AN IMPROVED LINE-SEPARABLE ALGORITHM FOR DISCRETE UNIT DISK COVER

FRANCISCO CLAUDE*,1, GAUTAM K. DAS†,2, REZA DORRIGIV‡,1, STEPHANE DUROCHER§,3, ROBERT FRASER¶,1, ALEJANDRO LÓPEZ-ORTIZ‖,1, BRADFORD G. NICKERSON**,2, and ALEJANDRO SALINGER††,1

[1] *David R. Cheriton School of Computer Science, University of Waterloo,, 200 University Ave. West, Waterloo, Ontario, N2L3G1, Canada*
[2] *Faculty of Computer Science, University of New Brunswick, P.O. Box 4400,, 540 Windsor Street, Fredericton, New Brunswick, E3B 5A3, Canada*
[3] *Department of Computer Science, E2-445 EITC, University of Manitoba, Winnipeg, Manitoba, R3T 2N2, Canada*

December 15, 2009

Abstract

Given a set $\mathcal{D}$ of m unit disks and a set $\mathcal{P}$ of n points in the plane, the discrete unit disk cover problem is to select a minimum cardinality subset $\mathcal{D}' \subseteq \mathcal{D}$ to cover $\mathcal{P}$. This problem is NP-hard [14] and the best

*fclaude@cs.uwaterloo.ca
†gdas@unb.ca
‡rdorrigiv@cs.uwaterloo.ca
§durocher@cs.umanitoba.ca
¶r3fraser@cs.uwaterloo.ca
‖alopez-o@cs.uwaterloo.ca
**bgn@unb.ca
††ajsalinger@cs.uwaterloo.ca

previous practical solution is a 38-approximation algorithm by Carmi et al. [5]. We first consider the line-separable discrete unit disk cover problem (the set of disk centers can be separated from the set of points by a line) for which we present an $O(n(\log n+m))$-time algorithm that finds an exact solution. Combining our line-separable algorithm with techniques from the algorithm of Carmi et al. [5] results in an $O(m^2n^4)$ time 22-approximate solution to the discrete unit disk cover problem.

1 Introduction

Recent interest in specific geometric set cover problems is partly motivated by applications in wireless networking. In particular, when wireless clients and servers are modeled as points in the plane and the range of wireless transmission is assumed to be constant (say one unit), the resulting region of wireless communication is a disk of unit radius centered on the point representing the corresponding wireless transmitting device. Under this model, sender a successfully transmits a wireless message to receiver b if and only if point b is covered by the unit disk centered at point a. This model applies more generally to a variety of facility location problems for which the Euclidean distance between clients and facilities cannot exceed a given radius, and clients and candidate facility locations are represented by discrete sets of points. Examples include:

- selecting locations for wireless servers (e.g., gateways) from a set of candidate locations to cover a set of wireless clients,
- positioning a fleet of water bombers at airports such that every active forest fire is within a given maximum distance of a water bomber,
- selecting a set of weather radar antennae to cover a set of cities, and
- selecting locations for anti-ballistic defenses from a set of candidate locations to cover strategic sites.

These problems can be modeled by the *discrete unit disk cover problem* (DUDC).

Definition 1. *Given a set $\mathcal{P}$ of n points and a set $\mathcal{O}$ of m points in the plane (candidate clients and facilities, respectively), the discrete unit disk cover problem is to find a set $\mathcal{O}' \subseteq \mathcal{O}$ (facilities) of minimum cardinality* *such that* $\text{Disk}(\mathcal{O}')$ *covers* $\mathcal{P}$*, where* $\text{Disk}(\mathcal{A})$ *denotes the set of unit disks centered on points in set* $\mathcal{A}$*.*

In this work, we consider the *line-separable discrete unit disk cover* (LSDUDC), where $\mathcal{P}$ and $\mathcal{O}$ are separated by a line l. We may arbitrarily set l to be horizontal, and further have the set $\mathcal{O}$ lie in the region above the line l. For clarification, we relabel $\mathcal{O}$ as $\mathcal{U}$ in the LSDUDC setting, denoting that the points of $\mathcal{U}$ are restricted to the upper half-plane defined by l.

- Input: A set $\mathcal{P}$ of n points in the plane (clients), and a set $\mathcal{U}$ of m points in the plane (candidate facilities), where $\mathcal{P}$ and $\mathcal{U}$ may be separated by a line.
- Output: Find a set $\mathcal{U}' \subseteq \mathcal{U}$ (facilities) of minimum cardinality such that $\text{Disk}(\mathcal{U}')$ covers $\mathcal{P}$.

