

Summer Research Report: A Variant of Covering Problem

Adit Kumar Parida¹ and Sourav Pati¹

¹National Institute of Technology Silchar, India,
{souravpati, aditparida}@student.nits.ac.in

June 30, 2016

Advisor : Minati De

University : Indian Institute Of Science, Bengaluru

Contents

1	Abstract	3
2	Introduction	4
2.1	Problems Definition	4
2.2	Motivation	4
3	Survey	4
3.1	Local Search Algorithm	5
4	Our Contribution	7
4.1	4-Factor Approximation Using Hexagonal Grid	8
4.2	3-Factor Approximation Using Rectangular Grid	10
4.3	Polynomial Time Approximation Scheme	11
5	Generalizing Problem to 3 Dimension	12
5.1	Coloring Scheme	12
5.2	Algorithm	13
6	Future Scope	14
7	Conclusion	14

1 Abstract

In this summer research report, we discuss about the problems we studied and analyzed for our research work. We studied 4-factor and 3-factor algorithms for solution of geometric minimum dominating set problem. We obtain a 4-factor algorithm by splitting the plane into hexagonal grid which takes $O(n^{7d})$. In 3 - factor algorithm it takes $O(n^{18d})$ time to obtain solution. A simplified version of the 3D problem is also considered and provided with a 8 - *factor* algorithm.

2 Introduction

2.1 Problems Definition

Given a set of points $p_i \in \mathcal{P}$ with demand d each, there exist a set $S \subseteq \mathcal{P}$ such that unit disks placed on the points in S will cover all the points in $\mathcal{P} \setminus S$ with all its demand satisfied. To satisfy one point's demand we need to cover it with at least d many disks.

2.2 Motivation

Our motivation comes from real world application in the usage of wireless sensor network where there are towers at certain points in the 2-Dimensional Euclidean space to satisfy the demands of users placed in regions within the same Euclidean space. Here a new problem may arise when the demand of each user may not be satisfied with a single tower and thereby require multiple towers to cover their demands. This real world problem can be represented mathematically as a multi cover problem or a piercing set problem where the points associated with the 2D space represent each user with a demand d , satisfied by multiple towers surrounding it.

3 Survey

We analysed PTAS for Weighted Set Cover on Unit Squares by Erlebach and Leeuwen [4] where a PTAS for Geometric Set Cover on any set of axis-parallel unit squares is proposed. A optimal solution for a set of points on a horizontal strip of height k was found in $n^{O(k)}$ with the use of sweep line algorithm. Combining this with the well-known shifting technique yields the PTAS.

A PTAS for the Weighted Unit Disk Cover Problem by Li and Jin[7], a paper on weighted disk cover problem provides $(1 + \epsilon)$ - factor algorithm. Overall time complexity of the dynamic program is $O(n^{2K^2+1}) = n^{O(1/\epsilon^4)}$.

Next, we focused on dominating set problem using local search to apply the multi cover concept for unweighted points. A research work done by Gibson and Pirwani in Approximation Algorithms for Dominating Set in Disk Graphs[6], which uses local search to find PTAS for the solution of problem. We start with a feasible solution first. Consider D is a set of disk from which we need to find a dominating set. $B \subseteq D$ is b -locally optimal if one cannot obtain a smaller dominating set by removing a subset $X \subseteq B$ of size at most b from B and replacing that with a subset of size at most $|X| - 1$ from $D \setminus B$. Local search gives a b -locally optimal set of disks for $b = c/\epsilon^2$ where $c > 0$ is a large constant. Local search start with arbitrary solution set B and continues to do local improvements. It stops

when no further local improvements are possible.

If a disk in b is properly contained in another disk of $D \setminus B$, then we replace it by the larger disk from $D \setminus B$. Our replacement step ensures that there is no disk in B which is properly contained in some other disk in D .

