $\mathcal{Q}$ as the standard, but we wish to maintain some of the old data for completeness. Our restrictions are that each new point in $\mathcal{Q}$ must have at least one old point from $\mathcal{P}$ within unit distance, and that a minimum number of points from $\mathcal{P}$ are maintained. This problem is an instance of the *discrete unit disk cover* (DUDC) problem.

Other applications of DUDC that are commonly used are selecting locations for wireless servers from a set of candidate locations to cover a set of wireless clients, or positioning emergency service centres (i.e. fire stations) from a set of candidate sites so that all points of interest (houses, etc.) are within a predefined maximum distance of the service centres.

1.1 Our Results

Given a set $\mathcal{P}$ of n points and a set $\mathcal{D}$ of m unit disks in a 2-dimensional plane, we first check the feasibility of the DUDC problem i.e., check whether each point in $\mathcal{P}$ is covered by at least one disk in $\mathcal{D}$ or not. If the answer to the first step is yes, then we propose an 18-approximation algorithm for the DUDC problem. The running time of the proposed algorithm is $O(n \log n + m \log m + mn)$.

1.2 Related Work

The DUDC problem is a geometric version of the general set cover problem which is NP-hard [14]. The general set cover problem is not approximable within $c \log |\mathcal{P}|$, for some constant c [18]. Obviously the general approximation algorithms for set cover apply to DUDC to get an easy $O(\log n)$ approximation (i.e. [8]), but a series of constant factor approximation algorithms and a PTAS have been presented for DUDC, mostly published within the past few years:

- O(1)-approximation, Brönnimann and Goodrich, 1995 [4];
- 108-approximation, Calinescu et al., 2004 [6];
- 72-approximation, Narayanappa amd Voytechovsky, 2006 [17];
- 38-approximation, Carmi et al., 2007 [7];
- 22-approximation, Claude et al., 2010 [5];
- (1+ε)-approximation, Mustafa and Ray, 2010 [16].

Using local search, Mustafa and Ray [16] developed a PTAS for the DUDC problem. Their algorithm runs in $O(m^{2(c/\varepsilon)^2+1}n)$ time, where $c \leq 4\gamma$ [16], and γ can be bounded from above by $2\sqrt{2}$ [10, 15]. The fastest worst-case running time is obtained by setting $\varepsilon = 1$ for a 2-approximation, which runs in $O(m^{2\cdot(8\sqrt{2})^2+1}n) = O(m^{257}n)$ time. Clearly, this algorithm is not practical for $m \geq 2$. Our present work is directed towards finding approximation algorithms for DUDC that are tractable.

Minimum Geometric Disk Cover. In the minimum geometric disk cover problem, the input consists of a set of points in the plane, and the problem is to find a set of unit disks of minimum cardinality whose union covers the points. Unlike DUDC, disk centers are not constrained to be selected from a given discrete set, but rather may be centered at arbitrary points in the plane. Again, this problem is NP-hard [9] and has a PTAS solution [11, 12].

Discrete *k*-Center. Given two sets of points in the plane $\mathcal{P}$ and $\mathcal{Q}$ and an integer k, find a set of k disks centered on points in $\mathcal{P}$ whose union covers $\mathcal{Q}$ such that the radius of the largest disk is minimized. Observe that set Q has a discrete unit disk cover consisting of k disks centered on points in P if and only if Q has a discrete k-center centered on points in P with radius at most one. This problem is NP-hard if k is an input variable [2], but when k is fixed, a solution can be found in $m^{O(\sqrt{k})}$ time [13], or $m^{O(k^{1-1/d})}$ time for points in $\mathbb{R}^d$ [1].

2 Preliminaries

Here, we describe two restricted discrete unit disk cover problems. The first problem is called the *Restricted Line Separable Discrete Unit Disk Cover (RLS-DUDC)* problem and the second one is called the *Line Separable Discrete Unit Disk Cover (LSDUDC)* problem. Both problems are defined on the 2-dimensional plane, and depend on the plane being divided into two half-planes ℓ^+ and ℓ^- defined by a line ℓ. The definition of RLSDUDC and LSDUDC are given below:

RLSDUDC: Given a set of n points $\mathcal{P}$ and a set $\mathcal{D}$ of m unit disks such that the points in $\mathcal{P}$ and the center of disks in $\mathcal{D}$ are separated by the line ℓ, and each point in $\mathcal{P}$ is covered by at least one disk in $\mathcal{D}$, find a minimum size set $\mathcal{D}^* \subseteq \mathcal{D}$ such that each point in $\mathcal{P}$ is covered by at least one disk in $\mathcal{D}^*$.

LSDUDC: Given a set of n points $\mathcal{P}$ in ℓ^+ and a set $\mathcal{D}$ of m unit disks with centers in $\ell^+ \cup \ell^-$ such that each point in $\mathcal{P}$ is covered by at least one disk centered in ℓ^-, find a minimum size set $\mathcal{D}^* \subseteq \mathcal{D}$ such that each point in $\mathcal{P}$ is covered by at least one disk in $\mathcal{D}^*$.

Theorem 1. *[5] The RLSDUDC problem can be solved optimally whereas we solve the LSDUDC problem with a 2-factor approximation. The running time for both solutions is $O(n \log n + mn)$ where $m = |\mathcal{D}|$ and $n = |\mathcal{P}|$.*

3 Testing Feasibility

Historically, discussions of set cover problems typically assume that a feasible solution exists, both to simplify the presentation and because the test is simple. We include a brief discussion here for completeness, but furthermore to demonstrate that the incorporation of a feasibility test does not affect the final running time of our algorithm.

