M. Tech. (Computer Science) Dissertation

Algorithms on Geometric Graphs

A dissertation submitted in partial fulfillment
of the requirements for the award of
M.Tech.(Computer Science) degree

By

Minati De
Roll No: CS0805

under the supervision of

Professor Subhas C. Nandy
Advanced Computing and Microelectronics Unit

I N D I A N S T A T I S T I C A L I N S T I T U T E
203, Barrackpore Trunk Road
Kolkata - 700 108

Acknowledgement

At the end of this course, it is my pleasure to thank everyone who has helped me along the way.

First of all, I want to express my sincere gratitude to my supervisor, Prof. Subhas C. Nandy, for introducing me to the world of computational geometry and giving me interesting problems. I have learnt a lot from him. For his patience, for all his advice and encouragement and for the way he helped me to think about problems with a broader perspective, I will always be grateful.

I would like to thank all the professors at ISI who have made my educational life exciting and helped me to gain a better outlook on computer science. I would also like to express my gratitude to Prof. B. P. Sinha, Prof. Arijit Bishnu, Goutam K. Das, Daya Gour for interesting discussions.

I would like to thank everybody at ISI for providing a wonderful atmosphere for pursuing my studies. I thank all my classmates who have made the academic and non-academic experience very delightful. Special thanks to my friends Somindu-di, Debosmita-di, Ishita, Anindita-di, Aritra-da, Butu-da, Koushik, Anisur, Soumyottam, Joydeep and many others who made my campus life so enjoyable. It has been great having them around at all times, good or bad.

My most important acknowledgement goes to my family and friends who have filled my life with happiness; most significantly, to my parents who have always encouraged me to pursue my passions and instilled a love of knowledge in me; to my brother, Debashis who have filled my heart with joy. I am indebted to my friends, Ayan, Nandita, Somrita, Mistu, Sudi, Sumi, Arpi, for their endless supply of encouragement, moral support and entertainment.

Contents

List of Figures

Chapter 1

Introduction

Geometric intersection graphs are intensively studied both for their practical motivations and interesting theoretical properties. Map labelling, frequency allocation in wireless network, resource allocation in line network are some of the areas where geometric intersection graphs play an important role in formulating problems. Here two types of problems are usually considered: (i) characterization problems, and (ii) solving some useful optization problems. In the characterization problem, given an arbitrary graph, one needs to check whether it belongs to the intersection graph of a desired type of objects. The second kind of problem deals with designing efficient algorithm for solving some useful optization problems for an intersection graph of a known type of objects. It is to be noted that several practically useful optization problems, for example, finding the largest clique, minimum vertex cover, maximum independent set, etc. are NP-hard for general graph. There are some problems for which getting an efficient approximation algorithm with good approximation factor is also very difficult. In this area of research, the geometric properties of the intersecting objects are used to design efficient algorithm for these optimization problems. The characterization problem is important in the sense that for the intersection graph of some types of objects, efficient algorithms are sometimes already available for solving the desired optimization problem.

The simplest type of geometric intersection graph is the interval graph, which is obtained by the overlapping information of a set of intervals on a real line. The characterization problem can be easily solved in $O(|V| + |E|)$ time by showing that the graph is chordal and its complementary graph is a comparability graph [Gol04]. All the standard graph-theoretic optimization probelms, for example, finding minimu vertex cover, maximum independent set, largest clique, minimum clique cover, minimum coloring, etc, can be solved in polynomial time for the interval graph [Gol04].

Any graph $G = (V, E)$ can be represented as the intersection graph of a set of axis parallel boxes in some dimension. The boxicity of a graph with n nodes is the minimum dimension d such that the given graph can be represented as an intersection graph of n axis parallel boxes in dimension d. A graph has boxicity at most one if and only if it as an interval graph. Many optimization problems can be solved or approximated more efficiently for graphs with bounded boxicity. For instance, the maximum clique problem for the intersectio graph of axis parallelrectangles in $2D$ can be computed in $O(n \log n)$ time using a plane sweep strategy [NB95].The maximum independent set of rectangle intersection graph is extensively used in map labelling. The maximum independent set for equal height rectangle intersection graph are shown to admit a PTAS. A 2-factor approximation algorithm is very easy to get in $O(n \log n)$ time [AvKS98]. In Chapter 4 we propose that piercing set for bounded height rectangles is fixed parameter tractable.

A graph $G = (V, E)$ is said to be a disk graph if it is obtained from the intersection of a set of disks. Unit disk graphs play important role in formulating different important problems in mobile ad hoc network. In mobile network, the base stations can be viewed as nodes on unit disk graph; the range of each base station is the same. Different problems on this network can be formulated as the graph-theoretic problems on unit disk graph. Recognizing whether an arbitrary graph is unit disk graph is NP-cmplete [BK98]. Maximum clique can be computed in polynomial time for unit disk graph[CCJ90].

In Chapter 3 we propose a PTAS for maximum independent set of unit disk graph. A 3-factor approximation algorithm for minimum clique cover of unit disk graph is also described in that chapter. We also propose a 4-factor approximation algorithm for the minimum piercing set of points for a set of unit disks distributed randomly on the plane. Here the piercing points can be chosen to be any point on the plane. In the *discrete piercing set problem*, a point set P is given. The unit circles are all centered at the points in P. The objective is to choose the minimum set of points in P to pierce all the circles. We propose a 15-factor approximation algorithm for this problem.

Chapter 2

Preliminaries

2.1 Geometric Intersection Graph

Definition 2.1 (Geometric Intersection Graph)**.** The geometric intersection graph $G = (V, E)$ of a set of geometric ob jects S is a graph whose nodes V correspond to the set of objects in S. Between a pair of nodes v_i amd v_j, there is an edge (v_i, v_j) if the corresponding objects in S intersect.

