Approximation Algorithms for the Discrete Piercing Set Problem for Unit Disks

Minati De*† Gautam K. Das‡ Subhas C. Nandy*

Abstract

In this note, we shall consider constant factor approximation algorithms for a variation of the discrete piercing set problem for unit disks. Here a set of points P is given; the objective is to choose minimum number of points in P to pierce all the disks of unit radius centered at the points in P. We first propose a very simple algorithm that produces a 14-factor approximation result in $O(n \log n)$ time. Next, we improve the approximation factor to 4 and then to 3. Both algorithms run in polynomial time.

1 Introduction

The piercing set of a set of objects S in $\mathbb{R}^2$ is a set of points Q such that each object in S contains at least one point in Q. Here the problem is, given the set S, compute a piercing set of minimum size. Let us consider the intersection graph $G = (V, E)$ of the objects in S. Its nodes V correspond to the members in S, and an edge $e = (u, v) \in E$, for a pair of vertices $u, v \in V$ implies that the two objects corresponding to the nodes u and v intersect. A clique C in the graph G implies that each pair of objects corresponding to the nodes in C are intersecting. But, it does not imply that all of them have a non-empty common intersection region. In other words, a clique C in G does not imply that the objects corresponding to the members in C can be pierced by a single point. However, if S consists of a set of axis-parallel rectangles, then the minimum piercing set corresponds to the minimum clique cover [1] of the intersection graph of the members in S.

The minimum clique cover problem for a set of axis-parallel unit squares in $\mathbb{R}^2$ is known to be NP-hard [17]. Hochbaum and Maass [16] proposed a PTAS for the minimum clique cover problem for a set of axis-parallel unit squares with time complexity $n^{O(1/\epsilon^2)}$. The time complexity was later improved to $n^{O(1/\epsilon)}$ by Feder and Greene [13], and by Gonzalez [14]. Chan [5] proposed a PTAS for squares of arbitrary size with time complexity $n^{O(1/\epsilon^2)}$. In fact, this algorithm works for any collection of fat objects. Chan and Mahmood [6] considered the problem for a set of axis-parallel rectangles of fixed height (but of arbitrary width), and proposed a PTAS with $n^{O(1/\epsilon^2)}$ time complexity.

The minimum clique cover problem for unit disk graph also has a long history. The problem is known to be NP-hard [9], and a 3-factor approximation algorithm is easy to obtain [19]. Recently, Dumitrescu and Pach [12] proposed an $O(n^2)$ time randomized algorithm for the minimum clique cover problem with approximation ratio 2.16. They also proposed a polynomial time approximation scheme (PTAS) for this problem that runs in $O(n^{1/\epsilon^2})$ time. It is an improvement on a previous PTAS with $O(n^{1/\epsilon^4})$ running time [22].

Since, the disks do not satisfy the Helly's property[2], the minimum piercing set problem for unit disks is different from the minimum clique cover problem for unit disk graph. The minimum piercing set problem for disks has a lot of applications in wireless communication where the objective is to place the base stations to cover a set of radio terminals (sensors) distributed in a region. The minimum piercing set problem for unit disks is also NP-hard [3, 12]. Carmi et. al [3] proposed an approximation algorithm for this problem where the approximation factor is 38. In particular, if the points are distributed below a straight line L, and the base stations (of same range) are allowed to be installed on or above L only then a 4-factor approximation algorithm can be obtained provided all the points lie within an unit distance from at least one base station.

In the discrete version of the minimum piercing set problem for unit disks, two sets of points P and Q are given. The unit disks are centered at the points of P, and the piercing points need to be chosen from Q. The objective is to choose minimum number of points from Q to pierce all the disks in P. The problem is known to be NP-hard [18]. The first constant factor approximation result on this problem is proposed by Calinescu et al. [2]. It uses linear programming relaxation method to produce an 108-factor approximation result. The approximation re-

*Indian Statistical Institute, Kolkata, India

†Visiting Carleton University, Canada, during April 1, 2011 - August 27, 2011. minati.isi@gmail.com

‡Indian Institute of Technology Guwahati, India

[1] The minimum clique cover problem for a graph $G = (V, E)$ is partitioning the vertex set V into minimum number of subsets such that the subgraph induced by each subset is a clique.