The DUDC problem is NP-hard [14], and the general set cover problem (i.e. the covering shapes are unrestricted) is not approximable within a factor $c \log n$, for any constant c [20]. In a recent result, Carmi et al. [5] describe a polynomial-time 38-approximate solution, improving on earlier 108-approximate [4] and 72-approximate solutions [19].

1.1 Our Results

We present an $O(n(\log n + m))$-time algorithm that returns an exact solution to the LSDUDC problem, as well as a thorough proof of correctness of the technique. By combining the LSDUDC algorithm with techniques from the algorithm of Carmi et al. [5], we present a 22-approximation algorithm to the DUDC problem, improving on the best previous practical polynomial-time approximation factor of 38.

1.2 Related Work

Line-Separable Discrete Unit Disk Cover. A solution to the LSDUDC problem was independently discovered and published by Ambühl et al. [3, Lemma 1], where they propose a dynamic programming algorithm with a time bound of $O(m^2 n)$ but whose correctness is not straightforward nor is it formally argued. This paper presents a faster algorithm together with a proof of correctness. We then observe that our new algorithm can be combined with a suitably modified version of the algorithm of Carmi et al. [5] to achieve an improved approximation factor for the general DUDC problem.

A similar problem to LSDUDC is studied in [4, 8], but their setting has the centers of the disks within a specified unit disk and the points to be covered are outside that disk.

Local Search for Geometric Hitting Problems. Using local search, Mustafa and Ray [17, 18] have recently presented a $(1+\epsilon)$-approximation to the DUDC problem for any $\epsilon \in (0, 1]$. Their algorithm runs in $O(m^{2(c/\varepsilon)^2+1}n)$ time, where $c \leq 4\gamma$ [18]. The value of γ can be bounded from above by $2\sqrt{2}$ [10, 15]. The fastest operation of this algorithm is obtained when $\varepsilon = 1$ for a 2-approximation, resulting in a running time of $O(m^{2\cdot(8\sqrt{2})^2+1}n) = O(m^{257}n)$ in the worst case. The corresponding running time increases for any $\epsilon < 1$. Clearly, this algorithm is not practical for large values of m. It is possible that a lower running time may be obtained through better bounding of the constant factors or improvements to their algorithm, but a practical implementation appears unlikely.

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 our problem, 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, 21] and has a PTAS solution [11, 12]. Of course the problem can be generalized further: see [6] for a discussion of geometric set cover problems.

Discrete *k*-Center. Also related is the discrete Euclidean k-center problem: given a set $\mathcal{O}$ of m points in the plane, a set $\mathcal{P}$ of n points in the plane, and an integer k, find a set of k disks centered on points in $\mathcal{O}$ whose union covers $\mathcal{P}$ such that the radius of the largest disk is minimized. Observe that set $\mathcal{P}$ has a discrete unit disk cover consisting of k disks centered on points in $\mathcal{O}$ if and only if $\mathcal{P}$ has a discrete k-center centered on points in $\mathcal{O}$ with radius at most one. This problem is NP-hard if k is an input variable [2]. When k is fixed, Hwang et al. [13] give a $m^{O(\sqrt{k})}$-time algorithm, and Agarwal and Procopiuc [1] give an $m^{O(k^{1-1/d})}$-time algorithm for points in $\mathbb{R}^d$.

2 Line Separated Discrete Unit Disk Cover

We begin by introducing notation and terminology for our discussion of the line-separable unit disk cover problem (LSDUDC). Here, two sets of points $\mathcal{U} = \{u_1, u_2, \ldots, u_m\}$ and $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$ are given. A horizontal line l is given such that each point in $\mathcal{U}$ is above l and each member of $\mathcal{P}$ is below l. Further, for each $p_i \in \mathcal{P}$, there exists at least one $u_j \in \mathcal{U}$ which maintains the condition $\delta(p_i, u_j) \leq 1$, where $\delta(p_i, u_j)$ is the Euclidean distance between points p_i and u_j.

