Minimum Dominating Set for a Point Set in $\mathbb{R}^2$

Ramesh K. Jallu∗ Prajwal R. Prasad † Gautam K. Das ∗

Abstract

In this article, we consider the problem of computing minimum dominating set for a given set $S$ of $n$ points in $\mathbb{R}^2$. Here the objective is to find a minimum cardinality subset $S'$ of $S$ such that the union of the unit radius disks centered at the points in $S'$ covers all the points in $S$. We first propose a simple 4-factor and 3-factor approximation algorithms in $O(n^6 \log n)$ and $O(n^{11} \log n)$ time respectively improving time complexities by a factor of $O(n^2)$ and $O(n^4)$ respectively over the best known result available in the literature [M. De, G.K. Das, P. Carmi and S.C. Nandy, Approximation algorithms for a variant of discrete piercing set problem for unit disk, Int. J. of Comp. Geom. and Appl., to appear]. Finally, we propose a very important shifting lemma, which is of independent interest and using this lemma we propose a $\frac{5}{2}$-factor approximation algorithm and a PTAS for the minimum dominating set problem.

Keywords: minimum dominating set, unit disk graph, approximation algorithm.

1 Introduction

A minimum dominating set $S'$ for a set $S$ of $n$ points in $\mathbb{R}^2$ is defined as follows: (i) $S' \subseteq S$ (ii) each point $s \in S$ is covered by at least one unit radius disk centered at a point in $S'$, and (iii) size of $S'$ is minimum. The minimum dominating set (MDS) problem for a point set $S$ of size $n$ in $\mathbb{R}^2$ involves finding a minimum dominating set $S'$ for the set $S$. We call this problem as a geometric version of MDS problem. The MDS problem for a point set can be modeled as an MDS problem in unit disk graph (UDG) as follows: A unit disk graph $G = (V,E)$ for a set $U$ of $n$ unit diameter disks in $\mathbb{R}^2$ is the intersection graph of the family of disks in $U$ i.e., the vertex set $V$ corresponds to the set $U$ and two vertices are connected by an edge if the corresponding disks have common intersection. The minimum dominating set for the graph $G$ is a minimum size subset $V'$ of $V$ such that for each of the vertex $v \in V$ is either in $V'$ or adjacent to a node in $V'$ in $G$. Several people have done research on MDS problem because of its wide applications such as wireless networking, facility location problem, to name a few. Our interest in this problem arose from the following reason: suppose in a city we have a set $S$ of $n$ important locations (houses, etc.); the objective is to provide some emergency services (ambulance, fire station, etc.) to each of the locations in $S$ so that each location is within a predefined distance of at least one service center. Note that positions of the emergency service centers are from the predefined set $S$ of locations only.

∗Indian Institute of Technology Guwahati, India
†National Institute of Technology Karnataka, India
1.1 Related Work

The MDS problem can be viewed as a general set cover problem, but it is an NP-hard problem \cite{16, 20} and not approximable within $c \log n$ for some constant $c$ unless P = NP \cite{26}. Therefore $O(\log n)$-factor approximation algorithm is possible for MDS problem by applying the algorithm for general set cover problem \cite{4}. Some exciting results for the geometric version of MDS problem are available in the literature.

In the discrete unit disk cover (DUDC) problem, two sets $P$ and $Q$ of points in $\mathbb{R}^2$ are given, the objective is to choose minimum number of unit disks $D'$ centered at the points in $Q$ such that the union of the disks in $D'$ covers all the points in $P$. Johnson \cite{20} proved that the DUDC problem is NP-hard. Mustafa and Ray in 2010 \cite{22} proposed a $(1 + \delta)$-approximation algorithm for $0 < \delta \leq 2$ (PTAS) for the DUDC problem using $\varepsilon$-net based local improvement approach. The fastest algorithm is obtained by setting $\delta = 2$ for a 3-factor approximation algorithm, which runs in $O(m^{65}n)$ time, where $m$ and $n$ are number of unit radius disks and number of points respectively \cite{11}. The high complexity of the PTAS leads to further research on constant factor approximation algorithms for the DUDC problem. A series of constant factor approximation algorithms for DUDC problem are available in the literature:

- 108-approximation algorithm \cite{C˘ alinescu et al., 2004} \cite{5}
- 72-approximation algorithm \cite{Narayanappa and Voytechovsky, 2006} \cite{24}
- 38-approximation algorithm in $O(m^2n^4)$ time \cite{Carmi et al., 2007} \cite{6}
- 22-approximation algorithm in $O(m^2n^4)$ time \cite{Claude et al., 2010} \cite{9}
- 18-approximation algorithm in $O(mn + n \log n + m \log m)$ time \cite{Das et al., 2012} \cite{11}
- 15-approximation algorithm in $O(m^6n)$ time \cite{Fraser and López-Ortiz, 2012} \cite{13}
- $(9 + \varepsilon)$-approximation algorithm in $O(m^{3(1+\frac{\varepsilon}{2})}n \log n)$ time \cite{Acharyya et al., 2013} \cite{1}

The DUDC problem is a geometric version of MDS problem for $P = Q$. Therefore all results for the DUDC problem are applicable to MDS problem.