[2] A set of object has the Helly property if each intersecting family has a non-empty intersection.

sult is then improved to 72 in [21], 38 in [3], and 22 in [8]. Finally, Das et al. [10] proposed an 18-factor approximation algorithm that runs in $O(n \log n + m \log m + mn)$ time, where $|P| = n$ and $|Q| = m$.

Another variation of the discrete piercing set problem for unit disks assumes $Q = P$. In other words, the unit disks corresponding to the points in P need to be pierced by choosing a minimum number points from P itself. In the literature, the problem is referred to as the *minimum dominating set problem for the unit disk graph* (or *MDS problem* in short). Here, an undirected graph is constructed with nodes corresponding to the points in P. Between a pair of nodes there is an edge if the distance between the two points is less than or equal to their common radius. A vertex in the graph dominates itself and all its neighbors. The objective is to choose minimum number of vertices to dominate all the vertices in the graph.

The problem is known to be NP-hard [7]. Ambuhl et al. [1] first proposed an approximation algorithm for this problem. They considered the weighted version of the problem where each node is attached with a positive weight. The objective is to find the minimum weight dominating set of the nodes in the graph. The approximation factor of their proposed algorithm is 72. Huang et al. [15] improved the approximation factor of the same problem to $6 + \epsilon$. Dai and Yu [11] further improved the approximation factor to $5 + \epsilon$. Though they have not analyzed the time complexity of their proposed algorithm, their algorithm needs $O(\frac{n^9}{\epsilon^2})$ time. Recently, Zou et al. [23] proposed a polynomial time $4 + \epsilon$ factor approximation algorithm. Nieberg and Hurink [20] proposed an $O(n^c)$ time PTAS for computing the minimum dominating set for unit disk graphs, where $c = (2r + 1)^2$, and r is an integer satisfying $(2r + 1)^2 < (1 + \epsilon)^{r/2}$. It accepts any undirected graph as input, and returns a $(1 + \epsilon)$ factor approximation solution for the dominating set problem, or a certificate indicating that the input graph is not a unit disk graph. For a 2-factor approximation result, the worst-case running time is obtained by setting $\epsilon = 1$; in that case, r will be equal to 22. Thus, the running time is $O(n^{(2r+1)^2}) = O(n^{(2\times 22+1)^2}) = O(n^{2025})$. Even for a 3-factor approximation result, the worst case time complexity (by putting $\epsilon = 2$) becomes $O(n^{625})$. Thus, this algorithm is not at all tractable from a practical point of view. Our present work is directed towards finding a tractable algorithm with a guaranteed constant factor approximation result. For the unweighted version of the discrete piercing set problem, the best known result is a 5-factor approximation algorithm proposed in [4], and it works for disks of arbitrary radii. This result is then used for the h-piercing problem, where the objective is to choose minimum number of points in P to pierce each disk by at least h points. The proposed approximation factor was $5(2^h - 1)$.

We propose three methods that use almost similar type approach for the discrete piercing problem with $Q = P$. The first one produces a 14-factor approximation result in $O(n \log n)$ time. The second one produces a 4-factor solution in $O(n^9)$ time, and the last one produces a 3-factor solution in $O(n^{18})$ time. Recall that the running time of the existing algorithm for producing a 3-factor approximation solution is $O(n^{625})$ [20]. Thus, our algorithm is a substantial improvement over the existing results in the literature. We can use this result to improve the approximation factor for the h-piercing problem [4] of constant radius disks to $3(2^h - 1)$ from $5(2^h - 1)$.

2 Approximation Algorithms

We are given a set of points P, where each point corresponds to a unit disk centered at that point. The objective is to choose a subset $P' \subseteq P$ of minimum cardinality such that the disk corresponding to each point in P contains at least one member of P'.

2.1 A simple 14-factor approximation algorithm

Consider a partitioning of the plane into a grid whose each cell is of size $\frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}}$. Since the maximum distance between any two points in a grid cell is less than or equal to 1, we can pierce all the disks centered at points of P in a particular cell by choosing any one member $p \in P$ lying in that cell. In other words, if we draw a disk of unit radius, and centered at p, it covers all the points lying inside that cell. Note that, it may cover point(s) in the other cell(s). But, we show that a disk centered at a point $p \in P$ inside a grid cell may cover (some or all) points in at most 14 other grid cells.