If a point $u_j \in \mathcal{U}$ (resp. $p_i \in \mathcal{P}$) is within a unit disk centered at a point $p_i \in \mathcal{P}$ (resp. $u_j \in \mathcal{U}$), then we use the term u_j is covered by p_i (resp. p_i is covered by u_j). Let d_i be the circle of unit radius centered at the point $p_i \in \mathcal{P}$ (i.e. $d_i = \text{Disk}(p_i)$). Also, let $\mathit{left}(i)$ be the left and $\mathit{right}(i)$ be the right intersection points of d_i with the horizontal line l. Without loss of generality, rename the points in $\mathcal{P}$ based on the intersection points $\{\mathit{left}(i) : p_i \in \mathcal{P}\}$ from left to right order. Let l^+ be the region above the line l and l^- be the region below the line l. Let $C(d_i) \subseteq \mathcal{U}$ be the set of points covered by the circle d_i centered at point $p_i \in \mathcal{P}$.

This formulation allows us to address the problem in the dual setting. Rather than seeking a subset of disks $\text{Disk}(\mathcal{U}') \subseteq \text{Disk}(\mathcal{U})$ which covers all of the points in $\mathcal{P}$, we are seeking a minimum cardinality subset of points $\mathcal{U}' \subseteq \mathcal{U}$ such that each disk in $\text{Disk}(\mathcal{P})$ is stabbed by at least one point in $\mathcal{U}'$, as shown in Figure 1.

In Algorithm 1, we present an algorithm for covering all the points in $\mathcal{P}$ using a minimum number of unit radius circles centered at points in $\mathcal{U}$. We show that this algorithm produces the optimum result. Note that covering all the points in $\mathcal{P}$ by the minimum number of unit circles centered at points in $\mathcal{U}$ implies each of the circles d_i corresponding to $p_i \in \mathcal{P}$ contains at least one point in $\mathcal{U}$.

Lemma 1. *In the arrangement of the circles centered at points in $\mathcal{P}$, if the circles d_i and d_j intersect, then at least one intersection point is in l^-. In other words, at most one intersection point of d_i and d_j is in l^+.*

Proof. Let p_i and p_j be the centers of the circles d_i and d_j respectively. We assume that d_i and d_j are circles of unit radius. If two circles of the same radius intersect, then both the intersection points lie on the line which is the perpendicular bisector of the segment joining their centers p_i and p_j. If both

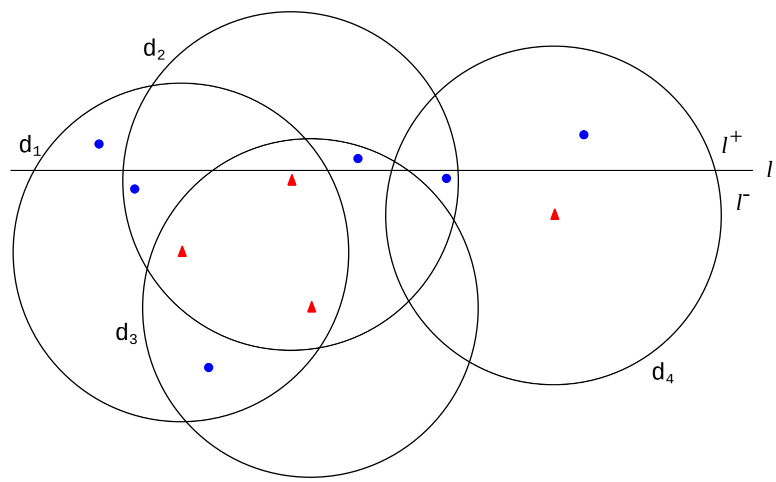

Figure 1: Demonstration of the arrangement of circles centered at points indicated by triangles.

the intersection points are in l^+, then at least one of either p_i or p_j must be in l^+, which leads to a contradiction as points $\mathcal{P}$ lie in l^-. $\square$

Claim 1. *In the arrangement of the circles centered at points in $\mathcal{P}$, if the circles d_i and d_j intersect in l^+ and left$(i) <$ left(j) then right$(i) <$ right(j). For example, see the circles d_1 and d_2 in Figure 1.*

Proof. The claim follows from Lemma 1. $\square$

Claim 2. *In the arrangement of the circles centered at points in $\mathcal{P}$, if the circles d_i and d_j intersect and if both the intersection points are in l^- such that left$(i) <$ left$(j) <$ right(i), then $d_j \cap l^+ \subset d_i \cap l^+$.*