Locality Condition: Let R be the disks obtained from optimal solution and B be the disks obtained from the local algorithm. Then there exists a planar graph with vertex set $R \cup B$, such that for every $d \in D$, there is a disk u from amongst the red dominators of d and a disk v among the blue dominators of d such that u, v is an edge in the graph. [6]

Above Condition was proved by the use of weighted voronoi diagram. Meticulous observation will prove that the diagram made by $R \cup B$ is actually star shaped and planar. Using the theorem of planarity by G. N. Frederickson, Fast algorithms for shortest paths in planar graphs with applications[5], we can use a planner separator to prove that our solution is $1 + \epsilon$ factor of optimal solution.

3.1 Local Search Algorithm

Algorithm:[3]

Input: A set of n objects \mathcal{P} in \mathbb{R}^2

1. Initialize \mathcal{Q} to an arbitrary subset of \mathcal{P} which is a feasible solution
2. while $\exists \mathcal{X} \subseteq \mathcal{Q}$ of size at most b , and $X_0 \subseteq \mathcal{P}$ of size at most $\mathcal{X} - 1$ such that $(\mathcal{Q} \cap \mathcal{X}) \cup X_0$ is a feasible solution. do
3. set $\mathcal{Q} \leftarrow (\mathcal{Q} \setminus \mathcal{X}) \cup X_0$
4. while \exists some object $Q \in \mathcal{Q}$ such that region $Q \setminus \bigcup_{P(\neq Q) \in \mathcal{Q}} P$ is completely contained in some object $R \in \mathcal{P} \setminus \mathcal{Q}$ do
- 5 replace Q by R in \mathcal{Q}
- 6 Report \mathcal{Q}

In the above case, a solution is feasible only when all the demands of the points are satisfied i.e. at least d many disks cover a point $p_i \in \mathcal{P}$. The subsets of \mathcal{Q} is called b -locally optimal if it is not possible to obtain another feasible solution by removing a subset X from \mathcal{Q} and replacing it by a subset X_0 of smaller size. We continue to update our solution until no such X_0 can be replaced. After obtaining the \mathcal{Q} from the first loop we check whether the object in \mathcal{Q} are properly contained by any other objects in the set. If they are contained by any other object, we replace it.

The time taken by the first loop is $O(n)$ and each iteration consumes time in $O(n^b)$. So total time for completion of first loop will be $O(n^{b+1})$. Second loop needs at most $O(n^2)$ time. Hence running time of the algorithm is $O(n^{b+1})$.

Analysis:

In case of geometric minimum dominating set local search is guaranteed to give $(1 + \epsilon)$ factor approximation. The proof is based on the graph separator theorem proposed by G. N. Frederickson, Fast algorithms for shortest paths in planar graphs, with applications[5]. Above theorem is applicable only to planar graphs. So we will approach our problem by creating the graph by setting the locality condition. If we can prove the graph to be planar then with the help of planarity theorem proposed by Frederickson we can prove that we have $(1 + \epsilon)$ - approximation algorithm.

Now, we consider an optimal solution is OPT . the optimal solution, OPT , must have less or equal number of disks compared to any other solutions obtained from our algorithm. But in every feasible solution set, the demands of points are satisfied. So at least d many disks must cover each point. \mathcal{ALG} is considered to be our solution set from the algorithm. We can derive our locality condition for multi cover problem from the locality condition stated in geometric minimum dominating set problem[6].

Notation:

Let OPT be the optimal solution and Q be the solution returned by our local search algorithm. Our algorithm ensures that there is no object $S \in \mathcal{S} \setminus Q$ which completely contains any object of Q . Similarly, we can assume that no object in OPT is completely contained in any object from $\mathcal{S} \setminus OPT$. Let $Q' = Q \setminus (Q \cap OPT)$, $OPT' = OPT \setminus (Q \cap OPT)$. We show that $|Q'| \leq (1 + \epsilon)|OPT'|$ which implies that $|Q| \leq (1 + \epsilon)|OPT|$.