The geometric version of MDS problem is known to be NP-hard \cite{8}. Nieberg and Hurink \cite{23} proposed $(1 + \varepsilon)$-factor approximation algorithm for $0 < \varepsilon \leq 1$. The fastest algorithm is obtained by setting $\varepsilon = 1$ for a 2-approximation result, which runs in $O(n^{81})$ time \cite{10}, which is not practical even for $n = 2$. Another PTAS for dominating set of arbitrary size disk graph is available in the literature proposed by Gibson and Pirwani \cite{17}. The running time of this PTAS is $n^{O\left(\frac{1}{\varepsilon^2}\right)}$.

Marathe et al. \cite{21} proposed a 5-factor approximation algorithm for the MDS problem. Ambühl et al. \cite{2} proposed 72-factor approximation algorithm for weighted dominating set (WDS) problem. In the WDS problem, each node has a positive weight and the objective is to find the minimum weight dominating set of the nodes in the graph. Huang et al. \cite{19}, Dai and Yu \cite{12}, and Zou et al. \cite{27} improved the approximation factor for WDS problem to $6 + \varepsilon, 5 + \varepsilon$, and $4 + \varepsilon$ respectively. First, they proposed $\gamma$-factor ($\gamma = 6, 5, 4$ in \cite{19}, \cite{12}, and \cite{27} respectively) approximation algorithm for
a subproblem and using the result of their corresponding sub-problems they proposed \((\gamma + \epsilon)\)-factor approximation algorithms. The time complexity of their algorithms are \(O(\alpha(n) \times \beta(n))\), where \(O(\alpha(n))\) is the time complexity of the algorithm for the sub-problem and \(O(\beta(n)) = O(n^{4(\frac{\alpha}{2})^2})\) is the number of times the sub-problem needs to be invoked to solve the original problem. The \((\gamma + 1)\)-factor approximation algorithm can be obtained by setting \(\epsilon = 1\), but the time complexity becomes a very high degree polynomial function in \(n\). Carmi et al. [7] proposed a 5-factor approximation algorithm of the MDS problem for arbitrary size disk graph. Fonseca et al. [14] proposed a \(\frac{11}{5}\)-factor approximation algorithm for the MDS problem in UDG which can be achieved in \(O(n + m)\) time, when the input is a graph with \(n\) vertices and \(m\) edges, and in \(O(n \log n)\) time, in the geometric version of the problem. The same set of authors also proposed a \(\frac{10}{9}\)-factor approximation algorithm for the MDS problem in UDG which runs in \(O(n^2m)\) time [15]. Recently, De et al. [10] considered the geometric version of MDS problem and proposed 12-factor, 4-factor, and 3-factor approximation algorithms with running time \(O(n \log n)\), \(O(n^8 \log n)\), and \(O(n^{15} \log n)\) respectively. They also proposed a PTAS with high degree polynomial running time.

1.2 Our Contribution

In this paper, we consider the geometric version of MDS problem and propose a series of constant factor approximation algorithms. We first propose 4-factor and 3-factor approximation algorithms with running time \(O(n^6 \log n)\) and \(O(n^{11} \log n)\) respectively improving the time complexities by a factor of \(O(n^2)\) and \(O(n^4)\) respectively over the best known result in the literature [10]. Finally, we propose a new shifting strategy lemma. Using our shifting strategy lemma we propose \(\frac{5}{2}\)-factor and \((1 + \frac{1}{k})^2\)-factor (i.e., PTAS) approximation algorithms for the MDS problem. The running time of proposed \(\frac{5}{2}\)-factor and \((1 + \frac{1}{k})^2\)-factor approximation algorithms are \(O(n^{20} \log n)\) and \(n^{O(k)}\) respectively. Though the time complexity of the proposed PTAS is same as the PTAS proposed by De et al. [10] in terms of \(O\) notation, but the constant involved in our PTAS is smaller than the same in [10].

2 4-Factor Approximation Algorithm for the MDS Problem

In this section, a set \(S\) of \(n\) points in \(\mathbb{R}^2\) is given inside a rectangular region \(\mathcal{R}\). The objective is to find an MDS for \(S\). Here we propose a simple 4-factor approximation algorithm. The running time of our algorithm is \(O(n^6 \log n)\), which is an improvement by a factor of \(O(n^2)\) over the best known existing result [10]. In order to obtain a 4-factor approximation algorithm, we consider a partition of \(\mathcal{R}\) into regular hexagons of side length \(\frac{1}{2}\) (see Figure 1(a)). We use \(cell\) to denote a regular hexagon of side length \(\frac{1}{2}\).