In this defination the tangent objects are assumed to intersect. Now let us formally define Disk Graph:

Definition 2.2 (Disk Graph)**.** A graph G is called Disk Graph if and only if there exists a set of disk $D = \{D_i | i = 1, \ldots, n\}$, such that G is the Intersection Graph of D . The set of disks is called the disk representation of G.

Definition 2.3 (Rectangular Intersection Graph)**.** A graph G is called Rectangular Intersection Graphh if and only if there exists a set of rectangle $R = \{R_i | i = 1, \ldots, n\}$, such that G is the Intersection Graph of R . The set of rectangles is called the rectangular representation of G.

In this thesis, we will consider all the rectangles defining a rectangle intersection graph, are axis parallel. If we take height of all rectangles same then that would be referred to as *unit height axis parallel rectangular intersection graph.* Similarly, we will also assume that all the disks defining a disk graph, are of same radii. Such a disk graph will be referred to as *unit disk graph.*

2.2 Approximation Algorithms

As we focus on proposing approximation algorithms for several useful optimization problems on different geometric intersection graphs, we now define the various types of approximation schemes to be considered in our work.

Definition 2.4 (Approximation Algorithms)**.** Let P be a maximization (resp. minimization) problem. Then an algorithm A is an α -factor approximation algorithm for P if and only if for any instance x of P, $A(x)$ runs in time polynomial in $|X|$ (size of X) and delivers a feasible solution $SOL(X)$, such that $SOL(X) \geq \alpha \times OPT$ (resp. $SOL(X) \leq \alpha \times OPT$). Here OPT denotes the optimum solution of the problem P for the given instance X.

Definition 2.5 (Polynomial Time Approximation Schemes)**.** Let P be a maximization (resp. minimization) problem. An algorithm A is a polynomial-time approximation scheme(PTAS) for P if and only if for any instance X of P and for any (fixed) $\epsilon > 0$, $A(X, \epsilon)$ runs in time polynomial in $|X|$ and delivers a feasible solution $SOL(X, \epsilon)$, such that $SOL(X, \epsilon) \geq (1-\epsilon) \times OPT$ (resp. $SOL(x, \epsilon) \leq (1+\epsilon) \times OPT$).

Definition 2.6 (Fully Polynomial Time Approximation Schemes)**.** Let P be a maximization (resp. minimization) problem. An algorithm A is a fully polynomial-time approximation scheme(FPTAS) for P if and only if for any instance X of P and for a (fixed) $\epsilon < 0$, $A(X, \epsilon)$ runs in time polynomial in $|X|$ and $\frac{1}{\epsilon}$, and delivers a feasible solution $SOL(X, \epsilon)$, such that $SOL(X, \epsilon) \geq (1 - \epsilon) \times OPT$ (resp. $SOL(X, \epsilon) \leq (1 + \epsilon) \times OPT$).

2.3 Fixed Parameter Algorithm

In this thesis, we will also discuss fixed parameter tractable algorithms, as defined below, for some geometric optimization problems.

Definition 2.7 (Parameterized Problem)**.** A parameterized problem is a subset of $\sigma^* \times N$, where σ is a finite alphabet and N is the set of natural numbers. An instance of a parameterized problem is a pair (X, k), where X is the given instance of the problem and k is called the parmeter.

Definition 2.8 (Fixed Parameter Tractable)**.** A problem P is said to be fixed-parameter tractable (FPT) if and only if for any instance (X, k) of P there exists an algorithm that delivers a feasible solution in time $O(f(k)poly(|X|)$, where $f(k)$ is an arbitrary (may be exponential) function on k that does not at all involve $|X|$, and $poly(|X|)$ is a polynomial function in $|X|$.

Thus, if k is assumed to be a constant, then the fixed-parameter tractable problems can be solved in polynomial time.

Chapter 3

Algorithms on Unit Disk Graphs

3.1 Introduction

In this chapter, we focus on proposing approximation algorithms for different optmization problems on unit disk graphs. An unit disk graph is the intersection graph of a set of disks of same radius (see Figure 3.1). We start with the *maximum independent set problem* for the unit disk graph in Section 3.2. In Sections 3.3, 3.4 and 3.5 we give approximation algorithms for *minimum clique cover*, *minimum piercing set* and *minimum discrete piercing set* problems respectively.

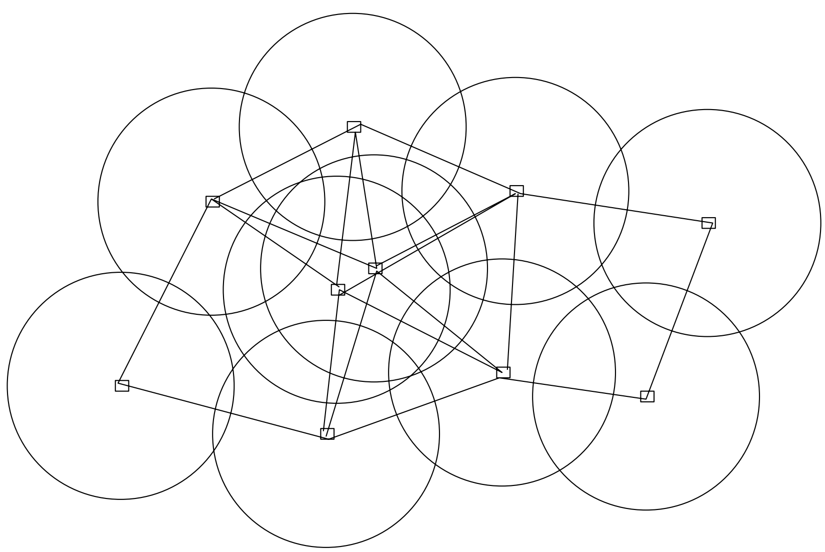

Figure 3.1: Unit disk graph

3.2 Maximum Independent Set for Unit Disk Graph

The problem of finding maximum independent set for unit disk graph is known to be NP-Complete [CCJ90]. So research in this topic is focused on getting an efficient approximation algorithm. In Section 3.2.2, we propose a polynomial time approximation scheme for the maximum independent set problem of unit disk graphs. This improves the currently known best time complexity for the problem. It borrows the idea of the $O(n^4)$ time 2-factor approximation algorithm for unit disk graph appeared in [KNSK]. For the self-completeness of this thesis, we first decribe that algorithm in Section 3.2.1.