For the feasibility check, given a point set $\mathcal{P}$ and a unit disk set $\mathcal{D}$, the test that all points in $\mathcal{P}$ are covered by at least one disk in $\mathcal{D}$ can be done in $O(m \log m + n \log m)$ time as follows:

Let $\mathcal{D}_{center}$ be the set of centers of the unit disks in $\mathcal{D}$. Draw the Voronoi diagram $VD(\mathcal{D}_{center})$ of the points in $\mathcal{D}_{center}$ [3]. For each point $p \in \mathcal{P}$, find its nearest point $q_p \in \mathcal{D}_{center}$ using the point location algorithm in the planar subdivision $VD(\mathcal{D}_{center})$. If $\delta(p, q_p) \leq 1$ for all $p \in \mathcal{P}$ we can say that all the points in $\mathcal{P}$ are covered by at least one unit disk in $\mathcal{D}$ where $\delta(a, b)$ = the Euclidean distance between points a and b. The time complexity follows from the following facts:

(i) constructing the Voronoi diagram needs $O(m \log m)$ time, and (ii) for each point $p \in \mathcal{P}$, the planar point location algorithm needs $O(\log m)$ time.

Thus, we have the following lemma:

Lemma 1. *The feasibility of the DUDC problem can be determined in $O(m \log m + n \log m)$ time.*

4 Discrete Unit Disk Cover Problem

Given the feasibility test, for the remainder of the discussion we will assume that all points in $\mathcal{P}$ are covered by at least one disk in $\mathcal{D}$. Let R be the axis aligned rectangle such that all points in $\mathcal{P}$ and all centers of the disks in $\mathcal{D}$ are inside R. In the DUDC algorithm, we first divide R by horizontal line segments $\ell_1, \ell_2, \ldots, \ell_{t-1}$ from top to bottom such that $\delta(\ell_i, \ell_{i+1}) = \frac{1}{\sqrt{2}}$ for $i = 0, 1, \ldots, t-1$ where ℓ_0 and ℓ_t are the top and bottom boundary of the rectangle R and $\delta(a, b)$ is the distance between two horizontal line segments a and b. We denote the horizontal strip bounded by the lines ℓ_i and ℓ_{i+1} as $[\ell_i, \ell_{i+1}]$.

The algorithm for the discrete unit disk cover problem is a two step algorithm. In the first step, we propose an algorithm for covering points in $\mathcal{P} \cap [\ell_i, \ell_{i+1}]$ which are covered only by the disks centered inside the strip $[\ell_i, \ell_{i+1}]$ for each $i = 0, 1, \ldots, t-1$. We denote this problem as the *within strip* problem. In the second step, we describe an algorithm for covering the remaining points in $\mathcal{P}$. We denote this problem as the *outside strip* problem.

4.1 Within Strip Problem

In this subsection, we propose a 6-approximation algorithm for the discrete unit disk cover problem with a restricted configuration. The points in $\mathcal{P}$ are inside a horizontal strip $\mathcal{H}$ with height $\frac{1}{\sqrt{2}}$ and the centers of the unit disks in $\mathcal{D}$ are also inside the same strip. The objective is to cover all the points in $\mathcal{P}$ using the minimum number of disks in $\mathcal{D}$. Without loss of generality, assume that the width of $\mathcal{H}$ is $\frac{k}{\sqrt{2}}$ for some positive integer k. Before describing the covering algorithm we partition the rectangle $\mathcal{H}$ into k squares of equal size $\frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}}$ by introducing vertical line segments $L_0, L_1, \ldots, L_k$. Assume L_0 and L_k are the left and right vertical boundary of the rectangle. Let the squares be $\sigma_1, \sigma_2, \ldots, \sigma_k$ in left to right order. Let $\mathcal{D}_i^\sigma (\subseteq \mathcal{D})$ be the set of disks centered in the square σ_i, and $\mathcal{P}_i^\sigma (\subseteq \mathcal{P})$ be the set of points in the square σ_i. The within strip discrete unit disk cover (WSDUDC) algorithm for $\mathcal{P}$ is described in Algorithm 1.

Lemma 2. *Algorithm 1 produces a 6-approximation result in $O(n \log n + mn)$ time.*

Proof. For analysis of the approximation factor consider a disk $d \in \mathcal{D}$. Without loss of generality, assume that the center of d is inside the square σ_i. Since the size of all the squares σ_j $(1 \leq j \leq k)$ is $\frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}}$, one disk centered in σ_j is sufficient to cover all the points in $\mathcal{P}_j^\sigma$.

To analyze the approximation factor consider the following 11 different cases as shown in Fig. 2. All other cases are similar to one of these cases. In Fig. 2, each square σ_i is bounded by dark vertical line segments and a square σ_i is partitioned into σ_i^1 and σ_i^2 of equal size by a dotted vertical line segment.