**Lemma 1** All points inside a single cell can be covered by an unit radius disk centered at any point inside that cell.

**Proof:** The lemma follows from the fact that the distance between any two points inside a regular hexagon of side length \(\frac{1}{2}\) is at most 1 (for demonstration see the Figure 1(b)).
Figure 1: (a) Regular hexagonal partition (b) single regular hexagon of side length $\frac{1}{2}$ contained in an unit radius disk, and (c) a septa-hexagon

**Definition 1** A septa-hexagon is a combination of 7 adjacent cells such that one cell is inscribed by six other cells as shown in Figure 1(c).

For a point set $U$, we use $\Delta(U)$ to denote the set of unit radius disks centered at the points in $U$. Let $U_1$ and $U_2$ be two point sets such that $U_1 \subseteq U_2$. We use $\chi(U_1, U_2)$ to denote the set of points such that $\chi(U_1, U_2) \subseteq U_2$ and an unit radius disk centered at any point in $\chi(U_1, U_2)$ covers at least one point of $U_1$.

2.1 Algorithm overview

Let us consider a septa-hexagon $C$. Recall that $C$ is a combination of 7 cells (regular hexagon of side length $\frac{1}{2}$). Let $S_1 = S \cap C$ and $S_2 = \chi(S_1, S)$. For the 4-factor approximation algorithm, we first find minimum size subset $S' \subseteq S_2$ such that $S_1 \subseteq \bigcup_{d \in \Delta(S')} d$. Call this problem as single septa-hexagon MDS problem. Using the optimum (minimum size) solution of single septa-hexagon MDS problem, we present our main 4-factor approximation algorithm. The Lemma 2 gives an important feature to design optimum algorithm for single septa-hexagon MDS problem.

**Lemma 2** If $OPT_C$ is a minimum cardinality subset of $S_2$ such that $S_1 \subseteq \bigcup_{d \in \Delta(OPT_C)} d$, then $|OPT_C| \leq 7$.

**Proof:** The septa-hexagon $C$ has at most 7 non-empty cells. From Lemma 1, we know that an unit radius disk centered at a point in a cell covers all points in that cell. Therefore one point from each of the non-empty cells is sufficient to cover all the points in $C$. Thus the Lemma follows. □

**Lemma 3** For a given set $S$ of $n$ points and a septa-hexagon $C$, the Algorithm computes an MDS for $S \cap C$ using the points of $S$ in $O(n^6 \log n)$ time.

**Proof:** The optimality of the Algorithm follows from the fact that Algorithm considers all possible set of sizes $0, 1, \ldots, 7$ (see Lemma 2) as its solution and reports minimum size solution. The line number 7 of the algorithm can be computed in $O(n \log n)$ time as follows: (i) computation of the set $S_1$ takes $O(n)$ time, (ii) computation of $S_2$ can be done in $O(n \log n)$ time using nearest
Algorithm 1: Algorithm $4$ Factor($S, C, n$)

1: **Input:** A set $S$ of $n$ points and a septa-hexagon $C$
2: **Output:** A set $S' (\subseteq S)$ such that $(S \cap C) \subseteq \bigcup_{d \in \Delta(S')} d$.
3: $S' \leftarrow \emptyset$
4: if $(S \cap C \neq \emptyset)$ then
5: Choose one arbitrary point from each non-empty cell of $C$ and add to $S'$.
6: $m \leftarrow |S'| /* m$ is at most 7 */
7: Let $S_1 = S \cap C$ and $S_2 = \chi(S_1, S)$.
8: for $(i = m - 1, m - 2, \ldots, 1)$ do
9: if $(i = 6)$ then
10: for (Each possible combination of 5 points $X = \{p_1, p_2, \ldots, p_5\}$ of $S_2$) do
11: Find $Y \subseteq S_1$ such that no point in $Y$ is covered by $\bigcup_{d \in \Delta(X)} d$.
12: Compute the farthest point Voronoi diagram of $Y$. \[3\]
13: Find a point $p$ (if any) from $S_2 \setminus X$ (using planar point location algorithm \[25\]) such that the farthest point in $Y$ from $p$ is less than or equal to 1. If such $p$ exists, then set $S' \leftarrow X \cup \{p\}$ and exit for loop.
14: end for
15: else
16: for (Each possible combination of $i$ points $X = \{p_1, p_2, \ldots, p_i\}$ of $S_2$) do
17: if $(S_1 \subseteq \bigcup_{d \in \Delta(X)} d)$ then
18: Set $S' \leftarrow X$ and exit from for loop
19: end if
20: end for
21: end if
22: end for
23: end if
24: Return $S'$

point Voronoi diagram of $S_1$ in $O(n \log n)$ time and for each point $p \in S$ apply planar point location algorithm to find the nearest point in $S_1$ in $O(\log n)$ time.