Proof. The result follows from the fact that there are at most two points of intersection between two distinct circles, and *left*$(i) <$ *left*$(j) <$ *right*(i). For a demonstration, see the circles d_2 and d_3 in Figure 1. $\square$

Theorem 1. *Algorithm 1 returns an optimal solution to the line-separable unit disk cover problem.*

Proof. Let $\mathcal{U}' = \{u'_1, u'_2, \ldots, u'_{m'}\}$ denote the set of points in the solution returned by Algorithm 1. Therefore, $d_1 \cap \mathcal{U}' \neq \varnothing$. Consider the value of k in Algorithm 1; if $k+1$ is the minimum index such that the region $\cap_{i=1}^{k+1} d_i$

Algorithm 1 LSDUDC($\mathcal{P}$,$\mathcal{U}$)

Input: Set $\mathcal{U}$ of points in l^+, and set $\mathcal{P}$ of points in l^-.
Output: Set $\mathcal{U}'$ of points covering all the points in $\mathcal{P}$.
$j \leftarrow 1, \mathcal{U}' \leftarrow \varnothing$
Sort $\mathcal{P}$ according to $left(i)$.
Compute the sets $C(d_i)$ of points for each $i = 1, 2, \ldots, n$.
while $(j \neq n+1)$ **do**
  Find the maximum index k such that $\cap_{i=j}^{k} C(d_i) \neq \varnothing$, but $\cap_{i=j}^{k+1} C(d_i) = \varnothing$.
  Let s be the rightmost point in $\cap_{i=j}^{k} C(d_i)$.
  $j \leftarrow k+1, \mathcal{U}' = \mathcal{U}' \cup \{s\}$.
end while
return $\mathcal{U}'$

does not contain any point from $\mathcal{U}$, then the algorithm chooses a point $s \in \cap_{i=1}^{k} C(d_i)$ such that s is the rightmost point in $\cap_{i=1}^{k} C(d_i)$. The same process is repeated from d_{k+1} onwards. We show that there exists an optimal solution containing point s.

In the optimal solution, the covering of $p_1, p_2, \ldots, p_{k+1}$ requires two points from $\mathcal{U}$. Therefore, if there exists a point $s' \in l^+$ which can cover both p_1 and p_{k+1}, then there exists at least one point $p_a \in \{p_2, p_3, \ldots, p_k\}$ such that p_a is not covered by s'. To cover the point p_a, we need one more point, say $s'' \in \mathcal{U}$. Let us now analyze the possible positions of s'' by partitioning the region $d_a \cap l^+$ (recall that d_a is the circle centered at point p_a) into three disjoint subregions as follows (see Figure 2):

RG_1**:** $(d_a \cap l^+) \setminus (d_{k+1} \cap l^+)$ (dark-shaded region)

RG_2**:** $d_1 \cap d_a \cap d_{k+1} \cap l^+$ (dotted region)

RG_3**:** $(d_a \cap l^+) \setminus (d_1 \cap l^+)$ (light-shaded region)

If $s'' \in RG_1$, then we may choose s instead of s'' for covering d_a, which in turn, covers $d_1, d_2, \ldots, d_k$. The circle d_{k+1} may be covered by some other point in $\mathcal{U}$ (which may be different from s'), which in turn may cover some other circle d_j, where $j > k+1$. Thus, the choice of $s \in \mathcal{U}'$ is correct.

If $s'' \in RG_2$, then d_1, d_a and d_{k+1} can be covered by a single point $s'' \in \mathcal{U}$. This implies that the purpose of choosing s' for covering d_1 and d_{k+1} jointly

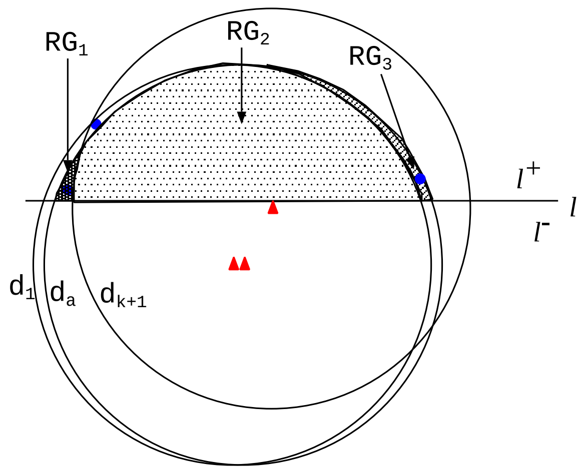

Figure 2: Illustration in support of Theorem 1.

can be served by s''. But, since $\cap_{i=1}^{k+1} C(d_i)$ is empty, there exists some other member $d_{a'}$, $a' \leq k$ which is not covered by s''. This situation happens for all choices of $s'' \in RG_2$ to cover d_1 and d_{k+1} jointly.