3.2.1 A 2-factor Approximation Algorithm

Lemma 3.1. *For a given set D of n unit disks (of radius 1) with centres lying in a horrizantal strip of width 2, the maximum independent set (non-overlapping subset of unit disks of maximum cardinality) can be optimally computed in $O(n^4)$ time.*

Proof. Let D be a set of n unit disks whose centres lie in the horrizantal strip H of width 2. We split the horrizantal strip into 1×1 squares by drawing a horizontal line and a set of vertical lines segments spanning the strip unit distance apart. Next, we delete all the vertical line segments that do not intersect any disk in D. Let us denote the x-coordinates of the remaining vertical line segments as $\mu_1, \mu_2, \ldots, \mu_q$, where $\mu_\alpha \leq \mu_{\alpha+1}$ for all $\alpha = 1, 2, \ldots, q-1$. Let D_α be the set of disks whose centres lie inside the strip H and are intersected by the vertical line at μ_α, i.e., $D_\alpha = \{d = (x, y) | d \in H \text{ and } \mu_{\alpha-1} < x \geq \mu_{\alpha-1}\}$. Now, consider the following two observations.

(i) The size of the maximum independent set among the set of disks D_α may be 1 or 2.

(ii) If we consider a pair of vertical lines at μ_α and μ_β, then there exists no pair of intersecting disks $d \in D_\alpha$ and $d' \in D_\beta$ if $|\mu_\alpha - \mu_\beta| > 1$

We now define a directed graph $G = (V, E)$ whose nodes are the disks in D. The nodes $V = V_1 \cup V_2 \cup \ldots \cup V_q$, where V_α consists of two sets of nodes, namely A_α and B_α. Each unit disk in D_α contributes a node in A_α, and each pair of non-intersecting disks of D_α contribute a node in B_α. A node ϕ_α is also added to V_α for each $\alpha = 1, 2, \ldots, q$. This corresponds to no disk in D_α. The weight of each node in V_α is equal to the number of disks it represents, Thus, ϕ_α is assigned a weight equal to 0. The weights of the node in A_α and B_α are 1 or 2 respectively. Next, we define the edges E of the graph G.

- No two nodes in V_α are connected.
- There is a directed edge from ϕ_α to each node in $V_{\alpha+1}$.
- Each pair of vertices $u \in V_\alpha$ and $v \in V_{\alpha+1}$ are connected by a directed edge (u, v) if disk(s) in node u do not intersect the disk(s) in node v.
- There is no need to add edges (u, v) where $u \in V_\alpha$ and $v \in V_\beta$, and $\beta - \alpha > 1$. The reason is that the vertex v is reachable from u via a directed path of weight 0 through ϕ marked vertices of the sets $V_{\alpha+1}, V_{\alpha+2}, \ldots, V_{\beta-1}$.

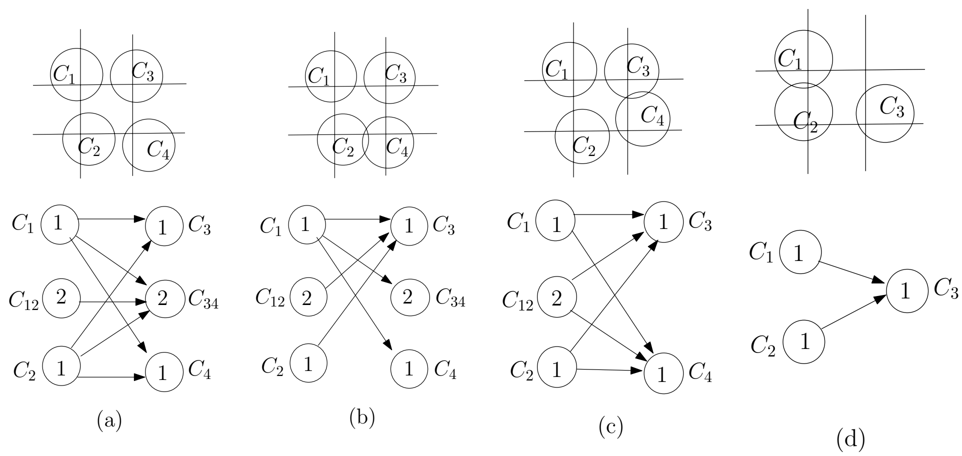

Figure 3.2: Assignment of edges among the nodes in V_α and $V_{\alpha+1}$

Figure 3.2 illustrates the edges among the nodes in two set V_α and $V_{\alpha+1}$ in the graph G. The graph G, defined thus, is a q-partite graph, where q is the number of vertical lines in the strip H. Finally, a source node s and a sink node t are added. The node s is connected with each node in V_1 by directed edge. Similarly, the nodes in V_q are connected to a sink node t by directed edges. Both the nodes s and t are assigned with weight 0. As the number of centres in the strip H is n, so $|V| = O(n^2)$ and $|E| = O(n^4)$ in the worst case.
The maximum independent set of unit disks in the strip H corresponds to the longest path in the directed acyclic graph G, and it can be found in time $O(|E|)$ [CLR07], which is $O(n^4)$ in our case. □