Algorithm 1 WSDUDC($\mathcal{P}, \mathcal{D}$)

Input: Set $\mathcal{P}$ of points in a horizontal strip $\mathcal{H}$ of height $\frac{1}{\sqrt{2}}$, and set $\mathcal{D}$ of unit disks centered in $\mathcal{H}$.
Output: Set $\mathcal{D}_1^* \subseteq \mathcal{D}$ of disks covering all the points in $\mathcal{P}$.
$\mathcal{D}_{-1}^\sigma \leftarrow \emptyset$, $\mathcal{D}_0^\sigma \leftarrow \emptyset$, $\mathcal{P}_{-1}^\sigma \leftarrow \emptyset$, $\mathcal{P}_0^\sigma \leftarrow \emptyset$, $\mathcal{P}_{k+1}^\sigma \leftarrow \emptyset$, $\mathcal{P}_{k+2}^\sigma \leftarrow \emptyset$ and $\mathcal{D}_1^* \leftarrow \emptyset$.
for ($i = 1, 2, \ldots, k$) **do**
    Compute the sets $\mathcal{D}_i^\sigma$ and $\mathcal{P}_i^\sigma$
end for
Set $i \leftarrow 1$
while ($i \leq k$) **do**
    while (($i \leq k$) and ($\mathcal{D}_i^\sigma = \emptyset$)) **do**
        $i = i + 1$
    end while
    if $\mathcal{D}_{i-2}^\sigma = \emptyset$ **then**
        let $\mathcal{Q}_{left}$ be the subset of $\mathcal{P}_{i-2}^\sigma \cup \mathcal{P}_{i-1}^\sigma$ such that each point in $\mathcal{Q}_{left}$ is covered by at least one disk in $\mathcal{D}_i^\sigma \cup \mathcal{D}_{i+1}^\sigma$ (see Fig. 1(a));
    else
        let $\mathcal{Q}_{left}$ be the subset of $\mathcal{P}_{i-1}^\sigma$ such that each point in $\mathcal{Q}_{left}$ is covered by at least one disk in $\mathcal{D}_i^\sigma \cup \mathcal{D}_{i+1}^\sigma$ (see Fig. 1(b)).
    end if
    Apply the RLSDUDC algorithm based on the line L_{i-1} to cover the points in $\mathcal{Q}_{left}$. Let $\mathcal{D}'$ be the set of disks in this solution.
    Set $\mathcal{D}_1^* = \mathcal{D}_1^* \cup \mathcal{D}'$ and $i = i + 1$.
    while (($i \leq k$) and ($\mathcal{D}_i^\sigma \neq \emptyset$)) **do**
        $i = i + 1$
    end while
    Let $\mathcal{Q}_{right}$ be the subset of $\mathcal{P}_i^\sigma \cup \mathcal{P}_{i+1}^\sigma$ such that each point in $\mathcal{Q}_{right}$ is covered by at least one disk in $\mathcal{D}_{i-2}^\sigma \cup \mathcal{D}_{i-1}^\sigma$ (see Fig. 1(c)). Apply the LSDUDC algorithm based on the line L_{i-1} to cover the points in $\mathcal{Q}_{right}$. Let $\mathcal{D}'$ be the set of disks in this solution. Set $\mathcal{D}_1^* = \mathcal{D}_1^* \cup \mathcal{D}'$ and $i = i + 1$.
end while
Set $j \leftarrow 1$
while ($j \leq k$) **do**
    If $\mathcal{D}_j^\sigma \neq \emptyset, \mathcal{P}_j^\sigma \neq \emptyset$ but $\mathcal{D}_j^\sigma \cap \mathcal{D}_1^* = \emptyset$ then add an arbitrary disk from $\mathcal{D}_j^\sigma$ to $\mathcal{D}_1^*$. Set $j = j + 1$.
end while
return $\mathcal{D}_1^*$

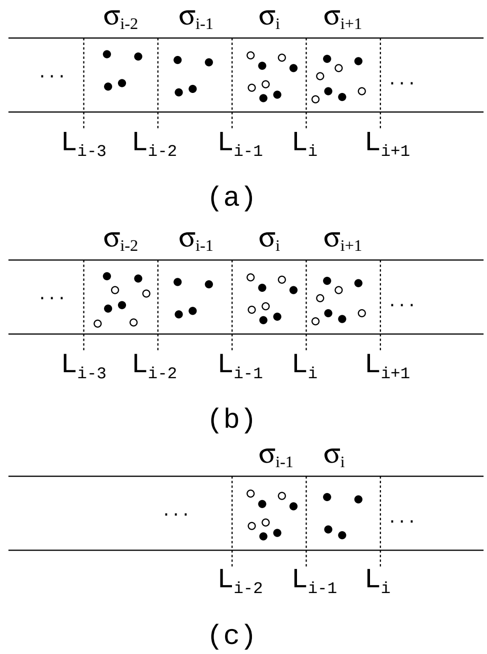

Fig. 1. Demonstration of Algorithm 1. Shaded circles indicate the points in $\mathcal{P}$ and empty circles indicate disk centers.