Locality Condition :

Lemma 1. *[Locality Condition] There exist a graph $\mathcal{G} = (\mathcal{ALG}, OPT, \mathcal{E})$, where there is at least d number of unit disks u_i covering a point $p_i \in P$ from set \mathcal{ALG} and d number of unit disks v_i from OPT such that there are d many edges between (u_i, v_i) .*

Analysis of Locality Condition:

We will consider an edge between $u \in \mathcal{ALG}$ and $v \in OPT$, if they cover a same point $p \in \mathcal{P}$. In our problem, each point has d demand. Hence there are at least d many disks covering each point from each set \mathcal{ALG} and OPT . Without loss of generality, we can assume that \mathcal{ALG} and OPT are disjoint sets. $\mathcal{ALG} \cap OPT = \phi$. So we can derive the conclusion that for a point p_i , d disks from \mathcal{ALG} and d disks from OPT are distinct disks. Consider a disk which covers p_i and belongs to \mathcal{ALG} . It will have an edge with disks from

\mathcal{OPT} covering the same point p_i . So for d disks in \mathcal{OPT} we will have total d number of edges to \mathcal{OPT} .

Lemma 2. *[Locality Condition for Set Cover] There exists a planar graph $\mathcal{G} = (\mathcal{Q}' \cup \mathcal{OPT}', \mathcal{E})$ such that for all $p \in \mathcal{P}'$ there exists an edge $(U, V) \in \mathcal{E}$ where both U and V covers \mathcal{P}' , and $U \in \mathcal{Q}'$ and $V \in \mathcal{OPT}'$.*

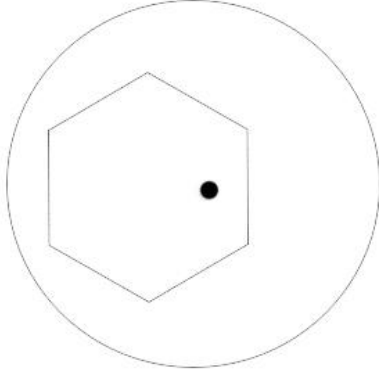
$$|\mathcal{Q}'_i| \leq |\mathcal{OPT}'_i| + |N(\mathcal{V}_i)|, \quad (1)$$

$$\begin{aligned}
|\mathcal{Q}'| &\leq |\mathcal{X}| + \sum_i |\mathcal{Q}'_i| && \text{(Each element of } \mathcal{Q}' \text{ either belongs to } \mathcal{Q}'_i \text{ or } \mathcal{X}) \\
&\leq |\mathcal{X}| + \sum_i |\mathcal{OPT}'_i| + \sum_i |N(\mathcal{V}_i)| && \text{(Follows from Equation 1)} \\
&\leq |\mathcal{OPT}'| + |\mathcal{X}| + \sum_i |N(\mathcal{V}_i)| && (\mathcal{OPT}'_i \text{ are disjoint subset of } \mathcal{OPT}') \\
&\leq |\mathcal{OPT}'| + \frac{c'_1(|\mathcal{Q}'| + |\mathcal{OPT}'|)}{\sqrt{b}} && (|N(\mathcal{V}_i)| \leq c_3\sqrt{r} \text{ and } |\mathcal{X}| \leq c_1(|\mathcal{Q}' + \mathcal{OPT}'|)/\sqrt{r}) \\
|\mathcal{Q}'| &\leq \frac{1 + c'_1/\sqrt{b}}{1 - c'_1/\sqrt{b}} |\mathcal{OPT}'| && \text{(By rearranging)} \\
|\mathcal{Q}'| &\leq (1 + \epsilon) |\mathcal{OPT}'| && (b \text{ is large enough constant times } \frac{1}{\epsilon^2}).
\end{aligned}$$

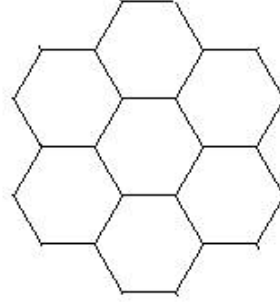
In the paper, Minimum Dominating Set Problem for Unit Disks Revisited by Carmi et al [1] a 5 - factor approximation algorithm is proposed which runs in $O(n \log k)$ time. (k is size of output). A 4 factor approximation algorithm is broached by dividing the plane into hexagonal grid instead of rectangular grid. Time complexity of the algorithms is $O(n^7)$, which can be improved to $O(n^6 \log n)$. After using shifting strategy we yield PTAS from Geometric Minimum Dominating Set problem[2]. This technique helps us in our work for multi cover problem.