The running time of the else part in the line number 15 of the algorithm is at most $O(n^6)$ time. The worst case running time of the algorithm comes from line numbers 9-14. The complexity of line numbers 11-13 is $O(n \log n)$ time. Therefore the running time of the line numbers 9-14 is $O(n^6 \log n)$ time. Thus the overall worst case running time of the proposed Algorithm [1] is $O(n^6 \log n)$. \[5\]

Let us consider a septa-hexagonal partition of $\mathbb{R}$ such that no point of $S$ is on the boundary of any septa-hexagon and a 4 coloring scheme of it (see Figure 2). Consider an unicolor septa-hexagon of color A (say). Its adjacent septa-hexagons are assigned colors B, C and D (say) such that opposite septa-hexagons are assigned the same color (see Figure 2).

**Lemma 4** If $C'$ and $C''$ are two same colored septa-hexagons, then $(C' \cup C'') \cap S \cap d = \emptyset$ for any unit radius disk $d$. 

\[5\]
Proof: According to the 4-coloring scheme, size of the septa-hexagons, and no point of $S$ is on the boundary of $C'$ and $C''$ the minimum distance between two points $s_1 \in C' \cap S$ and $s_2 \in C'' \cap S$ is greater than 2 (see Figure 2). Thus the lemma follows.

Theorem 1 The 4-coloring scheme gives a 4-factor approximation algorithm for the MDS problem in $O(n^6 \log n)$ time, where $n$ is the input size.

Proof: Let $N_1, N_2, N_3,$ and $N_4$ be the sets of septa-hexagons of colors $A, B, C,$ and $D$ respectively. Let $S'_i = S \cap \bigcup_{C \in N_i} C$ and $S'_2 = \chi(S'_i, S)$ for $1 \leq i \leq 4$. By Lemma 4, the pair $(S'_1, S'_2)$ can be partitioned into $|N_i|$ pairs $(S'_{1j}, S'_{2j})$ such that for each pair Algorithm 1 is applicable for solving the covering problem optimally to cover $S'_i$ using $S'_2$, where $1 \leq j \leq |N_i|$. Let $N'_i$ be the optimum solution for the set $S'_i$ ($1 \leq i \leq 4$) using the Algorithm 1. If $OPT$ is the optimum solution for the set $S$, then $|N'_i| \leq |OPT|$. Therefore $\sum_{i=1}^{4} |N'_i| \leq 4 \times |OPT|$. Thus the approximation factor of the algorithm follows.

The time complexity result of the theorem follows from Lemma 5 and the fact that each point in $S$ can participate in the Algorithm 1 at most constant number of times.

3 3-Factor Approximation Algorithm for the MDS Problem

Given a set $S$ of $n$ points in a rectangular region $R$, we wish to find an MDS for $S$. Here we present a 3-factor approximation algorithm in $O(n^{11} \log n)$ time for the MDS problem, which is an improvement by a factor of $O(n^4)$ over the best known result available in the literature [10].

Definition 2 A super-cell is a combination of 15 regular hexagons of side length $\frac{1}{2}$ arranged in three consecutive rows as shown in Figure 3.
3.1 Algorithm overview

Let us consider a super-cell $\mathcal{D}$. Let $S_1 = S \cap \mathcal{D}$ and $S_2 = \chi(S_1, S)$. In order to obtain 3-factor approximation algorithm for the MDS problem, we first find a minimum size subset $S' \subseteq S_2$ such that $S_1 \subseteq \bigcup_{d \in \Delta(S')} d$. Call this problem as a single super-cell MDS problem. Using the optimum solution of single super-cell MDS problem, we present our main 3-factor approximation algorithm.

Lemma 5 If $OPT_D$ is the minimum cardinality subset of $S_2$ such that $S_1 \subseteq \bigcup_{d \in \Delta(OPT)} d$, then $|OPT_D| \leq 15$.

Proof: The lemma follows from the Lemma 1 and the fact that the super-cell $\mathcal{D}$ has at most 15 non-empty cells.

We decompose a super-cell $\mathcal{D}$ into 3 regions namely $G_1^D$, $G_2^D$, and $G_3^D$ (see Figure 4 where $G_1^D$, $G_2^D$, and $G_3^D$ correspond to unshaded, light shaded, and dark shaded regions respectively).