Case 1: $\mathcal{D}^\sigma_{i-2} = \emptyset, \mathcal{D}^\sigma_{i-1} = \emptyset, \mathcal{D}^\sigma_{i+1} = \emptyset$ (see Fig. 2(a) where $\sigma_i = C$)
Case 2: $\mathcal{D}^\sigma_{i-1} = \emptyset, \mathcal{D}^\sigma_{i+1} = \emptyset$ (see Fig. 2(b) where $\sigma_i = C$)
Case 3: $\mathcal{D}^\sigma_{i-1} = \emptyset, \mathcal{D}^\sigma_{i+1} \neq \emptyset, \mathcal{D}^\sigma_{i+2} = \emptyset$ (see Fig. 2(c) where $\sigma_i = C$)
Case 4: $\mathcal{D}^\sigma_{i-2} = \emptyset, \mathcal{D}^\sigma_{i-1} \neq \emptyset, \mathcal{D}^\sigma_{i+1} = \emptyset$ (see Fig. 2(c) where $\sigma_i = D$)
Case 5: $\mathcal{D}^\sigma_{i-1} = \emptyset, \mathcal{D}^\sigma_{i+1} \neq \emptyset, \mathcal{D}^\sigma_{i+2} \neq \emptyset, \mathcal{D}^\sigma_{i+3} = \emptyset$ (see Fig. 2(d) where $\sigma_i = C$)
Case 6: $\mathcal{D}^\sigma_{i-2} = \emptyset, \mathcal{D}^\sigma_{i-1} \neq \emptyset, \mathcal{D}^\sigma_{i+1} \neq \emptyset, \mathcal{D}^\sigma_{i+2} = \emptyset$ (see Fig. 2(d) where $\sigma_i = D$)
Case 7: $\mathcal{D}^\sigma_{i-3} = \emptyset, \mathcal{D}^\sigma_{i-2} \neq \emptyset, \mathcal{D}^\sigma_{i-1} \neq \emptyset, \mathcal{D}^\sigma_{i+1} = \emptyset$ (see Fig. 2(d) where $\sigma_i = E$)
Case 8: $\mathcal{D}^\sigma_{i-1} = \emptyset, \mathcal{D}^\sigma_{i+1} \neq \emptyset, \mathcal{D}^\sigma_{i+2} \neq \emptyset, \mathcal{D}^\sigma_{i+3} \neq \emptyset, \mathcal{D}^\sigma_{i+4} = \emptyset$ (see Fig. 2(e) where $\sigma_i = C$)
Case 9: $\mathcal{D}^\sigma_{i-2} = \emptyset, \mathcal{D}^\sigma_{i-1} \neq \emptyset, \mathcal{D}^\sigma_{i+1} \neq \emptyset, \mathcal{D}^\sigma_{i+2} \neq \emptyset, \mathcal{D}^\sigma_{i+3} = \emptyset$ (see Fig. 2(e) where $\sigma_i = D$)
Case 10: $\mathcal{D}^\sigma_{i-3} = \emptyset, \mathcal{D}^\sigma_{i-2} \neq \emptyset, \mathcal{D}^\sigma_{i-1} \neq \emptyset, \mathcal{D}^\sigma_{i+1} \neq \emptyset, \mathcal{D}^\sigma_{i+2} = \emptyset$ (see Fig. 2(e) where $\sigma_i = E$)
Case 11: $\mathcal{D}^\sigma_{i-4} = \emptyset, \mathcal{D}^\sigma_{i-3} \neq \emptyset, \mathcal{D}^\sigma_{i-2} \neq \emptyset, \mathcal{D}^\sigma_{i-1} \neq \emptyset, \mathcal{D}^\sigma_{i+1} = \emptyset$ (see Fig. 2(e) where $\sigma_i = F$)

Now we describe the number of appearances of d in the proposed algorithm for the Cases 1, 2, 3 and 4. The other cases are handled similarly.

Case 1 ($\mathcal{D}^\sigma_{i-2} = \emptyset, \mathcal{D}^\sigma_{i-1} = \emptyset, \mathcal{D}^\sigma_{i+1} = \emptyset$ (see Fig. 2(a) where $\sigma_i = C$)): Disk d may appear in the solution of the LSDUDC algorithm with respect to vertical line L_i (line 22 of Algorithm 1). Also d may appear in the solution of the

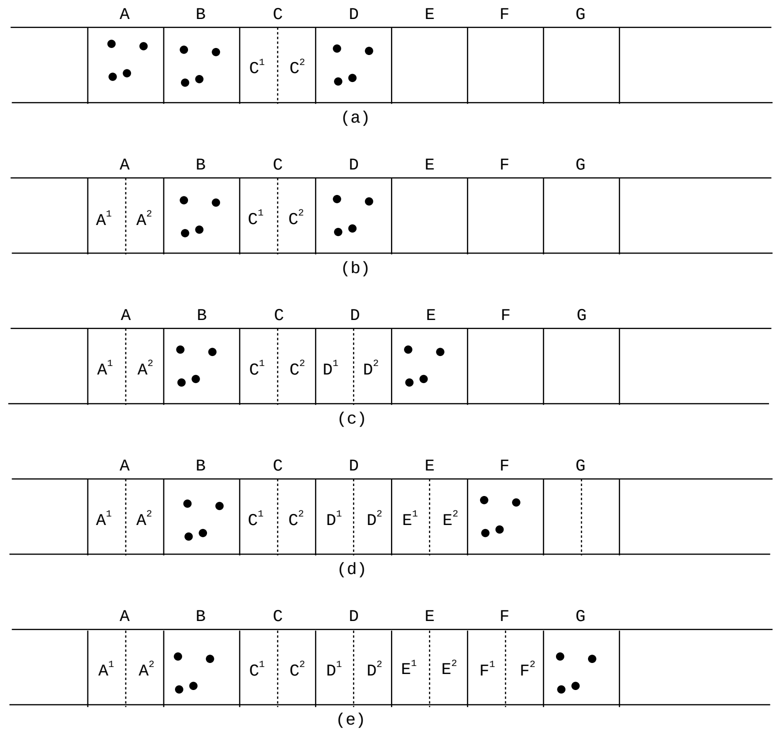

Fig. 2. Different cases in the proof of Lemma 2. Dots inside squares are points in $\mathcal{P}$.