4 Our Contribution

We fixed the demand $d = 2$ for simplification and used the algorithm as discussed in Minimum Dominating Set Problem for Unit Disks Revisited by Paz Carmi et al.[1]. We first obtained a 4-factor approximation algorithm for demand $d = 2$ using hexagonal grid with a time complexity of $O(n^{14})$ and then generalized it for a general case of demand d with a time complexity of $O(n^{7d})$.



(a) Unit Radius Disk Covering a Hexagonal Cell



(b) A Septa-Hexagon

Figure 1: Hexagonal Grid

Simplifying The problem.

Now, we consider $d = 2$ for the purpose of further simplification. We try to solve the problem with a different approach by dividing the plane \mathbb{R}^2 on which the points reside.

4.1 4-Factor Approximation Using Hexagonal Grid

We first partition a region \mathbb{R}^2 containing points in the set \mathcal{P} into regular hexagonal cells, where the side length of each cell is $\frac{1}{2}$. The length of the diagonal is 1.

Observation 1 *Since any pair of points within the hexagonal cell is at most 1 distance apart (length of diagonal is one), an unit disk placed at any point in the cell can cover all the points within that cell(see figure 1a).*

Definition A septa-hexagon is a combination of 7 adjacent cells such that the center cell is surrounded by six similar cells (see Figure 1b).

We consider hexagonal grid in \mathbb{R}^2 such that no point $p_i \in \mathcal{P}$ lie on the boundary of the grid. Next we implement a four coloring scheme on it, where each septa-hexagon is assigned a color and no two adjacent septa-hexagon is assigned same color. So if a septa-hexagon is colored A then its adjacent septa-hexagon will be colored B, C and D and each pair of those adjacent septa-hexagon with same color(say B) will be opposite to each other.(see figure 2)

Let us consider the septa-hexagon of a particular color and study its geometry. This leads to the following observation.

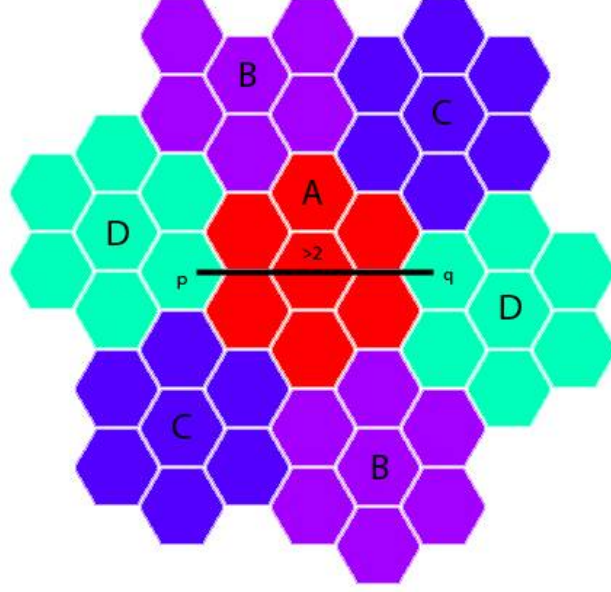


Figure 2: 4-Coloring Scheme in Hexagonal Grid

Observation 2 *Since the septa-hexagon is constituted of regular hexagonal cells of side length $1/2$, the minimum distance of two opposite points on the boundary of the septa-hexagon is 2.*

Now, let the septa-hexagon be \mathcal{T} and its two adjacent similar colored septa-hexagon be \mathcal{T}' and \mathcal{T}'' . We can conclude from the above observation that, if a tower of unit radius is placed at any point in a segment \mathcal{T}' will never pierce or intersect any point in the segment \mathcal{T}'' . Mathematically, if \mathcal{T}' and \mathcal{T}'' are two closest septa-hexagon with same color then a tower of disk D cannot simultaneously cover a point $p \in \mathcal{T}' \cap \mathcal{P}$ and a point $q \in \mathcal{T}'' \cap \mathcal{P}$. So, we can consider each colored segment independently i.e. let us take all the segment of color (say A).