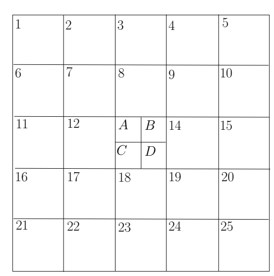

Figure 1: Discrete piercing set for unit disks

Consider the 5×5 grid structure as shown in Figure 1. The length of each side of a cell is $\frac{1}{\sqrt{2}}$. The cells are numbered as $1, 2, \ldots, 25$. The cell 13 is split into four parts, namely A, B, C and D. Observe that, a disk of radius 1 centered at any point in sub-cell A may cover (some or all) points in only 15 cells, numbered 2, 3, 4, 6, 7, 8, 9, 11, 12, 13, 14, 16, 17, 18 and 19. The same fact can be observed for the sub-cells B, C and D. We can further tighten the observation as stated below.

Observation 1 *A single unit disk centered at a point inside a cell can not cover points in more than 14 cells simultaneously.*

Proof. First we prove that a single unit disk centered at a point p in the cell A can not cover points in cell number 4 and 16 simultaneously (see Figure 1).

Let u and v be the bottom-left and top-right corners of the cells 4 and 16 respectively. Thus, $dist(u, v) = 2$, where $dist(.,.)$ denotes the Euclidean distance between a pair of points. Let p be a point properly inside cell A. Therefore, $dist(u, p) + dist(p, v) > 2$. This implies that at least one of $dist(u, p)$ and $dist(p, v)$ is greater than 1. Therefore, the point p can not cover a point inside cell 4 and a point inside cell 16 simultaneously. Thus, a single unit disk at a point $p \in A$ can cover (some or all) points in cells numbered 2, 3, 4, 6, 7, 8, 9, 11, 12, 13, 14, 16, 17, 18 and 19, but it can not cover a point in cell 4 and a point in cell 16 simultaneously.

Similarly, it can be shown that

- a single unit disk centered at a point $p \in B$ can cover (some or all) points in cells numbered 2, 3, 4, 7, 8, 9, 10, 12, 13, 14, 15, 17, 18, 19 and 20, but it can not cover a point in cell 2 and a point in cell 20 simultaneously.
- a single unit disk centered at a point $p \in C$ can cover (some or all) points in cells numbered 6, 7, 8, 9, 11, 12, 13, 14, 16, 17, 18, 19, 22, 23 and 24, but it can not cover a point in cell 6 and a point in cell 24 simultaneously.
- a single unit disk centered at a point $p \in D$ can cover (some or all) points in cells numbered 7, 8, 9, 10, 12, 13, 14, 15, 17, 18, 19, 20, 22, 23 and 24 but it can not cover a point in cell 10 and a point in cell 22 simultaneously.

Thus, the observation follows. □

In our approximation algorithm, we select one point from each cell that contains at least one point. The stepwise description of the proposed method is given in Algorithm 1.

Theorem 1 *The approximation factor of our algorithm is 14, and its running time is $O(n \log n)$.*

Proof. Consider a disk in the optimum solution. By Observation 1, it can cover points in at most 14 cells.

Algorithm 1 MDS_14-FACTOR(P)

Input: Set P of points in a 2-dimensional plane.
Output: A Set $P^* \subseteq P$ such that the unit disks centered at points in P^* cover all the points in P.
Set $P^* \leftarrow \emptyset$.
Consider a partitioning of the plane into a grid whose each cell is of size $\frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}}$.
/* A grid cell (α, β) is said to be less than another grid cell (γ, δ) if and only if either $\alpha < \gamma$ or $\alpha = \gamma$ and $\beta < \delta$ */
Consider a height balanced binary tree $\mathcal{T}$ for storing the non-empty grid cells. Each element of $\mathcal{T}$ is a tuple (α, β) indicating the indices of a non-empty cell. It is attached with any point $p_i \in P$ that lies in that cell (as the piercing point). For each point $p_i = (x_i, y_i) \in P$, we compute the indices of the grid cell $\alpha = \lceil \frac{x_i}{\sqrt{2}} \rceil$ and $\beta = \lceil \frac{y_i}{\sqrt{2}} \rceil$. If the tuple (α, β) is not in $\mathcal{T}$, we store it in $\mathcal{T}$ and attach p_i with it. Otherwise (i.e., if (α, β) is in $\mathcal{T}$), we have nothing to do.
for (each node v of $\mathcal{T}$) **do**
  Let p be the point attached to the node v. Set $P^* \leftarrow p$
end for
return P^*

But, we have chosen at most 14 different disks to cover those points. Thus, the approximation factor follows.