RLSDUDC algorithm with respect to the line L_{i-1} (line 17 of Algorithm 1). If no disk in $\mathcal{D}_i^\sigma$ appears in the solution at the time of running the LSDUDC and RLSDUDC algorithms, then d may appear in the solution (line 26 of Algorithm 1) to cover the points in $\mathcal{P}_i^\sigma$. Therefore, d may appear at most $2 \times 1 + 1 = 3$ times in Algorithm 1 for this case.

Case 2 ($\mathcal{D}_{i-1}^\sigma = \emptyset, \mathcal{D}_{i+1}^\sigma = \emptyset$ (see Fig. 2(b) where $\sigma_i = C$)): Disk d may appear in the solution of the LSDUDC algorithm with respect to vertical lines L_{i-2} and L_i (line 22 of Algorithm 1). Also d may appear in the solution of the RLSDUDC algorithm with respect to the line L_{i-1} (line 17 of Algorithm 1). Again, if no disk in $\mathcal{D}_i^\sigma$ appears in the solution at the time of running the LSDUDC and RLSDUDC algorithms, then d may appear in the solution (line 26 of Algorithm 1) to cover the points in $\mathcal{P}_i^\sigma$ or in $\mathcal{P}_{i-2}^\sigma$. In this case Algorithm 1 may choose at most $2 \times 2 + 1 + 1 = 6$ disks instead of d.

Case 3 ($\mathcal{D}_{i-1}^\sigma = \emptyset, \mathcal{D}_{i+1}^\sigma \neq \emptyset, \mathcal{D}_{i+2}^\sigma = \emptyset$ (see Fig. 2(c) where $\sigma_i = C$)): If $d \in \sigma_i^1$ (see Fig. 2(c) where $\sigma_i = C$ and $\sigma_i^1 = C^1$) then d may appear in the solutions of (i) the LSDUDC algorithm with respect to a vertical line L_{i-2} (line 22 of Algorithm 1) and (ii) the RLSDUDC algorithm with respect to the line L_{i-1} (line 17 of Algorithm 1). In the optimum solution d may cover some points in $\mathcal{P}_{i-2}^\sigma$ and some points in $\mathcal{P}_{i+1}^\sigma$. In Algorithm 1, we choose one disk in $\mathcal{D}_{i-2}^\sigma$

and one disk in $\mathcal{D}^{\sigma}_{i+1}$ to cover all the points in $\mathcal{P}^{\sigma}_{i-2}$ and $\mathcal{P}^{\sigma}_{i+1}$, respectively. If no disk in $\mathcal{D}^{\sigma}_{i}$ appears in the solution at the time of running the LSDUDC and RLSDUDC algorithms, then d may appear in the solution (line 26 of Algorithm 1) to cover the points in $\mathcal{P}^{\sigma}_{i}$. Algorithm 1 thus chooses at most $2+1+1+1=5$ disks instead of d in this case.

If $d \in \sigma^2_i$ (see Fig. 2(c) where $\sigma^2_i = C^2$), then d may appear in the solutions of (i) the LSDUDC algorithm with respect to a vertical lines L_{i-2} and L_i (line 22 of Algorithm 1) and (ii) the RLSDUDC algorithm with respect to the line L_{i-1} (line 17 of Algorithm 1). Again, in the optimum solution d may cover some points in $\mathcal{P}^{\sigma}_{i+1}$. In Algorithm 1, we choose one disk in $\mathcal{D}^{\sigma}_{i+1}$ to cover all the points in $\mathcal{P}^{\sigma}_{i+1}$. If no disk in $\mathcal{D}^{\sigma}_{i}$ appears in the solution at the time of running the LSDUDC and RLSDUDC algorithms, then d may appear in the solution (line 26 of Algorithm 1) to cover the points in $\mathcal{P}^{\sigma}_{i}$. Therefore, Algorithm 1 may choose at most $2 \times 2+1+1=6$ disks instead of d in this case.

Case 4 ($\mathcal{D}^{\sigma}_{i-2} = \emptyset, \mathcal{D}^{\sigma}_{i-1} \neq \emptyset, \mathcal{D}^{\sigma}_{i+1} = \emptyset$ (see Fig. 2(c) where $\sigma_i = D$)): If $d \in \sigma^1_i$ (see Fig. 2(c) where $\sigma_i = D$ and $\sigma^1_i = D^1$), then d may appear in the solutions of (i) the LSDUDC algorithm with respect to a vertical lines L_{i-3} and L_i (line 22 of Algorithm 1) and (ii) the RLSDUDC algorithm with respect to the line L_{i-2} (line 17 of Algorithm 1). Again, in the optimum solution d may cover some points in $\mathcal{P}^{\sigma}_{i-1}$. In the proposed algorithm, we choose one disk in $\mathcal{D}^{\sigma}_{i-1}$ to cover all the points in $\mathcal{P}^{\sigma}_{i-1}$. Note that, if no disk in $\mathcal{D}^{\sigma}_{i}$ appears in the solution at the time of running the LSDUDC and RLSDUDC algorithms, then d may appear in the solution (line 26 of Algorithm 1) to cover the points in $\mathcal{P}^{\sigma}_{i}$. Therefore Algorithm 1 chooses at most $2 \times 2+1+1=6$ disks instead of d.