Lemma 3. *[Polynomial Time Complexity] A solution comprising of towers placed in the region of a particular color(say A) of a 4-color scheme septa-hexagon \mathcal{ALG}_A , satisfying demand d , can be obtained in a polynomial time of order $O(n^{7d})$.*

Proof. Consider the septa-hexagon \mathcal{T} having color (say A). We first assume that each of the points has a demand $d = 2$. Since each hexagonal cell can be completely covered by the disk of unit radius, we can say that a maximum of 7 disks are necessary to cover a septa-hexagon once. In order to solve this we start by taking all combination of points $i = 1, 2, 3 \dots 7$ and return the result that is satisfied by minimum number of disks. This algorithm will return a solution better than optimal solution for a single septa-hexagon

and also the time complexity to go through all the combination will be of the order $O(n^7)$. Similarly, we can say that if each point $p \in \mathcal{P}$ is associated with demand 2, then we need to cover the entire individual region twice and for that we require a maximum of 14 disks. If we employ a similar strategy by taking all combination of points $i = 2, 3 \dots 14$ and return the result that is satisfied by minimum number of disks, we are still able to achieve it in a linear time of $O(n^{14})$. This can be generalized further with each point having demand d and still the time complexity will remain polynomial i.e. $O(n^{7d})$. \square

Let us consider the result we obtained after solving the region having color A is \mathcal{ALG}_A . We can observe that $|\mathcal{ALG}_A| \leq |\mathcal{OPT}|$ as \mathcal{ALG}_A covers a partial region of the entire plane in \mathbb{R}^2 , while \mathcal{OPT} covers the entire plane.

Lemma 4. *[4-Factor Approximation] $|\mathcal{ALG}| \leq 4 |\mathcal{OPT}|$, where \mathcal{ALG} is the solution obtained from the algorithm and \mathcal{OPT} is the optimal solution such that they cover all the point while satisfying the demand d .*

Proof. From above, we know that $|\mathcal{ALG}_A| \leq |\mathcal{OPT}|$. Similarly, $|\mathcal{ALG}_B|, |\mathcal{ALG}_C|, |\mathcal{ALG}_D| \leq |\mathcal{OPT}|$ as each of the colored elements cover partial region[1].

$$\begin{aligned} |\mathcal{ALG}_A| &\leq |\mathcal{OPT}|, \\ |\mathcal{ALG}_A| \cup |\mathcal{ALG}_B| \cup |\mathcal{ALG}_C| \cup |\mathcal{ALG}_D| &\leq 4|\mathcal{OPT}|, \\ |\mathcal{ALG}| &\leq 4|\mathcal{OPT}| \end{aligned}$$

\square

Theorem 1. *Given a set \mathcal{P} of n points in \mathbb{R}^2 with demand d each, proposed algorithm gives a 8- approximation solution to our problem in time complexity of $O(n^{7d})$.*

4.2 3-Factor Approximation Using Rectangular Grid

Main idea for our problem is taken from the paper[2]. In this case, we will partition the plane into horizontal strips of uniform width of $\frac{3}{\sqrt{2}}$. The odd numbered strip will be divided into blocks of length $= \frac{6}{\sqrt{2}}$. In the even numbered strips, it starts with block of length $\frac{3}{\sqrt{2}}$ and continues with $\frac{6}{\sqrt{2}}$ lengths. Then we start coloring the plane blocks using three different colors. Three colors are enough to color them in a way that two blocks with same color are not adjacent.