In order to justify the time complexity, we shall not construct the grid explicitly. We maintain a height balanced binary tree $\mathcal{T}$ for storing the non-empty grid cells. The processing of each point requires only the checking of the corresponding grid cell in $\mathcal{T}$. After processing all the points in P, we need to visit $\mathcal{T}$ for reporting the piercing points. Thus the time complexity result follows. □

2.2 Improving the approximation factor to 4

We now show that we can have a 4-factor approximation algorithm by increasing the worst case running time. We partition the plane into a grid whose each cell is of size $\frac{3}{\sqrt{2}} \times \frac{3}{\sqrt{2}}$ as shown in Figure 2(a).

Lemma 2 *The minimum piercing set of the unit disks centered at the points inside a grid cell of size $\frac{3}{\sqrt{2}} \times \frac{3}{\sqrt{2}}$ can be computed in $O(n^9)$ time.*

Proof. Let us consider a grid cell of size $\frac{3}{\sqrt{2}} \times \frac{3}{\sqrt{2}}$. We use χ to denote the cell, and P_χ to denote the set of points inside this cell. We split χ into 9 subcells of size $\frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}}$ (in Figure 2(b) it is shown separately.). In order to get the minimum cardinality subset of P for

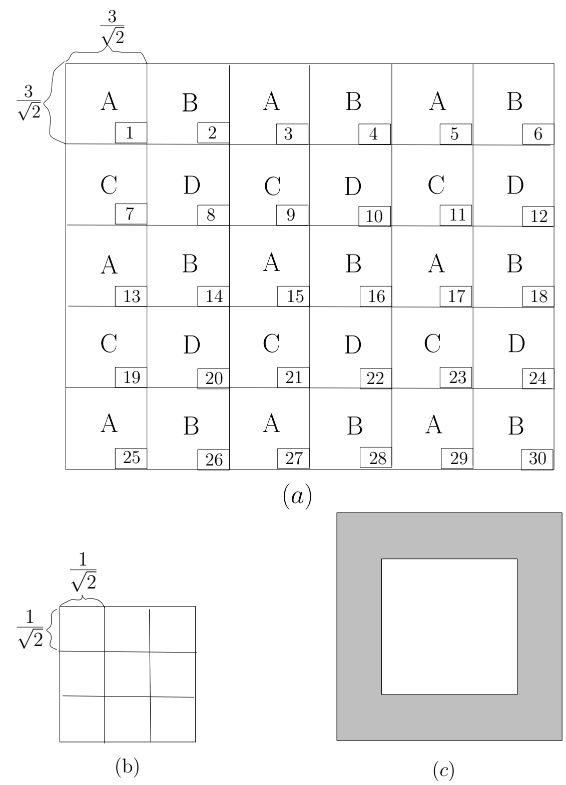

Figure 2: Proof of Lemma 2

piercing the disks centered at the points in P_χ, we need to identify the minimum number of points in P such that the disks centered at those points can cover all the points in P_χ. Note that, if all the 9 cells are non-empty, we need at most 9 disks to cover all the points in P_χ. The reasons are (i) the disk centered at any point inside a subcell covers all the points inside that subcell, and (ii) each non-empty subcell of χ can contribute one such point.

In order to cover the points P_χ, we need to consider the disks centered at the points in P that lie in χ and the shaded region around χ as shown in Figure 2(c). Let this set of points be Q_χ. We choose every point of Q_χ, and check whether the disk centered at that point covers all the points in P_χ. If it fails for all the points in Q_χ, then we choose each pair of points $p, q \in Q_\chi$ and test whether each point in P_χ lies inside at least one disk centered at p and q. If it fails again, we need to choose each triple of points of Q_χ and so on. Finally, we need to choose each tuple of 8 points from Q_χ and test whether each point in P_χ lies inside one of the disks centered to those 8 points. In each step, the checking needs $O(n)$ time. Thus, these 8 steps needs in total $O(n^9)$ time in the worst case. If the 8-th step also fails, we choose one point in each cell arbitrarily, and put a disk centered at those 9 points. Thus, the time complexity of this optimal algorithm follows. □

The stepwise description of the method described in Lemma 2 is given in Algorithm 2. Next, we use the Algorithm 2 for designing Algorithm 3 for getting a 4-factor approximation result for the discrete piercing problem.