Theorem 3.2. *Given a set C of n unit disks in a 2D plane, a subset of C of size at least $\frac{1}{2}OPT$ of non-intersecting disks can be obtained in $O(n^4)$ time, where OPT is the maximum number of mutually non-intersecting disks pesent in the set C [KNSK].*

Proof. We draw a set of horizontal lines $l_1, l_2, l_3, ..., l_{m+1}$ such that separation between each pair of consecutive lines is equal to the diameter of unit disks.
These lines partition the set C into subsets $C_1, C_2, ..., C_m$, where C_i is the set of unit disks in C whose centres lie between the strip bounded by the horizontal lines l_i and l_{i+1}.
For each set C_i, we can compute the maximum independent set I_i using Lemma 3.1 in $O(n_i^4)$ time where $n_i = |C_i|$. This is to be noted that $C_i \cap C_k = \phi$ if $|i - k| > 1$, but $C_i \cap C_{i+1}$ may not be empty. Thus, the members in $IS_{odd} = I_1 \cup I_3 \cup \ldots$ are all non-intersecting, and also $IS_{even} = I_2 \cup I_4 \cup \ldots$ are non-intersecting. We report the independent set $IS = max\{|IS_{odd}|, |IS_{even}|\}$. It is to be noted that $|IS_{odd}| + |IS_{even}| \geq OPT$. Thus $2 \times max\{|IS_{odd}|, |IS_{even}|\} \geq OPT$. Thus, the size of the reported answer IS is greater than $\frac{1}{2}OPT$.
The total time required for computing IS_{odd} or $IS_{even}|$ is $O(n^4)$, where n is the number of disks in the set C. Thus the time complexity result follows. □

3.2.2 PTAS

In this section, we first explain the problem of optimally solving the maximum independent set for a set of unit disks C (of radii 1) whose centres lie within a pair of parallel lines of a distance k apart, where k is a positive integer. The case $k = 1$ is handled in Section 3.2.1. Here we consider the case where $k > 1$. Next, we use this result to propose a PTAS.

Lemma 3.3. *The maximum independent set for a set of unit disks C whose centres lie inside a strip of width k can be computed in $O(kn^{4k})$ time using $O(n^{4k})$ space.*

Proof. We split the region into k horizantal strips $H_1, H_2, \ldots, H_k$, each of width 1, using $k + 1$ horizantal lines, and consider a set of vertical lines at unit distance apart. We denote the x-coordinate of the vertical lines by $\mu_1, \mu_2, \ldots, \mu_q$ after eliminating those vertical lines that do not intersect any members in C. Let $C_{\alpha j} = \{c_1, c_2, \ldots c_{\nu_{\alpha j}}\}$ be the set of unit disks whose centres lie inside the strip H_j and are intersected by the vertical line at μ_α. It is to be noted that in any independent set of C, at most two members of $C_{\alpha j}$ can be present. For each strip $j = 1, 2, \ldots, k$, we form a set ${C_{\alpha j}}^\star = \{\phi\} \bigcup \{c_i, i = 1, 2, \ldots, \nu_{\alpha j}\} \bigcup \{(c_i, c_l), i = 1, 2, \ldots, \nu_{\alpha j}, l = i + 1, \ldots \nu_{\alpha j},\ c_i$ and c_j are not intersecting$\}$. The numbers of elements in this set is $1 + \frac{\nu_{\alpha j}(\nu_{\alpha j}+1)}{2} = O({\nu_{\alpha j}}^2)$.
Next, we construct a node-weighted q-partite digraph $G = (V_1 \cup V_2 \cup \ldots \cup V_q, E)$, where the set of nodes V_α correspond to the vertical line at μ_α, and its members are formed as follows:

Consider all possible k-tuples with one element from each of the k sets $C_{\alpha j}{}^{\star}, j = 1, 2, \ldots, k$. There is a node in V_α for a particular k-tuple if and only if the disks corresponding to the elements in that k-tuple are mutually non-intersecting. The weight of a node is the number of unit disks it represents.

Between a pair of vertices $u \in V_\alpha$and $v \in V_{\alpha+1}$, there is a directed edge(u, v) if the disks corresponding to u and v do not intesect. As in Lemma 3.1, here also we do not have to consider edges (u, v), where $u \in V_\alpha$ and $v \in V_\beta$, where $\beta - \alpha > 1$. The digraph G is acyclic and q-partite, and the maximum weight path in G corresponds to the largest set of disks in C that are mutually non-overlapping.

The number of nodes in V_α is the number of k-tuples formed by the elements in $C_{\alpha j}, j = 1, 2, \ldots, k$. Since $|C_{\alpha j}|$ can be at most $O({\nu_{\alpha j}}^2)$, we have

$$|V_\alpha| = O(\textstyle\prod_{i=1}^{k} {\nu_{\alpha j}}^2) \leq O\left(\left(\frac{1}{k}\sum_{i=1}^{k} {\nu_{\alpha j}}^2\right)^k\right) \leq O({\nu_\alpha}^2),$$

where $\nu_\alpha = \sum_{j=1}^{k} \nu_{\alpha j}$ is the number of disks stabbed by the vertical line at μ_α. The time for creating a node in V_α is $O(k)$ since each element (disk) of the corresponding k-tuple neeeds to be checked with at most 4 disks in the same k-tuple with centres lying in its neighbouring strips. This, the time for computing the nodes in V_α is $O(k{\nu_\alpha}^{2k})$. Again, since the disks participating in the formation of nodes in V_α and $V_\beta(\alpha \neq \beta)$ are disjoint and $\sum_{\alpha=1}^{q} = n$, the total number of nodes in the graph G is $O(\sum_{\alpha=1}^{q} {\nu_\alpha}^{2k}) = O(n^{2k})$. The number of edges $|E| = O(n^{4k})$. Deciding whether a pair of vertices in two sets $u \in V_\alpha$ and $w \in V_{\alpha+1}$ form an edge (u, w) may take $O(k)$ time, since each node of u needs to be checked with at most 6 disks of node w (in the same and adjacent layer) for possible intersection. Thus the computation of all the edges in the graph G needs $O(kn^{4k})$ time in the worst case. The maximum weighted path in the graph G can be obtained in $O(|E|)$ time. The space complexity follows from the number of edges in G.