Algorithm 1 selects $s \in \mathcal{U}'$ for covering $d_1, d_2, \ldots, d_k$, and it also is free to choose some other point in $\mathcal{U}$ (possibly different from s'') which can cover d_{k+1} and some other circle d_j, where $j > k+1$.

Finally, if $s'' \in RG_3$, then $s'' \in d_{k+1}$ since $s' \in (d_1 \cap d_{k+1} \cap l^+) \setminus (d_a \cap l^+)$ (see Figure 2). As in the earlier case, the choice of $s \in \mathcal{U}'$ (instead of s') for covering $d_1, d_2, \ldots, d_k$ is fine. d_{k+1}, along with some other circles d_j, $j > k+1$, possibly covered by some other point (may be s'') in l^+. □

2.1 Analysis of the LSDUDC Algorithm

Theorem 2. *Algorithm 1 has a worst-case running time of* $O(n(\log n + m))$.

Proof. We examine each significant step of the algorithm in turn to determine the running time.

1. Sort the points in $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$ from left to right order.

2. Compute the sets $C(d_i)$ of points for each $i = 1, 2, \ldots, n$.

3. While($j \neq n+1$), find the maximum index $k(\leq n)$ such that $\cap_{i=j}^{k} C(d_i) \neq \varnothing$, but $\cap_{i=j}^{k+1} C(d_i) = \varnothing$ for $k < n$.

The first step may be completed in $O(n \log n)$ time using a standard sorting technique, and the second step may be done brute force in $O(mn)$ time. In the while loop, there is at most one iteration for each point in $\mathcal{P}$, and the algorithm runs greedily and thus each point in $\mathcal{P}$ is used at most twice. Checking for membership in the intersection of all disks seen so far may be done in linear time in the number of points in $\mathcal{U}$, for a total worst case time of $O(mn)$ for the while loop. Therefore, the total running time of the Algorithm 1 is $O(n(\log n + m))$. □

3 Approximate Discrete Unit Disk Cover

We now show that Algorithm 1 for the line-separable discrete unit disk cover (LSDUDC) problem leads to a 22-approximation algorithm for the discrete unit disk cover (DUDC) problem. The approximation algorithm is based on a suitable adaptation of the 38-approximation algorithm of Carmi et al. [5]. In that work, the DUDC problem is supported by a variant of the LSDUDC problem: suppose we are given a set of disks $\mathcal{D} = \mathcal{L} \cup \mathcal{U}$. The disks in $\mathcal{U}$ are centered above a line l, and the set $\mathcal{L} = \mathcal{D} \backslash \mathcal{U}$ are centered below l. We are also given a set of points $\mathcal{P}$ covered by $\mathcal{U}$. The goal is to obtain the set $\mathcal{G} \subseteq \mathcal{D}$ of smallest cardinality such that every point in $\mathcal{P}$ is covered by a disk in $\mathcal{G}$.

Note that our line-separable algorithm does not immediately result in a straightforward improvement to the approximation factor of the algorithm of Carmi et al. Their proof of correctness uses the fact that their 2-approximation to the LSDUDC problem consists of disks forming the lower boundary of $\mathcal{U}$, which is defined as the semi-chain.

Definition 2 (Carmi et al. [5])**.** *The* semi-chain $\mathcal{S}$ *is the ordered (from left to right) set of all lower circular arcs below the line l of the disks in $\mathcal{U}$. The set of indices associated with $\mathcal{S}$ forms a consecutive set of indices $i, i+1, \ldots, j$ for $i \leq j$. Carmi et al. call an interval from i to j an* interval cell *and denote it by* $\texttt{icell}(i, j)$*. Let B denote the region $l^- \cap (\cup_{i=1}^{m} u_i)$ $(u_i \in \mathcal{U})$, which corresponds to the region below l contained by all of the circular arcs in $\mathcal{S}$.*

Our solution does not necessarily use disks that contribute to the semi-chain $\mathcal{S}$. Instead, we first solve the LSDUDC problem optimally using Algorithm 1 on the set of disks $\mathcal{U}$ to obtain a disk set $\mathcal{U}'$. Let $\mathcal{U}' = \{u'_1, u'_2, \ldots, u'_{m'}\}$ be the ordered set of unit disks from left to right based on the left intersection point of l with the disks in $\mathcal{U}'$. We then use the greedy *minimum assisted cover* algorithm of Carmi et al. [5] over the sets $\mathcal{U}'$ and $\mathcal{L}$ to obtain an improved solution $\mathcal{E}$ for covering points in $\mathcal{P}$.