Lemma 6 For any unit radius disk $d$ and a super-cell $\mathcal{D}$, $(G_1^D \cup G_3^D) \cap d = \emptyset$.

Proof: The lemma follows from the fact that if $s$ and $t$ are two arbitrary points of $G_1^D$ and $G_3^D$ respectively, then the Euclidean distance between $s$ and $t$ is greater than 2.
Let $S_1 = S \cap D$ and $S_2 = \chi(S_1, S)$, where $D$ is a super-cell. Our objective is to find a minimum cardinality set $S' \subseteq S_2$ such that $S_1 \subseteq \bigcup_{d \in \Delta(S')} d$.

Let $S'_1 = S_1 \cap G_D^1$, $S'_2 = S_1 \cap G_D^2$, and $S'_3 = S_1 \cap G_D^3$. A point on a boundary can be assigned to any set associated with that boundary. Let $S'_2 = \chi(S'_1, S_2)$, $S'_3 = \chi(S'_1, S_2)$, and $S'_3 = \chi(S'_1, S_2)$. The Lemma 6 says that $S'_1 \cap S'_3 = \emptyset$.

Algorithm 2: Algorithm 3_Factor($S, D, n$)

1: **Input:** A set $S$ of $n$ points and a super-cell $D$
2: **Output:** A set $S' \subseteq S$ such that $(S \cap D) \subseteq \bigcup_{d \in \Delta(S')} d$
3: $S' \leftarrow S$
4: Find the sets $S'_1, S'_2, S'_3, S'_2, S'_2,$ and $S'_3$ as defined above.
5: **for** (Each possible combination $X = \{p_1, p_2, \ldots, p_j\}$ of $j (0 \leq j \leq 9)$ points in $S'_2$) **do**
6:  **if** ($S'_2 \subseteq \bigcup_{d \in \Delta(X)} d$) **then**
7:   Let $U$ and $V$ be the subsets of $S'_1$ and $S'_3$ respectively such that no point in $U \cup V$ is covered by $\bigcup_{d \in \Delta(X)} d$.
8:  Let $Z$ be the minimum size subset of $S'_1$ such that $U \subseteq \bigcup_{d \in \Delta(Y)} d$.
9:  Let $Z$ be the minimum size subset of $S'_2$ such that $V \subseteq \bigcup_{d \in \Delta(Z)} d$.
10: **if** ($|S'_2| > |X| + |Y| + |Z|$) **then**
11:   Set $S' \leftarrow X \cup Y \cup Z$
12: **end if**
13: **end if**
14: **end for**
15: **Return** $S'$

Lemma 7 For a given set $S$ of $n$ points and a super-cell $D$, the Algorithm 2 computes an MDS for $S \cap D$ using the points of $S$ in $O(n^{11} \log n)$ time.

**Proof:** In the case of selecting 3 points in $S'_2$ in line number 8 of the algorithm, we can choose one point from each of the non-empty cells of $G_D^1$. Therefore, the worst case of line number 8 appears for the case of choosing all possible combinations of two points in $S'_2$. This can be done in $O(n^2 \log n)$ using the technique of the Algorithm 1 (line numbers 12-13). Similar analysis is applicable to line number 9. Line numbers 6-7 and 10-12 can be implemented in $O(n)$ time.

The worst case running time of the algorithm depends on the for loop in the line number 5. In this for loop, we are choosing all possible 9 points from a set of $n$ points in worst case. Therefore the time complexity of the Algorithm 2 is $O(n^{11} \log n)$.

The optimality of the algorithm follows from the Lemma 6 and fact that Algorithm 2 considers all possible combinations as its solution and returns minimum size solution.

Note that Algorithm 2 checks if condition in line number 6 because of the definition of $S'_2, S'_2$, and $S'_3$.

Let us consider a super-cell partition of $\mathcal{R}$ such that no point of $S$ lies on the boundary and a 3-coloring scheme (see Figure 5). Consider an unicolor super-cell which has been assigned color A (say). Its adjacent super-cells are assigned colors B, and C alternately (see Figure 5).
Lemma 8 If $D'$ and $D''$ are two same colored super-cells, then $(D' \cup D'') \cap S \cap d = \emptyset$ for any unit radius disk $d$.

Proof: The lemma follows from the following facts: (i) size of the super-cells $D'$ and $D''$ (ii) no point of $S$ on the boundary of $D'$ and $D''$, and (iii) the 3-coloring scheme.

Theorem 2 The 3-coloring scheme gives a 3-factor approximation algorithm for the MDS problem in $O(n^{11} \log n)$ time, where $n$ is the input size.

Proof: The follows by the similar argument of Theorem [1].