□

Theorem 3.4. *For a given integer $k \geq 1$, one can obtain an $(1 - \frac{1}{k+1})$-factor approximation algorithm for finding maximum independent set for unit disk graph in $O(k^2 n^{4k})$ time using $O(n^{4k})$ space.*

Proof. As in Theorem 3.2, here also we split the whole region into strips $H_1, H_2, \ldots,$ H_m, in top to bottom order by drawing horrizontal lines 1 distance apart.Let C_i denotes the set of disks whose centres lie in the strip H_i, and $C = \bigcup_{i=1}^{m} C_i$. Let $C_i^j = C_i \cup C_{i+1} \cup \ldots \cup C_{i+j+1}; C_i^0 = \phi$. We use IS_i^j to denote the maximum independent set of the set of unit disks C_i^j. We now form k distinct maximum independent set problems, namely $MIS_j, j = 1, 2, \ldots, k$, where each of the problem MIS_j denotes the problem of finding the the maximum independent set for the unit disks ${C_1}^{j-1} \cup \bigcup_{i \geq 0} {C_{i(k+1)+j+1}}^k = C \setminus \bigcup_{i \geq 0} C_{i(k+1)+j}$, lying within a strip of width k. It is to be noted that the subset of unit disks ${C_{i(k+1)+j}}^k \cap {C_{i'(k+1)+j}}^k = \phi$, for $i \neq i'$, and also the subset ${C_1}^{j-1} \cap {C_{i(k+1)+j+1}}^k = \phi$, for each $j = 1, 2, \ldots, k$. By Lemma 3.3, we can optimally compute ${IS_1}^{j-1}$, and ${IS_{i(k+1)+j}}^k$ for all $i \geq 1$, and then by concatenating the solutions we can get the optimum solution IS_j for the problem MIS_j.

Finally, we report $max_{j=1}^k IS_j$.

Then from the *shifting lemma* of Hochhbaum and Maass [nWM85] we get $(1 - \frac{1}{k+1})$-factor approximation algorithm. Time complexity result follows from the fact that the time for solving MIS_j is $O(kn^{4k})$ for each $j = 1, 2, \ldots, k$, and we need to solve all the problems $MIS_j, j = 1, 2, \ldots, k$.

□

3.3 Minimum Clique Cover for Unit Disk Graph

We have a unit disk graph $G = (V, E)$, where the set of nodes V corresponds to a set of unit disks placed on a 2D plane; an edge between a pair of vertices implies that the corresponding two disks mutually intersect. The disk-layout of the graph G is given. Our objective is to identify minimum number of cliques that can cover all the nodes in the graph G. Cerioli et al. [CFFF04] proved that the problem is NP-Complete, and proposed a 3-approximation algorithm for this problem. Very recently Dumitrescu and Pach [DP09] proposed a randomized algorithm that produces solution with approximation ratio 2.16. We also propose a 3-factor approximation algorithm for this problem using the shifting paradigm as used in Subsection 3.2.1.

Lemma 3.5. *If the centres of the disks of radius* 1 *lie inside a strip bounded by a pair of parallel lines at a distance* 1 *then the unit disk graph becomes a co-comparability graph.*

Proof. We have to show that If the centres of the disks of radius 1 lie inside a strip bounded by a pair of parallel lines at a distance 1 then the complement of the unit disk graph becomes a comparability graph. So we will give edge between two disks if they are not connected. And we will have to show that in this graph all the edges are transitively oriented. To show that transitive orientation exist , we give the directed edge from disk a to disk b if disk a and disk b are not connected and disk a lies left to the disk b.

Let a, b, c be three disks of radius 1 lying left to right inside a strip bounded by a pair of parallel straight lines at a distance 1 and. We will have to prove that if disk a and disk b do not intersect, disk b and disk c do not intersect, then disk a and disk c also do not intersect.

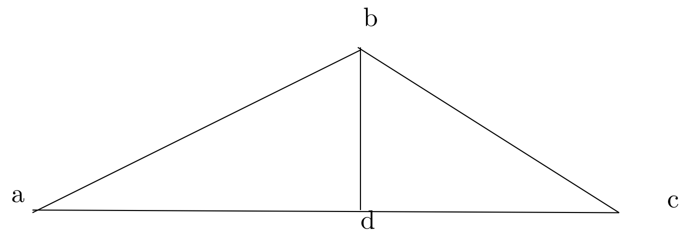

Figure 3.3: Demonstration for the proof of co-comparability graph

In Figure 3.3, let $dist(a, b) = 2 + \epsilon_1$ and $dist(b, c) = 2 + \epsilon_2$. Let $[b, d]$ be the perpendicular on $[a, c]$. As a, b, c lie inside a strip bounded by a pair of parallel straight lines at a distance 1, let $dist(b, d) = 1 - \epsilon_3$. Let $dist(a, d) = x_1$ and $dist(d, c) = x_2$, we need to prove that $x_1 + x_2 > 2$.
From traingular inequality, we get $dist(a, d) + dist(b, d) \geq dist(a, b)$
i.e, $x_1 + 1 - \epsilon_3 \geq 2 + \epsilon_1$
i.e, $x_1 > 1$
Similarly, we get $x_2 > 1$, which leads $x_1 + x_2 > 2$.