Definition 3 (Carmi et al. [5])**.** *Consider a unit disk $\hat{L} \in \mathcal{L}$ which intersects B. Given an interval cell $\texttt{icell}(i, j)$, if the set $\{d_i, d_j, \hat{L}\}$ covers all the points covered by the disks in the interval cell, then this new set is called an* assisting set *for the interval $[i, j]$. In the special case where $j = i+1$, $\{d_i, d_j\}$ forms the assisting set of the interval $[i, j]$. The assisting set $\{d_i, d_j, \hat{L}\}$ is said to contain a* left assisting pair*, which is simply the set $\{d_i, \hat{L}\}$. In special cases where an assisting set is composed of only one or two disks, the leftmost individual disk is considered a left assisting pair for these purposes. Finally, an* assisted cover *is simply the family of these left assisting pairs which together form a cover of the points in $\mathcal{P}$.*

Now we wish to compare the cardinality of $\mathcal{E}$ with that of the global minimum disk cover $\mathcal{G}$. Consider the upper and lower components of the solutions $\mathcal{E}$ and $\mathcal{G}$, i.e., $\mathcal{E}_{\mathcal{U}} = \mathcal{E} \cap \mathcal{U}$, $\mathcal{E}_{\mathcal{L}} = \mathcal{E} \cap \mathcal{L}$, $\mathcal{G}_{\mathcal{U}} = \mathcal{G} \cap \mathcal{U}$, and $\mathcal{G}_{\mathcal{L}} = \mathcal{G} \cap \mathcal{L}$. Note that $|\mathcal{G}| \leq |\mathcal{E}|$ since $\mathcal{G}$ is the global minimum. Similarly, since $\mathcal{E}$ is the minimum assisted cover based on $\mathcal{U}'$, it follows that $|\mathcal{E}| = |\mathcal{E}_{\mathcal{U}}| + |\mathcal{E}_{\mathcal{L}}| \leq |\mathtt{ac}(\mathcal{U}', \mathcal{G}_{\mathcal{L}})| + |\mathcal{G}_{\mathcal{L}}|$, where $\mathtt{ac}(\mathcal{U}', \mathcal{G}_{\mathcal{L}})$ is the smallest subset of $\mathcal{U}'$ that forms an assisted cover with $\mathcal{G}_{\mathcal{L}}$.

Now we will show that $2|\mathcal{G}_{\mathcal{U}}| \geq |\mathtt{ac}(\mathcal{U}', \mathcal{G}_{\mathcal{L}})|$. Given a disk d in $\mathcal{G}_{\mathcal{U}}$, there are two cases: either d lies above the lower boundary of $\mathtt{ac}(\mathcal{U}', \mathcal{G}_{\mathcal{L}})$, i.e., d is contained in the union of all the disks in $\mathtt{ac}(\mathcal{U}', \mathcal{G}_{\mathcal{L}})$, or d contains one or more arc segments of the lower boundary of $\mathtt{ac}(\mathcal{U}', \mathcal{G}_{\mathcal{L}})$. In the first case, Carmi et al. show that at most two disks in $\mathtt{ac}(\mathcal{U}', \mathcal{G}_{\mathcal{L}})$ suffice to cover d and, hence, for every such disk in the global optimum solution $\mathcal{G}$ there are at most two disks in $\mathtt{ac}(\mathcal{U}', \mathcal{G}_{\mathcal{L}})$. In the second case, let $\mathcal{V}$ denote the subset of disks that have lower boundary segments that are contained in d. The set of arc segments of the disks in $\mathcal{V}$ consists, from left to right, of a partially-covered arc segment of the lower boundary, zero or more fully-covered arc segments, and a partially-covered arc segment. Let $\mathcal{W}$ denote the disks whose arcs are partially covered together with d. $\mathcal{W}$ dominates $\mathcal{V}$ and hence there is at most one arc of the lower boundary fully contained in d; otherwise replacing $\mathcal{V}$ with $\mathcal{W}$ results in a cover of smaller cardinality, deriving a contradiction, since $\mathcal{V} \subset \mathcal{U}'$, and $\mathcal{U}'$ is the optimal LSDUDC solution. Recall that all disks in $\mathcal{V}$ and $\mathcal{U}$ are centered above l, and all points in $\mathcal{P}$ are below l. Furthermore, observe that the partially-covered arc disks must contain points not contained in the fully-covered disk; otherwise they can also be eliminated while reducing the cardinality of the cover. As those disks contain other points, each of the disks is partially covered by at least one other disk in $\mathcal{G}$. We arbitrarily associate each disk covered more than once to its leftmost disk in $\mathcal{G}$. Thus, of the (at most) three disks in $\mathcal{V}$, at most two are associated to d. In sum, in either case each disk in $\mathcal{G}_{\mathcal{U}}$ has at most two associated disks in $\mathtt{ac}(\mathcal{U}', \mathcal{G}_{\mathcal{L}})$ from which it follows that $2|\mathcal{G}_{\mathcal{U}}| \geq |\mathtt{ac}(\mathcal{U}', \mathcal{G}_{\mathcal{L}})|$. Hence,