4 Shifting Strategy and its Application to the MDS Problem

In this section, we first propose a shifting strategy for the MDS problem, which is a generalization of the shifting strategy proposed by Hochbaum and Maass [18]. Next we propose $\frac{2}{3}$-factor approximation algorithm and a PTAS algorithm for MDS problem using our shifting strategy.

4.1 The Shifting Strategy

Our shifting strategy is very similar to the shifting strategy in [18]. We include a brief discussion here for completeness. Let a set $S$ of $n$ points be distributed inside an axis aligned rectangular region $R$. Our objective is to find an MDS for $S$.

Definition 3 A monotone chain $c$ with respect to line $L$ is a chain of line segments such that any line perpendicular to $L$ intersect it only once. We define the distance between two monotone chains $c'$ and $c''$ as the minimum Euclidean distance between any two points $p'$ and $p''$ on the chains $c'$ and $c''$ respectively. A monotone strip denoted by $M_s$ and is defined by the area bounded by any two monotone chains $c'$ and $c''$ such that the area is left closed and right open.

Consider a set $c_1, c_2, \ldots, c_r$ of $r$ monotone chains with respect to the line parallel to $y$-axis from left to right dividing the region $R$ such that distance between each pair of monotone chains is at least
\( D(> 0) \), where \( c_l \) and \( c_r \) are the left and right boundary of \( R \) respectively (see Figure 6). Let \( \mathcal{A} \) be an \( \alpha \)-factor approximation algorithm, which provides a solution of any \( \ell \) consecutive monotone strips for the MDS problem.

![Figure 6: Demonstration of shifting strategy](image)

**Theorem 3** We can design an \( \alpha(1 + \frac{1}{\ell}) \)-factor approximation algorithm for finding an MDS for \( S \).

**Proof:** The algorithm is exactly same as the algorithm proposed by Hochbaum and Maass [18]. The approximation factor follows from exactly the same argument proved in the shifting lemma [18]. \( \square \)

### 4.2 \( \frac{5}{2} \)-Factor Approximation Algorithm for the MDS Problem

Here we propose a \( \frac{5}{2} \)-factor approximation algorithm for MDS problem for a given set \( S \) of \( n \) points in \( \mathbb{R}^2 \) using shifting strategy discussed in Subsection [4.1].

**Definition 4** A duper-cell is a combination of 30 cells (regular hexagon of side length \( \frac{1}{2} \)) as shown in Figure 7. A duper-cell \( \mathcal{E} \) generates four monotone chains with respect to vertical and horizontal lines along its boundary. See Figure 7 where \( uv, vw, wx, \) and \( xu \) are the monotone chains. We rename them as **left**, **bottom**, **right**, and **top** monotone chains.

The basic idea is as follows: first optimally solve the subproblem **duper-cell** i.e., find an MDS for the set \( S \cap \mathcal{E} \), where \( \mathcal{E} \) is a duper-cell and then apply shifting strategy in both horizontal and vertical directions separately. The Lemma [4] leads to restriction on the size of the MDS, which is at most 30. Therefore an easy optimum solution for MDS can be obtained in \( O(n^{30}) \) time. Here we propose a different technique for the MDS problem leading to lower time complexity as follows:

We divide the duper-cell \( \mathcal{E} \) into 2 groups unshaded region \( (U_R) \) and shaded region \( (S_R) \) as shown in Figure 7. Let \( \mu \) be the common boundary of the regions and two extended lines (see Figure 7).
Algorithm 3: \texttt{MDS\_for\_duper-cell}(S, E, n)

1: \textbf{Input:} A set $S$ of $n$ points and a duper-cell $E$.
2: \textbf{Output:} A set $S' (\subseteq S)$ for an MDS of $S \cap E$.
3: Find $Q_1$ and $Q_2$ as described above.
4: Let $S_L^1$ and $S_R^2$ be the set of points in $S \setminus (Q_1 \cup Q_2)$ such that each disk in $\Delta(S_L^1)$ and $\Delta(S_R^2)$
   covers at least one point in $S \cap U_R$ and $S \cap S_R$ respectively.
5: $S' \leftarrow \emptyset, X \leftarrow \emptyset$
6: \textbf{for} ($i = 0, 1, \ldots, 9$) \textbf{do}
7: \hspace{1em} choose all possible $i$ disks in $\Delta(Q_1)$ (resp. $\Delta(Q_2)$) and for each combination of $i$ disks find
   $S_L^1$ and $S_R^2$ such that $S_L^1 \subseteq (S \cap U_R)$ and uncovered by that $i$ disks, and $S_R^2 \subseteq (S \cap S_R)$ and
   uncovered by that $i$ disks.
8: \hspace{1em} Call Algorithm 2 for finding an MDS for the sets $S_L^1$ and $S_R^2$ separately.
9: \textbf{end for}
10: Return $S'$

Let $Q_1$ and $Q_2$ be two sets of points in the left (resp. right) of $\mu$ such that each disk in $\Delta(Q_1)$ and
$\Delta(Q_2)$ intersects $\mu$.