Now, we divide each block into smaller blocks of size $\frac{1}{\sqrt{2}}$ of each side. If we calculate, 18 number of smaller blocks can be accommodated in the larger block of size $\frac{6}{\sqrt{2}} \times \frac{3}{\sqrt{2}}$. The

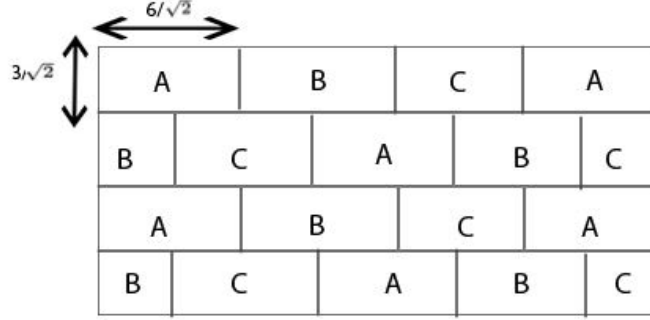


Figure 3: 3-Coloring Scheme In Square Grid

diagonal of each smaller block will be 1 unit each. So one disk is enough to cover one smaller block. At most 18 disks will be required to cover the larger block. Now, we can pick 18 disks from n disks inside the block to cover it completely. But in our problem each point has a demand d . So we have to pick d disks from the smaller blocks to cover the block entirely and satisfy the demands of points inside it. For the larger block it will require at most $18 \times d$ disks to cover the points and satisfy all the demands as well. So the time taken for choosing at most $18d$ disks will be $O(n^{18d})$.

Lemma 5. [3-Factor Approximation] $|\mathcal{ALG}| \leq 3 |\mathcal{OPT}|$, where \mathcal{ALG} is the solution obtained from the algorithm and \mathcal{OPT} is the optimal solution such that they cover all the point while satisfying the demand d .

Proof. Consider a block b_i . Solution acquired to the block from our algorithm be \mathcal{ALG}_i .

So the solution to block colored A is \mathcal{ALG}_A and others are \mathcal{ALG}_B and \mathcal{ALG}_C respectively.

Optimal solution is noted by the symbol \mathcal{OPT} .

$$\begin{aligned} |\mathcal{ALG}_A| &\leq |\mathcal{OPT}|, \\ |\mathcal{ALG}_A| \cup |\mathcal{ALG}_B| \cup |\mathcal{ALG}_C| &\leq 3|\mathcal{OPT}|, \\ |\mathcal{ALG}| &\leq 3|\mathcal{OPT}| \end{aligned}$$

□

Theorem 2. Given a set \mathcal{P} of n points in \mathbb{R}^2 with demand d each, proposed algorithm gives a 3-approximation solution to our problem in time complexity of $O(n^{18d})$.

4.3 Polynomial Time Approximation Scheme

Using shifting strategy similar to which used in [2], we can obtain PTAS for our problem. We divide the plane in to squares of size $2k \times 2k$ which can contain $(\lceil 2\sqrt{2}k \rceil)^2$ number of

cubes of size $\frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}}$. So we will need at most $(\lceil 2\sqrt{2}k \rceil)^2 d$ disks to cover the complete square. So using brute force we can choose these disks in $O(n^{\lceil 2\sqrt{2}k \rceil^2 d})$ time complexity.

Theorem 3. *Given a set \mathcal{P} of n points in \mathbb{R}^2 and an integer $k \geq 1$, the proposed algorithm delivers a discrete piercing set of the disks centered at all the points of \mathcal{P} in $O(k^2 n^{\lceil 2\sqrt{2}k \rceil^2 d})$ time, whose size is at most $(1 + \frac{1}{k})^2 \text{OPT}$, where OPT is the size of the Optimal solution[2].*

5 Generalizing Problem to 3 Dimension

Our problem can be generalized to a 3 Dimensional space where instead of unit disks we will consider unit spheres. Each sphere will have demand $d = 1$. We can divide the plane into cubes of size $\frac{4}{\sqrt{3}} \times \frac{4}{\sqrt{3}} \times \frac{4}{\sqrt{3}}$.