Algorithm 2 OPT(χ, P_χ, Q_χ)

Input: The cell χ of size $\frac{3}{\sqrt{2}} \times \frac{3}{\sqrt{2}}$, set $P_\chi \subseteq P$ of points inside cell χ, and set $Q_\chi \subseteq P$ of points such that each unit disk centered at the points in Q_χ covers at least one point in P_χ.
Output: Set $P^* \subseteq Q_\chi$ such that the unit disks centered at points in P^* cover all the points in P_χ.
Set $P^* \leftarrow \emptyset$ and *flag* $\leftarrow$ *false*.
for (k = 1, 2, ..., 8) **do**
  Choose each k points from each Q_χ, and check whether the disks centered at these k points cover all the points in P_χ.
  If the answer of the above step is true, then add these k points in the set P^*, set *flag* = *true*, and break the *for* loop.
end for
if (*flag* = *false*) **then**
  Divide χ into 9 subcells of size $\frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}}$. Choose one point of P_χ from each of these 9 subcells and add these 9 points to the set P^*.
end if
return P^*

Observe that a unit disk in the optimum solution of a cell χ does not cover any point of some other cell ψ unless ψ is one of the eight neighboring cells of χ. We color the cells with minimum number of colors such that the unit disks placed in the cells of same color are non-overlapping irrespective of which point is chosen (as the center of the disk) in those cells. Thus, if we color cell number 1 (top-left cell) of the grid by A, we need to assign three different colors, say B, C and D to its three neighboring cells numbered 2, 7 and 8, which in turn are neighbors to each other (see Figure 2). But, we can again assign color A to cell 3. Thus, we have the following result.

Lemma 3 *The minimum number of colors required to color the cells of the grid is 4.*

Proof. We assign color to the cells in the grid from top row to the bottom row, and the cells in each row are colored from left to right. While assigning color to a cell, at most three of its neighbors are already colored. These are all different since the corresponding cells are neighbor to each other. So, we may assign the remaining fourth color to it. □

Theorem 4 *A 4-factor approximation algorithm for* *the minimum discrete piercing set problem for unit disk exists with time complexity* $O(n^9)$.

Proof. Consider the cells colored by A. Since the distance between any two cells with color A is at least $\frac{3}{\sqrt{2}}(> 2)$, a unit disk can covers only the points in a single cell with color A. Therefore, the set of disks in the optimum solution of one cell colored with A do not cover any point in any other cell colored with A. Thus, the optimum solution of the points of all the cells colored with A can be computed by choosing each A colored cell independently, and computing its optimum solution. Let us denote this solution by OPT_A. Surely $|OPT_A| \leq |OPT|$, where OPT is the optimum solution for the point set P distributed on the plane. Similarly OPT_B, OPT_C and OPT_D denote the optimum solution of the cells colored as B, C and D. The approximation factor of our algorithm follows from the fact that $|OPT_A| + |OPT_B| + |OPT_C| + |OPT_D| \leq 4|OPT|$, and our reported answer is $OPT_A \cup OPT_B \cup OPT_C \cup OPT_D$.

The time complexity follows from the fact that we are using at most $O(n)$ points while computing the optimum solution of a cell, and we are computing the optimum solution for only non-empty cells, which may be $O(n)$ in the worst case. □

2.3 Improving the approximation factor to 3

We now improve the previous method by reducing dependency between cells. As in the earlier sections, here also we need to partition the region into cells as follows. We split the plane into horizontal strip of width $\frac{3}{\sqrt{2}}$. Each odd numbered strip is divided into equal sized cells of width $\frac{6}{\sqrt{2}}$. The horizontal width of the last cell may be less than $\frac{6}{\sqrt{2}}$, depending on the horizontal width of the region. Each even numbered strip is divided into cells such that the first cell is of width $\frac{3}{\sqrt{2}}$, and the other cells are of width $\frac{6}{\sqrt{2}}$, excepting the last cell as mentioned for odd numbered strips. Next, we assign color to the cells of the odd numbered strips using three colors A, B and C as shown in Figure 3. Now consider the cells in the even numbered strips, say strip 2. Cell 8 shares sides of two cells 1 and 2, which are already colored by A and B. So, we can color cell 8 by C. By the same reason, cells 7 and 9 are colored by C and A. Thus, the cells of each odd numbered strips are colored using the sequence $B, C, A, B, C, A, \ldots$. Such a coloring admits that no part of the disk centered at a point inside a cell of a particular color i $(\in \{A, B, C\})$ will lie in another cell of the same color i.