If $d \in \sigma^2_i$ (see Fig. 2(c) where $\sigma^2_i = D^2$), then d may appear in the solutions of (i) the LSDUDC algorithm with respect to a vertical lines L_{i-3} and L_i (line 22 of Algorithm 1). Again, in the optimum solution d may cover some points in $\mathcal{P}^{\sigma}_{i-1}$. In Algorithm 1, we choose one disk in $\mathcal{D}^{\sigma}_{i-1}$ to cover all the points in $\mathcal{P}^{\sigma}_{i-1}$. If no disk in $\mathcal{D}^{\sigma}_{i}$ appears in the solution of above LSDUDC algorithm then d may appear in the solution (line 26 of Algorithm 1) to cover the points in $\mathcal{P}^{\sigma}_{i}$. Thus Algorithm 1 chooses at most $2 \times 2+1=5$ disks instead of d in this case.

The time complexity of the lemma follows from the fact that each disk in $\mathcal{D}$ can participate a constant number of times in the LSDUDC and RLSDUDC algorithms and the running time of both the LSDUDC and RLSDUDC algorithms is $O(n \log n + mn)$ (see Theorem 1). □

4.2 Outside Strip Point Cover Problem

Let $\mathcal{D}^U_i$ and $\mathcal{D}^L_i$ be the set of disks centered above and below the line ℓ_i (defined in Section 4) respectively.

Compute the sets $\mathcal{P}^L_1, \mathcal{P}^L_2, \ldots, \mathcal{P}^L_t, \mathcal{P}^U_t, \mathcal{P}^U_{t-1}, \ldots, \mathcal{P}^U_1 \subseteq \mathcal{P}$ in order as follows:

Set $\mathcal{P}' = \mathcal{P}$. For each $i = 1, 2, \ldots, t$ compute $\mathcal{P}^L_i = \mathcal{P}' \cap \mathcal{D}^U_i$ and set $\mathcal{P}' = \mathcal{P}' \setminus \mathcal{P}^L_i$. For each $i = t, t-1, \ldots, 1$ compute $\mathcal{P}^U_i = \mathcal{P}' \cap \mathcal{D}^L_i$ and set $\mathcal{P}' = \mathcal{P}' \setminus \mathcal{P}^U_i$.

Note that all sets $\mathcal{P}_i^L$ are covered by the disks centered inside the horizontal strip $[\ell_{i-1}, \ell_i]$. Similarly, all sets $\mathcal{P}_i^U$ are covered by the disks in $\mathcal{D}$ whose centers are inside the horizontal strip $[\ell_i, \ell_{i+1}]$ but no points in $\mathcal{P}_i^U$ are covered by a disk centered above the line ℓ_{i-1}, where ℓ_{i+1} is a horizontal line below ℓ_i.

Without loss of generality assume that $t = 4s$. Now, consider six sets of sets related to $\mathcal{P}_i$ for $i = 1, 2, \ldots 6$ defined below:

$$\begin{aligned}
\mathcal{S}_1 &= \{\mathcal{Q}_{1i}(= \mathcal{P}^L_{1+4i})|i = 0, 1, \ldots, s-1\}\\
\mathcal{S}_2 &= \{\mathcal{Q}_{2i}(= \mathcal{P}^L_{2+4i})|i = 0, 1, \ldots, s-1\}\\
\mathcal{S}_3 &= \{\mathcal{Q}_{3i}(= \mathcal{P}^L_{3+4i})|i = 0, 1, \ldots, s-1\}\\
\mathcal{S}_4 &= \{\mathcal{Q}_{4i}(= \mathcal{P}^L_{4+4i})|i = 0, 1, \ldots, s-1\}\\
\mathcal{S}_5 &= \{\mathcal{Q}_{5i}(= \mathcal{P}^U_{1+2i})|i = 0, 1, \ldots, 2s-2\}\\
\mathcal{S}_6 &= \{\mathcal{Q}_{6i}(= \mathcal{P}^U_{2+2i})|i = 0, 1, \ldots, 2s-2\}
\end{aligned}$$

We characterize the elements in each of the sets $\mathcal{S}_i$ for $i = 1, 2, \ldots, 6$ using the following lemma:

Lemma 3. *If $p \in \mathcal{Q}_{uv}$ and $p' \in \mathcal{Q}_{uw}$ are two points such that for $u = 1, 2, 3, 4$, $v \neq w$ and $v, w \in \{1, 2, \ldots s-1\}$, and for $u = 5, 6$, $v \neq w$ and $v, w \in \{1, 2, \ldots, 2s-2\}$, then a single disk in $\mathcal{D}$ cannot cover both the points p and p'.*

Proof. We prove the lemma for $u = 1$ and $u = 5$. A proof for the other cases arises in a similar fashion.

Case $u = 1$: From the definition of $\mathcal{Q}_{1v}$ and $\mathcal{Q}_{1w}$; $\mathcal{Q}_{1v} = \mathcal{P}^L_{1+4v}$ and $\mathcal{Q}_{1w} = \mathcal{P}^L_{1+4w}$. Now, from the definition of $\mathcal{P}^L_{1+4v}$ and $\mathcal{P}^L_{1+4w}$ each element of $\mathcal{P}^L_{1+4v}$ and $\mathcal{P}^L_{1+4w}$ is covered by the disks centered inside the horizontal strip $[\ell_{4v},\ \ell_{1+4v}]$ and $[\ell_{4w},\ \ell_{1+4w}]$, respectively. Since $v \neq w$ and the height of each horizontal strip is $\frac{1}{\sqrt{2}}$, so the distance between a point inside the strip $[\ell_{4v},\ \ell_{1+4v}]$ and a point inside the strip $[\ell_{4w},\ \ell_{1+4w}]$ is greater than 2. Thus, in this case the lemma follows from the fact that the disks are unit radius disks.