□

Lemma 3.6. *If the centres of the disks of radius* 1 *lie inside a strip bounded by a pair of parallel lines at a distance* 1 *then minimum clique cover can be found in* $O(n^2)$ *time.*

Proof. By Lemma 3.5, the graph G is a co-comparability graph. We consider the complemetary graph $\overline{G} = V, \overline{E}$, where a pair of vertices $u, v \in V$ are connected in $\overline{E}$ if the edge $(u, v) \notin E$, and a pair of vertices $u, v \in V$ are not connected in $\overline{E}$ if the edge $(u, v) \in E$. The graph $\overline{G}$ is a comparability graph, and its maximum sized clique can be found optimally in $O(|\overline{V}| + |\overline{E}|)$ time [Gol04]. The size of the minimum clique cover in G is the same as the size of the maximum sized clique in $\overline{G}$. Hence the lemma is true. □

Now, we will prove the main result in this section.

Theorem 3.7. *For the problem of finding mimimum clique cover for unit disk graph, a 3-factor approximation solution can be obtained in* $O(n^2)$ *time.*

Proof. Let D be the disk layout corresponding to the unit disk graph G. We draw a set of horizontal lines $l_1, l_2, l_3, ..., l_{m+1}$ such that separation between each pair of consecutive lines is equal to the common radius of disks.

These lines partition the set D into subsets $D_1, D_2, ..., D_m$, where D_i is the set of unit disks in D whose centres lie in the strip bounded by the lines l_i and l_{i+1}.

We can compute the minimum clique cover c_i for each set D_i using Lemma 3.6 in $O(n_i^2)$ time where $n_i = |D_i|$.

The disks D_i may intersect some disk(s) in D_{i-1}, D_{i-2} and D_{i+1}, D_{i+2}, but they never intersect any disk in $D \backslash \{D_i \cup D_{i-1} \cup D_{i-2} \cup D_{i+1} \cup D_{i+2}\}$. So, we consider three disjoint sets of disks $D_1 \cup D_4 \cup D_7 \cup \ldots$, $D_2 \cup D_5 \cup D_8 \cup \ldots$ and $D_3 \cup D_6 \cup D_9 \cup \ldots$ and compute their clique covers C_1, C_2 and C_3 separately. Clearly $C_1 \cup C_2 \cup C_3$ will be a clique cover of D. If C_{opt} is the minimum clique cover then $|C_{opt}| \geq \max\{|C_1|, |C_2|, |C_3|\}$. Thus, $|C| + |C_2| + |C_3| \leq 3|C_{opt}|$. Thus, we have a 3-factor approximation algorithm for computing the minimum clique cover of an unit disk graph.

Since the algorithm runs in each strip independently, and there is no common disk in any pair of strips, the total time complexity of the algorithm is obtained by adding the time complexities of running the algorithm in each strip. Thus, the time complexity of the proposed algorithm is $O({n_1}^2 + {n_2}^2 + \ldots) = O(n^2)$, where $n_i = |D_i|$, for $i = 1, 2, \ldots, m$, and $\sum_{i=1}^{m} n_i = n$. □

3.4 Minimum Piercing Set for a set of Unit Disks in the plane

Here, we focus on the geometric version of the mimimum clique cover problem for the unit disk graph. As Helly property does not hold for a set of disks, there may exist a clique in the unit disk graph, but the corresponding disks may not have any common region as shown in Figure 3.4(a). Thus, unlike the rectangle intersection graph, the problem of finding minimum piercing set for a set of unit disks is not the same as finding minimum clique cover in the unit disk graph. We can get a 4-factor approximation algorithm for *unit disks* as follows.

Theorem 3.8. *For the problem of finding mimimum piercing set for a set of unit disks in the plane, a 4-factor approximation algorithm can be computed.*

Proof. We split the region using horizontal lines such that every pair of consecutive lines are at 2 units apart. Let k be the number of horizontal lines needed to split the given region R. This ensures that each disk will be intersected by exactly one line. For each strip, we compute the minimum size set of piercing points for the set

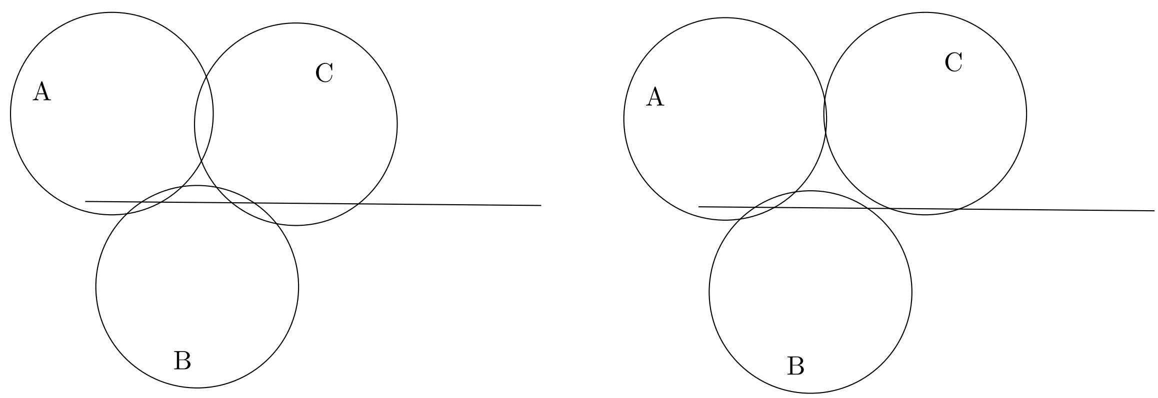

Figure 3.4: Demonstration of our algorithm for piercing set of unit disks

of unit disks such that a portion of each of these disks lies inside that strip. We present a 2-factor approximation algorithm for this problem.