Lemma 9 An MDS for the set of points inside a duper-cell $E$ can be computed optimally in
$O(n^{20} \log n)$ time, where $n$ is the input size.

Proof: The time complexity of line number 8 of the Algorithm 3 is $O(n^{11} \log n)$ (see Lemma 7).
The line number 8 executes at most $O(n^9)$ time by the \textbf{for} loop in line number 6. Therefore the time complexity of the lemma follows.

In the \textbf{for} loop (line number 6 of the algorithm), we considered all possible $i$ ($0 \leq i \leq 9$) disks in $\Delta(Q_1)$ and $\Delta(Q_2)$ separately. Since the number of cells that can intersect with such $i$ disks is
at most 9, therefore the range of $i$ is correct. For each combination of $i$ disks, we considered all
possible combinations to solve the problem for $S_L^1$ and $S_R^2$ separately. Therefore the correctness of
the algorithm follows. \hfill \Box

Theorem 4 The shifting strategy discussed in Subsection 4.1 gives a $\frac{5}{2}$-factor approximation algo-

rithm, which runs in $O(n^{20} \log n)$ time for the MDS problem, where $n$ is the input size.

Figure 7: Demonstration of $\frac{5}{2}$-factor approximation algorithm.
Proof: The distance between the monotone chains left and right of $E$ is greater than 8, the distance between the monotone chains bottom and top is 2, and the diameter ($D$) of the disks is 2. Now, if we apply shifting strategy in horizontal and vertical directions separately, then we get $(1 + \frac{1}{4})(1 + \frac{1}{2})$-factor i.e. $\frac{5}{2}$-factor approximation algorithm in $O(n^{20} \log n)$ time (see Lemma 9) for the MDS problem. □

4.3 A PTAS for MDS Problem

In this section, we present a $(1 + \frac{1}{k})^2$-factor approximation algorithm in $n^{O(k)}$ time for a positive integer $k$. Suppose a set $S$ of $n$ points within a rectangular region $R$ is given. Consider a partition of $R$ into regular hexagonal cells of side length $\frac{1}{2}$. The idea of our algorithm is to solve the MDS problem optimally for the points inside regular hexagons (say $F$) such that the distance between left and right (resp. bottom and top) monotone chains is $2k$ (see Figure 8) and using our proposed shifting strategy carefully (see Subsection 4.1).

To solve the MDS problem in $S \cap F$ we further decompose $F$ into four parts using the monotone chains $L_1$ and $L_2$ as shown in Figure 8. The number of disks in the optimum solution intersecting the chain $L_1$ with centers left (resp. right) side of $L_1$ is at most $\lceil 3 \times 3 \times \frac{2k}{4} \rceil$ which is less than $10k$ and the number of disks in the optimum solution intersecting the chain $L_2$ with centers bottom (resp. top) side of $L_2$ is at most $\lceil 5 \times 3 \times \frac{2k}{4} \rceil$ which is less than $8k$. Next we apply recursive procedure to solve four independent sub-problems of size $k \times k$. If $T(n, 2k)$ is the running time of the recursive algorithm for the MDS problem for $S \cap F$, then using the technique of [10] we have the following recurrence relation: $T(n, 2k) = 4 \times T(n, k) \times n^{10k+8k}$, which leads to the following theorem.

**Theorem 5** For a given set $S$ of $n$ points in $\mathbb{R}^2$, the proposed algorithm produces an MDS of $S$ in $n^{O(k)}$ time, whose size is at most $(1 + \frac{1}{k})^2 \times |OPT|$, where $k$ is a positive integer and $OPT$ is the optimum solution.
5 Conclusion

In this paper, we proposed a series of constant factor approximation algorithms for the MDS problem for a given set $S$ of $n$ points. Here we used hexagonal partition very carefully. We first presented a simple 4-factor and 3-factor approximation algorithms in $O(n^6 \log n)$ and $O(n^{11} \log n)$ time respectively, which improved the time complexities of best known result by a factor of $O(n^2)$ and $O(n^4)$ respectively [10]. Finally, we proposed a very important shifting lemma and using this lemma we presented a $\frac{5}{2}$-factor approximation algorithm and a PTAS for the MDS problem. Though the complexity of the proposed PTAS is same as that of the PTAS proposed by De et al. [10] in terms of $O$ notation, but the constant involved in our PTAS is smaller than the same in [10].