Case $u = 5$: From the definition of $\mathcal{Q}_{5v}$ and $\mathcal{Q}_{5w}$; $\mathcal{Q}_{5v} = \mathcal{P}^U_{1+2v}$ and $\mathcal{Q}_{5w} = \mathcal{P}^U_{1+2w}$. Now, from the definition of $\mathcal{P}^U_{1+2v}$ each element of $\mathcal{P}^U_{1+2v}$ is covered by the disks centered inside the horizontal strip $[\ell_{1+2v},\ \ell_{2+2v}]$, but no points in $\mathcal{P}^U_{1+2v}$ are covered by a disk centered above the line ℓ_{2v}. Similarly, from the definition of $\mathcal{P}^U_{1+2w}$, each element of $\mathcal{P}^U_{1+2w}$ is covered by the disks centered inside the horizontal strip $[\ell_{1+2w},\ \ell_{2+2w}]$, but no points in $\mathcal{P}^U_{1+2w}$ are covered by a disk centered above the line ℓ_{2w}. Since $v \neq w$, there does not exist a disk which covers one point in $\mathcal{P}_L^{1+2v}$ and a point in $\mathcal{P}_L^{1+2w}$. $\square$

We now describe the algorithm for covering the points in $\mathcal{P}^L_1, \mathcal{P}^L_2, \ldots, \mathcal{P}^L_t, \mathcal{P}^U_t, \mathcal{P}^U_{t-1}, \ldots, \mathcal{P}^U_1$. The algorithm executes the following steps:

Lemma 4. *Algorithm 2 produces a 12-factor approximation result in $O(n \log n + mn)$ time.*

Algorithm 2 OSDUDC($\mathcal{P}, \mathcal{D}$)

```
Input: A points set P and a unit disks set D.
Output: A set D*_2 ⊆ D.
D*_2 ← ∅
for (i = 1, 2, ..., t) do
    Compute the set P_i^L
end for
for (i = 1, 2, ..., t) do
    Run the LSDUDC algorithm on the point set P_i^L based on the line ℓ_i. Let D' ⊆ D
    be the output of the algorithm. Set D*_2 = D*_2 ∪ D'.
end for
for (i = t, t − 1, ..., 1) do
    Compute the set P_i^U
end for
for (i = t, t − 1, ..., 1) do
    Run the LSDUDC algorithm on the point set P_i^U based on the line ℓ_i. Let
    D' ⊆ D be the output of the algorithm. Set D*_2 = D*_2 ∪ D'.
end for
return D*_2
```

Proof. The set of points in $\mathcal{P}_1^L, \mathcal{P}_2^L, \ldots, \mathcal{P}_t^L, \mathcal{P}_t^U, \mathcal{P}_{t-1}^U, \ldots, \mathcal{P}_1^U$ can be partitioned into 6 sets of sets such that a disk cannot cover two points from a set $\mathcal{S}_i$ such that the two points are in different subsets $\mathcal{Q}_{ij'}$, $\mathcal{Q}_{ij''}$, $j' \neq j''$ (see Lemma 3). Since we can divide the plane into strips such that each strip contains at least one point in $\mathcal{P}$, $t = O(n)$. Thus, the approximation factor of the lemma follows from the fact that the LSDUDC algorithm produces a 2-factor approximation result (see Theorem 1) and can participate in at most 6 sets. The time complexity of the lemma follows from the fact that (i) each disk in $\mathcal{D}$ can participate 6 times in the LSDUDC algorithm and (ii) the running time of LSDUDC algorithm is $O(n \log n + mn)$ (see Theorem 1). □

Now, we describe the algorithm (Algorithm 3) for the DUDC problem using Algorithm 1 and Algorithm 2 as follows:

Theorem 2. *Algorithm 3 produces an 18-approximation result in $O(m \log m + n \log n + mn)$ time for the discrete unit disk cover problem.*

Proof. The approximation result follows from Lemma 2 and Lemma 4. The time complexity follows from the following facts:

(i) In line number 4, feasibility checking of the DUDC problem needs $O(m \log m + n \log m)$ time (see Section 4).

(ii) In line number 10, the running time of all WSDUDC($\mathcal{P}_i, \mathcal{D}_i$) is $O(|\mathcal{P}_i| \log |\mathcal{P}_i| + |\mathcal{D}_i||\mathcal{P}_i|)$ (see Lemma 2). Therefore, the running time of the for loop (lines 7-11) is $O(n \log n + mn)$.

(iii) In line number 13, the running time of OSDUDC($\mathcal{P}, \mathcal{D}$) is $O(n \log n + mn)$ (see Lemma 4).

Algorithm 3 DUDC($\mathcal{P}, \mathcal{D}$)