Let C be the set of circles intersected by a horizontal line. We consider the circle $c \in C$ having center with minimum x-coordinate, and all the circles $C_1 \subseteq C$ that intersect it. If we consider the arrangement of circles in $C_1 \cup \{c\}$, we may have to put at most two piercing points to pierce the circles in $C_1 \cup \{c\}$ (see Figure 3.4). In the optimal piercing set, one of these clique points must be chosen for piercing the circle c. In our algorithm, we select both these piercing points, and discard all the circles in $C_1 \cup \{c\}$. We repeat the same steps until all the members in C are exhausted. Thus we have a 2-factor approximation algorithm for piercing the members in C.

Let the optimum piercing set for the unit disks intersected by the i-th horizontal line by K_i. Thus, $\cup_{i=1}^{k} K_i$ is a 2-factor approximation of the optimum solution of the minimum piercing set. But, we could not compute K_i, instead we computed a piercing set K_i' such that $|K_i'| \leq 2|K_i|$. So, $\cup_{i=1}^{k} K_i'$ is a 4-factor approximation solution for the minimum piercing set problem. □

3.5 Minimum Discrete Piercing Set for Unit Disk Graph

A set of point P is given in a plane. The *minimum discrete piercing set* problem for unit disks is to select minimum number of points that pierces (lies inside) all the unit disks centered at the points in P. No polynomial time algorithm for this problem is known to date for this problem. We will propose an approximation algorithm for this problem. The approximation factor of our algorithm is 15.

Observation 3.1. The points which will be covered by putting a disk (of radius 1 unit) at a point are inside that disk.

Observation 3.2. Unit Disk at a point will be intersected by those disks whose centres lie inside a circle of radius 2 unit.

3.5.1 Algorithm

We choose the disk L having leftmost center. We need to cover this disks, and the best possible performance would be to cover all the disks that intersect L by a single point. But, this may not always be feasible. In the following lemma, we show that we may need at most 15 points to cover L and all other disks that intersect L.

Lemma 3.9. *All the disks intersected by the disk at left most point can be covered by at most 15 disks.*

Proof. Let L be the disk having left most center o (see the half-disk having smaller radius (1 unit) in Figure 3.5(a)). All the disks having center inside L must contain the point o. Thus, if we choose o, we can cover all these circles. But, there area several other circles that may also intersect L. These have center inside the half-circle having radius 2 in Figure 3.5(a). It needs to be observed that if we consider a rectangle of size $3\sqrt{2} \times \frac{3}{\sqrt{2}}$ with o at the middle of the leftmost vertical line as in Figure 3.5(b), and divide the region into 18 equal parts each of size $\frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}}$, then the outer circle does not enter into the squares C and D. Also the squares A and B are completely inside of the inner circle. Thus the annulus R can be covered by 14 squares each of size $\frac{1}{\sqrt{2}} \times \frac{1}{\sqrt{2}}$. Since the size of the diagonal of each square is equal to 1, the maximum distance between any two points inside the square, so choosing a single point inside each square at the center, one can cover all the disks centered inside that square. □

Now, we have the main theorem.

Theorem 3.10. *A 15-factor approximation algorithm for the problem of minimum discrete piercing set for unit disks centered on a given set of points can be computed in $O(n^2)$ time.*

Proof. We sort all the points by their x-coordinates. Take the left most point (here the point is o), i.e, whose x-coordinate value is least. By Lemma 3.9, all the disks that intersect the disk L centered at o can be covered by at most 15 points. We remove L, and all the disks that intersect L. As the other disks do not intersect L,

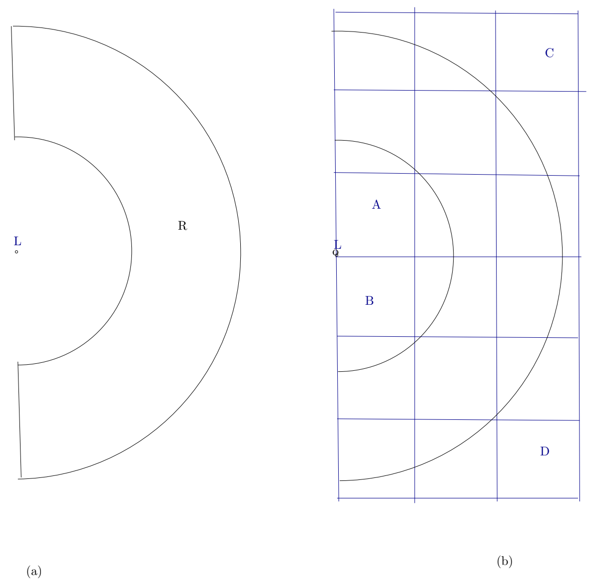

Figure 3.5: Demonstration of our algorithm for Discrete Piercing Set of Unit Disk Graph

there is no chance to cover any of them with L using a single point. We repeat the same process. Thus, the approximation factor of the result is justified.
Choosing left-most point needs $O(n)$ time. Removing o and the other points inside the larger half-circle needs another $O(n)$ time in the worst case. Thus, the time comlexity is justified.

□

Chapter 4

Algorithms on Rectangle Intersection Graph

In this chapter, we consider the minimum clique cover problem for the rectangular intersection graph of unit height axis-parallel rectangles (see Figure 4.1). Note that, the length of the rectangles can be arbitrary. In Section 4.1, we describe a 2-factor approximation algorithm for the unit-height axis-parallel rectangle intersection graph. In section 4.2 we propose a fixed-parameter tractable algorithm for computing the decision version of the piercing set problem for bounded height rectangles in grid.