$$2|\mathcal{G}| = 2(|\mathcal{G}_{\mathcal{U}}| + |\mathcal{G}_{\mathcal{L}}|) \geq 2|\mathcal{G}_{\mathcal{U}}| + |\mathcal{G}_{\mathcal{L}}| \geq |\mathtt{ac}(\mathcal{U}', \mathcal{G}_{\mathcal{L}})| + |\mathcal{G}_{\mathcal{L}}| \geq |\mathcal{E}_{\mathcal{U}}| + |\mathcal{E}_{\mathcal{L}}| = |\mathcal{E}|,$$

which gives the approximation factor of two as desired. Carmi et al. [5] prove that any disk can be used in up to eight applications of the assisted LSDUDC algorithm, for which they have a 4-approximation. These operations, followed by a 6-approximation for any remaining disks results in an $8 \times 4 + 6 = 38$-approximation for the general DUDC problem. As we have shown that our technique provides a 2-approximation for the assisted line separated discrete unit disk cover problem, we now have an approximation ratio of $8 \times 2 + 6 = 22$ for general discrete unit disk cover.

3.1 Algorithm Analysis

There are essentially two main components to the algorithm for solving DUDC by Carmi et al. [5]. First, they apply a grid of size $3/2 \times 3/2$ to the input data. Algorithm 1 supplemented by their assisting disk technique is run on all grid lines. Note that the number of relevant grid lines is $O(n)$. Our technique runs in $O(n(\log n + m))$, and the assisting disk operation is easily implementable in $O(mn)$, so the running time of the first component is dominated by our step.

The second major component to their technique is finding the 6-approximation for the DUDC of all disk centers and points contained in each of the $3/2 \times 3/2$ squares of the grid. Their technique is based on the application of a subset of nine properties depending on where the disk centers are located. First, they determine whether a solution exists using one or two centers by brute force, which is easily done in $O(m^2 n)$ time. The determination of which properties may be applied can be done in $O(m)$ time, and there are only two expensive steps that may be used in any of the procedures, each of which may only be used a constant number of times. First is the assisted LSDUDC technique, whose running time is $O(n(\log n + m))$, see Theorem 2. The second technique that may be required is to determine the optimal disk cover of a set of points using centers contained in one of the $1/2 \times 1/2$ squares, which can be solved in $O(m^2n^4)$ time using the technique presented in [16]. The center of each disk can only be contained in one square, and so this operation is never performed twice for any given disk. Therefore, the complete DUDC algorithm achieves worst-case performance when all of the disk centers in the plane are confined to a single $1/2 \times 1/2$ square, so that the $O(m^2n^4)$ operation is performed over the entire data set.

4 Conclusions

This paper presents a polynomial-time algorithm that returns an exact solution to the LSDUDC problem, as well as a proof of correctness of the approach. This algorithm for the line-separable problem allows us to improve the approximation algorithm of Carmi et al. [5], resulting in a 22-approximate solution to the general DUDC problem, which runs in $O(m^2n^4)$ time in the worst case.