The maximum number of disks (centered at points in P) required to cover all the points is a cell of size $\frac{3}{\sqrt{2}} \times \frac{6}{\sqrt{2}}$ is 18. Arguing as in Subsection 2.2, the worst case time

Algorithm 3 MDS_4-FACTOR(P)

Input: Set P of points in a 2-dimensional plane.
Output: Set $P^* \subseteq P$ such that the unit disks centered at points in P^* cover all the points in P.
Set $P^* \leftarrow \emptyset$.
Consider a partitioning of the plane into a grid whose each cell is of size $\frac{3}{\sqrt{2}} \times \frac{3}{\sqrt{2}}$. /* A grid cell (α, β) is said to be less than another grid cell (γ, δ) if and only if either $\alpha < \gamma$ or $\alpha = \gamma$ and $\beta < \delta$ */
Consider an height balanced binary tree $\mathcal{T}$ for storing the non-empty grid cells. Each element of $\mathcal{T}$ is a tuple $\chi = (\alpha, \beta)$ indicating the indices of a non-empty cell. It is attached with two sets namely, P_χ and Q_χ where $P_\chi \subseteq P$ is the set of points inside the cell χ and $Q_\chi \subseteq P$ is the set of points such that the disk centered at the points in Q_χ covers at least one point in P_χ. For each point $p_i = (x_i, y_i) \in P$, we compute the indices of the grid cell $\alpha = \lceil \frac{3x_i}{\sqrt{2}} \rceil$ and $\beta = \lceil \frac{3y_i}{\sqrt{2}} \rceil$. If the tuple (α, β) is not in $\mathcal{T}$, we store it in $\mathcal{T}$ with corresponding P_χ and Q_χ. Otherwise (i.e., if (α, β) is in $\mathcal{T}$), we just modify the sets P_χ and Q_χ.
for (each node v of $\mathcal{T}$) **do**
  Run $\frac{3}{\sqrt{2}}$_X_$\frac{3}{\sqrt{2}}$_OPT(χ, P_χ, Q_χ) /* Algorithm 2 */
  Let P_1^* be the output of the above algorithm. Set $P^* = P^* \cup P_1^*$
end for
return P^*

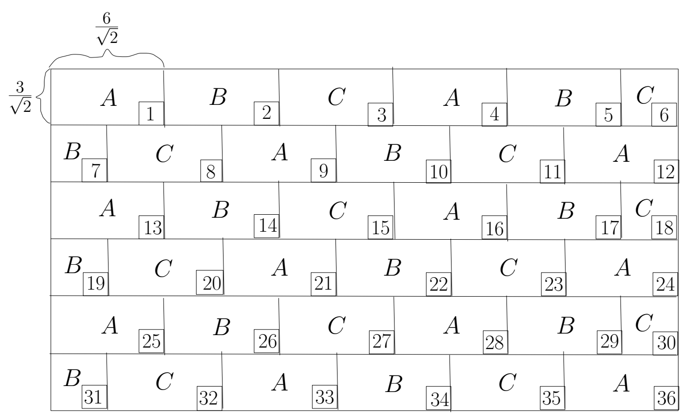

Figure 3: Coloring of cells for 3-factor approximation algorithm

needed for computing the minimum number of disks required to cover the points of P in such a cell is $O(n^{18})$. Thus, we have the following result:

Theorem 5 *A 3-factor approximation algorithm for the minimum discrete piercing set problem for unit disks exists with time complexity* $O(n^{18})$.

3 Conclusion

We proposed constant factor approximation algorithms for a variation of the discrete piercing set problem for unit disks, where the points chosen for piercing the disks will be from the set of center points of the disks given for piercing. The most simple algorithm produces 14-factor approximation result in $O(n \log n)$ time. We then improve the approximation factor to 4. Finally, we propose a 3-factor approximation algorithm, which is an improvement of the existing result by a factor of $\frac{5}{3}$ [20]. Though, the time complexity of our proposed 4- and 3-factor approximation algorithms are high, in actual scenario, they terminate very fast.