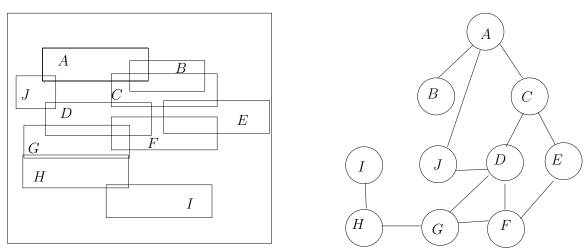

Figure 4.1: A set of unit height axis-parallel rectangle, and the corresponding rectangle intersection graph

4.1 Minimum Piercing Set for Unit Height Rectangles

We have a set of axis parallel rectangles in the plane. Our objective is to find minimum piercing set for these rectangles. The problem is NP-hard. So we try to design an efficient algorithm that is guaranteed to produce solution close to the optimum size. We now show that we can have a 2-factor approximation algorithm for the problem.
Let R be a set of n unit-height rectangles in the plane. We draw a set of horizontal lines $l_1, l_2, l_3, ..., l_m$, satisfying the following properties

(i) separation between each pair of consecutive lines is more than 1,

(ii) each line intersects at least one rectangle, and

(iii) each rectangle is intersected by some line.

The minimum separation condition implies that a rectangle cannot be intersected by more than one line. These lines can be drawn in an incremental approach. These lines partition the set R into subsets of rectangles $R_1, R_2, ..., R_m$, where R_i is the set of rectangles in R that are intersected by the line l_i.
We compute the minimum clique cover C_i for each set R_i using the method of optimally computing the minimum clique cover for an interval graph [Gol04].
The rectangles in R_i may intersect some rectangle(s) in R_{i-1} and R_{i+1}, but they never intersect any rectangle in $R \setminus \{R_i \cup R_{i-1} \cup R_{i+1}$.
So, we consider two disjoint sets of rectangles $R_1 \cup R_3 \cup R_5 \cup \ldots$ and $R_2 \cup R_4 \cup R_6 \cup \ldots$, and compute their clique covers C and C'. Clearly $C \cup C'$ will be a clique cover. If C_{opt} is the minimum clique cover then $|C_{opt}| \geq \max\{|C|, |C'|\}$. Thus, $|C| + |C'| \leq 2|C_{opt}|$. Thus, we have the following theorem.
The time complexity of this algorithm is $O(n \log n)$, since getting the set of rectangles in one of the groups, say $R_1 \cup R_3 \cup R_5 \cup \ldots$, is $O(n)$. Computing the minimum clique cover of the interval graph needs $O(n \log n)$ time. Thus, we have the following theorem.

Theorem 4.1. *For the problem of finding mimimum piercing set for unit height rectangle intersection graph, a 2-factor approximation algorithm can be computed in $O(n \log n)$ time.*

4.2 Piercing Set for Bounded Height Rectangles in Grid

Let us consider a set of rectangles R in a grid of unit 1, i.e, the boundaries of all the rectangles are along the grid lines. We also assume that the height of each rectangle is same, and is qual to b. We have to pierce all the rectangles with minimum number of points. We now consider the decision version of the problem, i.e., can all the rectangle be pierced by k points ? We proceed as follows:

We will consider a rectangle ρ whose right side is the left-most among all the rectangles in R. There is at most b possible points which are topologically equivalent with respect to piercing the rectangle ρ. Let these be the points of intersection of the right boundary of ρ and the b horizontal lines of the grid. We consider a tree, whose root nodes have at most b children corresponding to each of these points. We explore each child of the root. While considering a child of the root, it is considered as the root of a subtree. We delete ρ and all the rectangles that are overlapped on ρ. Again, choose a rectangle having left-most right boundary among the remaining rectangles. The search proceeds in a depth first manner. After exploring k levels in a path of the tree, if all the rectangles are not pierced, then the chosen piercing points along this path can not pierce all the rectangles in R. We need not have to proceed further. We return the set of rectangles deleted for progressing in the current node, and backtrack.

If any of these paths show a complete piercing of the set of rectangles in R, we return an *affirmative* answer, otherwise, the answer is *negative*.

As the depth of the tree is bounded by k, and each node of the tree has at most b children, size of the tree is bounded by b^{k+1}.

For each node we need only to remove all the rectangles pierced by the chosen point. This will take some $O(n)$ time. So our proposed algorithm takes $O(b^{k+1}n)$ time in the worst case. Thus, we have the following theorem:

Theorem 4.2. *The problem of finding mimimum piercing set for bounded height rectangles in a grid is fixed-parameter tractable.*

Bibliography

[AvKS98] P. K. Agarwal, M. van Kreveld, and S. Suri. Label placement by maximum independent set in rectgles. *Computational Geometry: Theory and Application*, 11:209–218, 1998.

[BK98] H. Breu and D. G. Kirkpatrick. Unit disk graph recognition is np-hard. *Computational Geometry: Theory and Application*, 9:3–24, 1998.

[CCJ90] B.N. Clark, C.J. Colbourn, and D. S. Johnson. Unit disk graph. *Discrete Mathematics*, 86:165–177, 1990.

[CFFF04] 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.

[CLR07] T.H. Cormen, C. E. Lieserson, and R. L. Rivest. *Introduction to Algorithm*. Prentice Hall, 2007.

[DP09] Adrian Dumitrescu and Janos Pach. Minimum clique partition in unit disk graphs, 2009.

[Gol04] Martin Charles Golumbic. *Algorithmic Graph Theory and Perfect Graph*. Annals of Discrete Mathematics, 2 edition, 2004.

[KNSK] S. Kolay, S. C. Nandy, and S. Sur-Kolay. 2-factor approximation algorithm for computing maximum independent set of a unit disk graph.

[NB95] S. C. Nandy and B. B. Bhattacharya. A unified algorithm for finding maximum and minimu point enclosing rectangles and cuboids. *Computers and Mathematics with Application*, 29(8):45–61, 1995.

[nWM85] D. S. Hochbaum nad W. Maass. Approximation schemes for covering and packing problems in image processing and vlsi. *Journal of the ACM*, 32:130–136, 1985.