Theorem 3. *Given sets $\mathcal{P}$ of m points and $\mathcal{D}$ of n disks in the plane, we can compute a 22-approximation of the DUDC problem in $O(m^2n^4)$ time in the worst case.*

Acknowledgments

The authors wish to thank Paz Carmi for sharing his insights and discussing details of his results on the discrete unit disk cover problem [5]. In addition, the authors acknowledge Sariel Har-Peled with whom a preliminary problem was discussed that inspired our examination of the disk cover problem. 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, and by the UNB Faculty of Computer Science.

References

[1] P. Agarwal and C. Procopiuc, Exact and approximation algorithms for clustering, *Alg.*, **33**, (2002) 201–226.

[2] P. Agarwal and M. Sharir, Efficient algorithms for geometric optimization. *ACM Comp. Surv.*, **30**, (1998) 412–458.

[3] C. Ambühl, T. Erlebach, M. Mihal'ák and M. Nunkesser, Constant-factor approximation for minimum-weight (connected) dominating sets in unit disk graphs, in *Proc. of Approx., Rand., and Comb. Opt. Alg. and Tech*, (2006), pp. 3–14.

[4] G. Călinescu, I. Măndoiu, P.J. Wan and A. Zelikovsky, Selecting forwarding neighbours in wireless ad hoc networks. *Mob. Net. and Appl.*, **9**(2), (2004) 101–111.

[5] P. Carmi, M. Katz and N. Lev-Tov, Covering points by unit disks of fixed location, in *Proc. Int'l Symp. on Alg. and Comp*, (2007), pp. 644–655.

[6] K. Clarkson and K. Varadarajan, Improved approximation algorithms for geometric set cover. *Disc. and Comp. Geom.*, **37**(1), (2007) 43–58.

[7] F. Claude, R. Dorrigiv, S. Durocher, R. Fraser, A. López-Ortiz and A. Salinger, Practical Discrete Unit Disk Cover Using an Exact Line-Separable Algorithm, in *Proc. Int'l Symp. Algs. and Comp.*, (2009).

[8] G. K. Das, S. Das and S. C. Nandy, Homogeneous 2-hop broadcast in 2D. *Comp. Geom.: Theory and Appl.*, **43**, (2010) 182–190.

[9] R. Fowler, M. Paterson and S. Tanimoto, Optimal packing and covering in the plane are NP-complete. *Inf. Proc. Lett.*, **12**(3), (1981) 133–137.

[10] G. Frederickson, Fast algorithms for shortest paths in planar graphs, with applications. *SIAM J. on Comp.*, **16**(6), (1987) 1004–1022.

[11] T. Gonzalez, Covering a set of points in multidimensional space, *Inf. Proc. Lett.*, **40**, (1991), 181–188.

[12] D. Hochbaum and W. Maass, Approximation schemes for covering and packing problems in image processing and VLSI. *J. ACM* **32**, (1985) 130–136.

[13] R. Hwang, R. Lee and R. Chang, The generalized searching over separators strategy to solve some NP-hard problems in subexponential time. *Alg.*, **9**, (1993) 398–423.

[14] D. Johnson, The NP-completeness column: An ongoing guide. *J. of Alg.* **3**(2), (1982) 182–195.

[15] I. Koutis and G. Miller, A linear work, $o(n^{1/6})$ time, parallel algorithm for solving planar Laplacians., in *Proc. Symp. Disc. Alg.*, (2007), pp. 1002–1011.

[16] N. Lev-Tov, Algorithms for Geometric Optimization Problems in Wireless Networks. Ph.D. thesis, Weizmann Institute of Science (2005).

[17] N. Mustafa and S. Ray, PTAS for geometric hitting set problems via local search, in *Proc. Symp. on Comp. Geom.*, (2009), pp. 17–22.

[18] N. Mustafa and S. Ray, Improved results on geometric hitting set problems. `www.mpi-inf.mpg.de/~saurabh/Papers/Hitting-Sets.pdf` (2009).

[19] S. Narayanappa and P. Voytechovsky, An improved approximation factor for the unit disk covering problem, in *Proc. Can. Conf. on Comp. Geom.*, (2006).

[20] R. Raz and S. Safra, A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP, in *Proc. ACM Symp. on Theory of Comp.*, (1997), pp. 475-484.

[21] K. Supowit, Topics in Computational Geometry. Ph.D. thesis, University of Illinois at Urbana-Champaign (1981).