References

[1] R. Acharyya, M. Basappa and G.K. Das. Unit disk cover problem in 2D. In ICCSA, LNCS - 7972, pp. 73–85, 2013.
[2] C. Ambühl, T. Erlebach, M. Mihaláč and M. Nunkesser. Constant-factor approximation for minimum-weight (connected) dominating sets in unit disk graphs. In APPROX-RANDOM, pp. 3–14, 2006.
[3] M. de Berg, O. Cheong, M. van kreveld and M. H. Overmars. Computational Geometry: Algorithms and Applications (3. ed.). Springer-Verlag, 2008.
[4] V. Chvátal. A greedy heuristic for the set-covering problem. Mathematics of Operations Research, 4(3):233–235, 1979.
[5] G. Călinescu, I. I. Mandoiu, P. J. Wan and A. Zelikovsky. Selecting forwarding neighbors in wireless ad hoc networks. Mobile Network Applications, 9(2):101–111, 2004.
[6] P. Carmi, M. J. Katz and N. Lev-Tov. Covering points by unit disks of fixed location. In ISAAC, pp. 644–655, 2007.
[7] P. Carmi, M. J. Katz and N. Lev-Tov. Polynomial-time approximation schemes for piercing and covering with applications in wireless networks. Comput. Geom., 39(3):209–218, 2008.
[8] B. N. Clark, C. J. Colbourn and D. S. Johnson. Unit disk graphs. Discrete Mathematics, 86(1-3):165–177, 1990.
[9] F. Claude, G. K. Das, R. Dorrigiv, S. Durocher, R. Fraser, A. López-Ortiz, B. G. Nickerson and A. Salinger. An improved line-separable algorithm for discrete unit disk cover. Discrete Math. Alg. and Appl., 2(1):77–88, 2010.
[10] M. De, G. K. Das, P. Carmi and S.C. Nandy. Approximation algorithms for a variant of discrete piercing set problem for unit disks. Int. J. of Comput. Geom. and Appl., 2013 (to appear).
[11] G. K. Das, R. Fraser, A. López-Ortiz and B. G. Nickerson. On the discrete unit disk cover problem. Int. J. of Comput. Geom. and Appl., 22:407–419, 2012.

[12] D. Dai and C. Yu. A $5+\epsilon$-approximation algorithm for minimum weighted dominating set in unit disk graph. Theor. Comput. Sci., 410(8-10):756–765, 2009.

[13] R. Fraser and A. López-Ortiz. The within-strip discrete unit disk cover problem. In CCCG, pp. 53–58, 2012.

[14] G.D. da Fonseca, C. M. H. de Figueiredo, V. G. P. de Sá and R. Machado. Linear time approximation for dominating sets and independent dominating sets in unit disk graphs. In WAOA 2012, LNCS 7846, pp. 82–92, 2013.

[15] G. D. da Fonseca, C. M. H. de Figueiredo, V. G. P. de Sá and R. Machado. Linear-time sub-5 approximation for dominating sets in unit disk graphs. preprint, arXiv:1204.3488, 2012.

[16] M. R. Garey and D. S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, 1979.

[17] M. Gibson and I. A. Pirwani. Algorithms for dominating set in disk graphs: breaking the log$n$ barrier. In ESA, pp. 243–254, 2010.

[18] D. S. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and VLSI. J. ACM, 32(1):130–136, 1985.

[19] Y. Huang, X. Gao, Z. Zhang and W. Wu. A better constant-factor approximation for weighted dominating set in unit disk graph. Journal of Combinatorial Optimization, 18:179–194, 2008.

[20] D. S. Johnson. The np-completeness column: an ongoing guide. J. Algorithms, 3(2):182–195, 1982.

[21] M. V. Marathe, H. Breu, H. B. H. III, S. S. Ravi and D. J. Rosenkrantz. Simple heuristics for unit disk graphs. Networks, 25(2):59–68, 1995.

[22] N. H. Mustafa and S. Ray. Improved results on geometric hitting set problems. Discrete & Computational Geometry, 44(4):883–895, 2010.

[23] T. Nieberg and J. Hurink. A PTAS for the minimum dominating set problem in unit disk graphs. In WAOA, pages 296–306, 2005.

[24] S. Narayanappa and P. Vojtechovský. An improved approximation factor for the unit disk covering problem. In CCCG, pp. 15–18, 2006.

[25] F. P. Preparata and M. I. Shamos. Computational Geometry: An Introduction. Springer-Verlag, 2009.

[26] R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In STOC, pp. 475–484, 1997.
[27] F. Zou, Y. Wang, X. Xu, X. Li, H. Du, P. J. Wan and W. Wu. New approximations for minimum-weighted dominating sets and minimum-weighted connected dominating sets on unit disk graphs. *Theor. Comput. Sci.*, 412(3):198–208, 2011.