Input: A points set $\mathcal{P}$ and a unit disks set $\mathcal{D}$.
Output: A set $\mathcal{D}^* \subseteq \mathcal{D}$.
$\mathcal{D}^* \leftarrow \emptyset$ and $\mathcal{D}_1^* \leftarrow \emptyset$.
Check all the points in $\mathcal{P}$ is covered by at least one disk in $\mathcal{D}$ or not. If the answer is yes then set *flag* = *true*, otherwise set *flag* = *false*.
if (flag = true) **then**
  Let $\mathcal{H}_i = [\ell_i, \ell_{i+1}]$ for $i = 0, 1, \ldots, t-1$ be the horizontal strips of width $\frac{1}{\sqrt{2}}$ defined in Section 4.
  for (i=0,1, ..., t-1) **do**
    Let $\mathcal{D}_i \in \mathcal{D}$ be the set of disks having centers in $\mathcal{H}_i$.
    Let $\mathcal{P}_i \in \mathcal{P} \cap \mathcal{H}_i$ be the set of points such that each point in $\mathcal{P}_i$ is covered by the disks in $\mathcal{D}_i$ only.
    Run the algorithm WSDUDC($\mathcal{P}_i, \mathcal{D}_i$) proposed in Algorithm 1. Let $\mathcal{D}_1^i$ be the output of this algorithm.
  end for
  $\mathcal{D}_1^* = \mathcal{D}_1^0 \cup \mathcal{D}_1^1 \ldots \cup \mathcal{D}_1^{t-1}$
  Run the algorithm OSDUDC($\mathcal{P}, \mathcal{D}$) proposed in Algorithm 2. Let $\mathcal{D}_2^*$ be the output of this algorithm.
  $\mathcal{D}^* = \mathcal{D}_1^* \cup \mathcal{D}_2^*$
end if
return $\mathcal{D}^*$

5 Conclusion

Here, we have considered the discrete unit disk cover (DUDC) problem where a set $\mathcal{P}$ of n points and a set $\mathcal{D}$ of m unit disks in 2-dimensional plane are given, the objective is (i) to check whether each point in $\mathcal{P}$ is covered by at least one disk in $\mathcal{D}$ or not and (ii) if so, then find a minimum cardinality subset $\mathcal{D}^* \subseteq \mathcal{D}$ such that unit disks in $\mathcal{D}^*$ cover all the points in $\mathcal{P}$. We provide an 18-factor approximation algorithm for the DUDC problem. The running time of the proposed algorithm is $O(n \log n + m \log m + mn)$. The previous best known solution for the same problem was a 22-factor approximation algorithm with running time $O(m^2 n^4)$. Therefore, our solution is a significant improvement over the best known solution in terms of approximation factor as well as running time of the algorithm.

Acknowledgements Funding for this project was provided by the Natural Sciences and Engineering Research Council of Canada (NSERC), partially under the NSERC Strategic Grant on Optimal Data Structures for Organization and Retrieval of Spatial Data.

References

1. P. K. Agarwal and C. M. Procopiuc, *Exact and approximation algorithms for clustering*, Algorithmica, vol 33, pp. 201-226, 2002.

2. P. K. Agarwal and M. Sharir, *Efficient algorithms for geometric optimization*, ACM Computing Serveys, vol 30, pp. 412-458, 1998.
3. M. de. Berg, M. V. Kreveld, M. Overmars and O. Schwarzkopf, Computational Geometry Algorithms and Applications, Springer-Verlag, 1997.
4. H. Brönnimann and M. Goodrich. *Almost optimal set covers in finite VC-dimension*, *Disc. and Comp. Geom.*, 14(1):463–479, 1995.
5. F. Claude, G. K. Das, R. Dorrigiv, S. Durocher, R. Fraser, A. Lopez-Ortiz, B. G. Nickerson and A. Salinger, *An improved line-separable algorithm for discrete unit disk cover*, Disc. Math., Alg. and Appl., vol 2, pp. 77-87, 2010.
6. G. Călinescu, I. Măndoiu, P.J. Wan and A. Zelikovsky, *Selecting forwarding neighbours in wireless ad hoc networks*, Mobile Networks and Applications, vol 9, pp. 101-111, 2004.
7. P. Carmi, M. J. Katz and N. Lev-Tov, *Covering points by unit disks on fixed location*, Proc. of the 18th International Symposium on Algorithms and Computation, LNCS - 4835, pp. 644-655, 2007.
8. V. Chvátal. *A Greedy Heuristic for the Set-Covering Problem*, *Mathematics of Operations Research*, 4(3):233–235, 1979.
9. R. Fowler, M. Paterson and S. Tanimoto, *Optimal packing and covering in the plane are NP-complete*, Information Processing Letters, vol 12, pp. 133-137, 1981.
10. G. Frederickson, *Fast algorithms for shortest paths in planar graphs, with applications*, SIAM J. on Computing, vol 16, pp. 1004-1022, 1987.
11. T. Gonzalez, *Covering a set of points in multidimensional space*, Information Processing Letters, vol 40, pp. 181-188, 1991.
12. D. Hochbaum and W. Maass, *Approximation schemes for covering and packing problems in image processing and VLSI*, J. ACM, vol 32, pp. 130-136, 1985.
13. R. Hwang, R. Lee and R. Chang, *The generalized searching over separators strategy to solve some NP-hard problems in subexponential time*, Algorithmica, vol 9, pp. 398-423, 1993.
14. D. S. Johnson, *The NP-completeness column:An ongoing guide*, J. Algorithms, vol 3, pp. 182-195, 1982.
15. I. Koutis and G. L. Miller, *A linear work, $o(n^{1/6})$ time, parallel algorithm for solving planar Laplacians*, Proc. of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1002-1011, 2007.
16. N. H. Mustafa and S. Ray, *Improved results on geometric hitting set problems*, Discrete and Computational Geometry, to appear.
17. S. Narayanappa and P. Vojtechovsky, *An improved approximation factor for the unit disk covering problem*, Proc. of the 18th Canadian Conference on Computational Geometry, pp. 15-18, 2006.
18. R. Raz and S. Safra, *A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP*, Proc. of the 29th ACM Symposium on Theory of Computing, pp. 475-484, 1997.