Definition 1. *Disjoint cubes : Two cubes are said to be disjoint or non adjacent if and only if they do not share a face or an edge or a vertex.*

One cube can be adjacent to maximum 26 number of cubes. The position will be in the middle most cube of the 3 dimensional cubic matrix with $3 \times 3 \times 3$ cubes arranged in a cubic manner. Proposed matrix can be denoted in the 3- dimensional array notation(matrix[dimension][row][column]). We will denote our matrix by the variable $M[\text{dimension}][\text{row}][\text{column}]$. We will adapt the zero-based indexing system. So 0^{th} element will be the first element in our sequence. For an example; $M[0][0][0]$ is the first element in the first row of the first dimension.

Our next step is to color the cubes inside of our matrix. The rule to color the cubes is : no adjacent cubes are given the same color. Following this rule we can color the entire matrix using at least 8 different colors.

5.1 Coloring Scheme

We can infer from the structure of the matrix M that the middle cube in position $M[1][1][1]$ will be adjacent to every other cube in our matrix. Hence, we can give it a color different to every other cubes. Assume the color is A . We can remove the colored cube from our matrix for simplification of further coloring. As no adjacent cube will be given same color, we start looking for disjoint subset of cubes from the remaining matrix and give them a single color. The corner most cubes are disjoint to each other. So the cubes $M[0][0][0]$, $M[0][0][2]$, $M[0][2][0]$, $M[0][2][2]$, $M[2][0][0]$, $M[2][0][2]$, $M[2][2][0]$, $M[2][2][2]$ are given one color, say B . The middle cube of the opposite faces can be given one color. So

$M[0][1][1]$ and $M[2][1][1]$ is colored with C . Similarly we can color $M[1][1][0]$ and $M[1][1][2]$ with D and $M[1][0][1]$ and $M[1][2][1]$ with E . We remove them after coloring. Now, 3 disjoint subsets, $\{M[1][0][0], M[1][0][2], M[1][2][0], M[1][2][2]\}$, $\{M[0][0][1], M[0][2][1], M[2][0][1], M[2][2][1]\}$, $\{M[0][1][0], M[0][1][2], M[2][1][0], M[2][1][2]\}$ remain which can be given color F, G, H respectively.

Consider our algorithm (discussed later) provides solution for each of the colored cubes. Let us consider our algorithm returns $\mathcal{ALG}_A, \mathcal{ALG}_B, \mathcal{ALG}_C, \mathcal{ALG}_D, \mathcal{ALG}_E, \mathcal{ALG}_F, \mathcal{ALG}_G, \mathcal{ALG}_H$ as our solution set for each colored group. Optimal solution for the problem is considered as \mathcal{OPT} .

Observation 1. Consider two cubes, \mathcal{C}_i and \mathcal{C}_j , are of same color, then a single sphere can not contain a point from the space covered by \mathcal{C}_i and \mathcal{C}_j , because the distance between the two points is more than 2.

5.2 Algorithm

The number of sphere required to completely cover a cube of size $\frac{1}{\sqrt{3}} \times \frac{1}{\sqrt{3}} \times \frac{1}{\sqrt{3}}$ is 1. As the diagonal of the cube is 1, we can cover the cube in just one sphere. Our larger cube of size $\frac{4}{\sqrt{3}} \times \frac{4}{\sqrt{3}} \times \frac{4}{\sqrt{3}}$ can contain total of 64 numbers of smaller cubes. So we will need at least 64 number of spheres to completely contain a larger cube. We check all combination of 64 cubes to find our solution which can be carried out in $O(n^{64})$ time complexity.

Lemma 6. [8-Factor Approximation] $|\mathcal{ALG}| \leq 8 |\mathcal{OPT}|$, where \mathcal{ALG} is the solution obtained from the algorithm and \mathcal{OPT} is the optimal solution such that they cover all the points.