Finally, our algorithm can also be used to solve the h-piercing problem for unit disks as defined in [20]. Following the same method as in [20], it can be shown that the result obtained using our method is no worse than $3(2^h - 1)$-factor of the optimum solution of h-piercing problem. Thus, the result produced by our algorithm for the h-piercing problem is an improvement by a factor $\frac{5}{3}$ over the existing best known result [20].

References

[1] C. Ambuhl, T. Erlebach, M. Mihalak and M. Numkesser. Constant-factor approximation for minimum-weight (connect) dominating sets in unit disk graphs. Proc. of the *APPROX-RANDOM*, pp. 3–14, 2006.

[2] G. Calinescu, I Mandoiu, P. J. Wan and A. Zelikovsky. Selecting forwarding neighbors in wireless ad hoc networks. *Mobile Network Applications*, 9(2):101–111, 2004.

[3] P. Carmi, M. Katz and N. Lev-Tov. Covering points by unit disks of fixed location. Proc. of the *18th. International Symposium on Algorithms and Computation*, pp. 644–655, 2007.

[4] P. Carmi, M. J. Katz and N. Lev-Tov. Polynomial-time approximation schemes for piercing and covering with applications in wireless networks. *Computational Geometry: Theory and Applications*, 39(3):209–218, 2008.

[5] T. M. Chan. Polynomial-time approximation schemes for packing and piercing fat objects. *Journal of Algorithms*, 46(2):178–189, 2003.

[6] T. M. Chan and A. -A. Mahmood. Approximating the piercing umber of unit height rectangles. Proc. of the *17th. Canadian Conference on Computational Geometry*, pp. 15–18, 2005.

[7] B. N. Clark, C. J. Colbourn and D. S. Johnson. Unit disk graphs. *Discrete Math.*, 86(1):165–177, 1990.

[8] F. Claude, G. K. Das, R. Dorrigiv, S. Durochar, R. Fraser, A. Lopez-Ortiz and B. G. Nickerson. An improved line-separable algorithm for discrete unit disk cover. *Discrete Mathematics, Algorithms and Applications*, 2(1):77–87, 2010.

[9] M. R. Cerioli, L. Faria, T. O. Ferreira and F. Protti. On minimum clique partition and maximum independent set on unit disk graphs and penny graphs: complexity and approximation. *Electronic Notes in Discrete Mathematics*, 18:73–79, 2004.

[10] G. K. Das, R. Fraser, A. Lopez-Ortiz and B. G. Nickerson. On the discrete unit disk cover problem. Proc. of the *5th. Workshop on Algorithm and Computation*, LNCS-6552, pp. 146–157, 2011.

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

[12] A. Dumitrescu and J. Pach. Minimum clique partition in unit disk graphs. *Graphs and Combinatorics*, 27(3):399–411, 2011.

[13] T. Feder and D. H. Greene. Optimal algorithms for approximate clustering. Proc. of the *20th ACM Symposium on Theory of Computing*, pp. 434–444, 1988.

[14] T. F. Gonzalez. Covering a set of points in multidimensional space. *Information Processing Letters*, 40(4):181–188, 1991.

[15] 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(2):179–194, 2008.

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

[17] M. R. Garey and D. S. Johnson. Computers and Intractability – A Guide to the Theory of NP-completeness. Freeman, New York, 1979.

[18] D. Johnson. The NP-completeness column: an ongoing guide. *Journal of Algorithms*, 3(2):182–195, 1982.

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

[20] T. Nieberg and J. Hurink. A PTAS for the minimum dominating set problem in unit disk graphs. Proc. of the *3rd International Workshop on Approximation and Online Algorithms*, LNCS - 3879, pp. 296–306, 2006.

[21] S. Narayanappa and P. Voytechovsky. An improved approximation factor for the unit disk covering problem. Proc. of the *18th Canadian Conference on Computational Geometry*, 2006.

[22] I. Pirwani and M. Salavatipour. A weakly robust PTAS for minimum clique partition in unit disk graphs. Proc. of the *12th Scandinavian Symposium and Workshops on Algorithm Theory*, LNCS-6139, pp. 188–199, 2010.

[23] F. Zou, Y. Wang, X. -H. Xu, X. Li, H. Du, P. Wan and W. Wu. New approximations for minimum-weighted dominating sets and minimum-weighted connected dominating sets on unit disk graphs. *Theoretical Computer Science*, 412(3):198–208, 2011.