Proof. Consider a block b_i . Solution acquired to the block from our algorithm be \mathcal{ALG}_i .

Optimal solution be \mathcal{OPT} . Now:

$$\begin{aligned} |\mathcal{ALG}_A| &\leq |\mathcal{OPT}|, \\ |\mathcal{ALG}_A| \cup |\mathcal{ALG}_B| \cup |\mathcal{ALG}_C| \cup |\mathcal{ALG}_D| \cup |\mathcal{ALG}_E| \cup |\mathcal{ALG}_F| \cup |\mathcal{ALG}_G| \cup |\mathcal{ALG}_H| &\leq 8|\mathcal{OPT}|, \\ |\mathcal{ALG}| &\leq 8|\mathcal{OPT}| \end{aligned}$$

□

Theorem 4. Given a set \mathcal{P} of n points in \mathbb{R}^3 using the coloring scheme and applying our algorithm gives us $\mathcal{ALG} \leq 8 \mathcal{OPT}$, where \mathcal{ALG} is the solution returned to us by our algorithm and \mathcal{OPT} is the optimal solution in $O(n^{64})$ time complexity.

6 Future Scope

Further work is possible by considering multi cover problems with variable demands. Here each point can be associated with some demand other than a fixed demand d . Also, study related to the multi cover or piercing set problem where there is a fixed demand d , but each point is associated with variable weights. In this case, we can use a similar strategy as discussed in PTAS for Weighted Set Cover on Unit Squares by Thomas Erlebach and Erik Jan van Leeuwen [4] and A PTAS for the Weighted Unit Disk Cover Problem by Jian Li and Yifei Jin[7], by dividing the plane into a small region of $k \times k$ squares and use it to obtain a near OPT or OPT solution. Then we can employ the concept of shifting strategy to solve it further for a good Approximation Algorithm. In the case of 3 Dimensional problem we can give uniform demand $d \geq 2$ which can be further improved to more complex problem such as giving variable demand to each points in 3D space.

7 Conclusion

We have a solution of 4- approximation algorithm for the problem of uniform multi set cover problem which can be solve in $O(n^{7d})$ time complexity. Algorithm can be improved to 3 - approximation which can be solved in $O(n^{18d})$ time complexity. Then, we obtained PTAS by using shifting strategy. We considered a similar problem in 3- dimensional space and calculated a 8- factor approximation algorithm which requires $O(n^{64})$ time complexity.

References

- [1] Paz Carmi, Gautam K. Das, Ramesh K. Jallu, Subhas C. Nandy, Prajwal R. Prasad, and Yael Stein. Minimum dominating set problem for unit disks revisited. *Int. J. Comput. Geometry Appl.*, 25(3):227, 2015.
- [2] Minati De, Gautam K. Das, Paz Carmi, and Subhas C. Nandy. Approximation algorithms for a variant of discrete piercing set problem for unit disks. *Int. J. Comput. Geometry Appl.*, 23(6):461, 2013.
- [3] Minati De and Abhiruk Lahiri. Geometric dominating set and set cover via local search. *CoRR*, abs/1605.02499, 2016.
- [4] Thomas Erlebach and Erik Jan van Leeuwen. PTAS for weighted set cover on unit squares. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 13th International Workshop, APPROX 2010, and 14th*

International Workshop, RANDOM 2010, Barcelona, Spain, September 1-3, 2010. Proceedings, pages 166–177, 2010.

- [5] Greg N. Frederickson. Fast algorithms for shortest paths in planar graphs, with applications. *SIAM J. Comput.*, 16(6):1004–1022, 1987.
- [6] Matt Gibson and Imran A. Pirwani. Approximation algorithms for dominating set in disk graphs. *CoRR*, abs/1004.3320, 2010.
- [7] Jian Li and Yifei Jin. A PTAS for the weighted unit disk cover problem. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, pages 898–909, 2015.