

# Approximation Algorithms for Disk Graphs

Arman Boyacı

December 17, 2008

## 1 Introduction

A disk graph(DG) is an intersection of set of disk graphs in the Euclidean plane. There are many applications that use disk graph modeling to represent the corresponding problem. Different types of graph problems are solved in these applications. For example, in chemistry testing the physical feasibility of a graph modeling molecular (recognition problem), in telecommunication, making frequency assignments for the transmitter/receiver stations (coloring problem), also selecting a minimum number of transmitters so that all other stations are within the range one of the chosen transmitters (domination problem), in facility location, placing facilities to given locations where the proximity of facilities is undesirable (independent set). With the high increasing demand in GSM telecommunication, the frequency assignment problem became notably important. This area of research is well studied in the literature. Several surveys in this topic are [Hale, 1980, Aardal et al., 2001].

In this subclass of graphs, most of the graph problems remain unfortunately NP-hard. In other words, there is no known polynomial time algorithm to solve problems optimally in this domain. However, since there exist enough motivation to study these graphs, approximation algorithms are suggested to generate relatively good solutions in a reasonable time.

Three equivalent models can be defined to represent problems. In *intersection model*, each vertex corresponds to a disk on a plane and an edge appears between two vertices when the corresponding disks intersect. We assume that tangent disks intersect. In *containment model*, n disks in the plane form a graph with n vertices corresponding to n disks and an edge between two vertices if one of the corresponding disks contains the other's center. Finally, a purely geometric definition is also available. N points in the plane form a graph with n vertices corresponding to n points and an edge between two vertices if and only if the Euclidean distance between two corresponding points is at most some specified bound d. This last model referred as *proximity model*. Transforming between intersection and containment models can be easily done by simply doubling or halving the diameter of disks. Transforming between the intersection and proximity models requires only an identification of the disks centers with the points in the plane and the disk diameter with d. Therefore given any of the three models, the other two can be produced in linear time.

Relevant subclasses of DG are the followings: coin graphs (CG),  $\sigma$ -disk graphs ( $\sigma$ -DG), unit disk graphs (UDG). In CG, disk are not allowed to overlap but to touch. In  $\sigma$ -DG, diameter ratio of disks is bounded by some constant  $\sigma$ . A disk graph is called unit disk graph if all disks' diameters are equal. A grid graph is a unit disk graph in whose intersection model all disks have centers with integer coordinates and radius  $\frac{1}{2}$ .

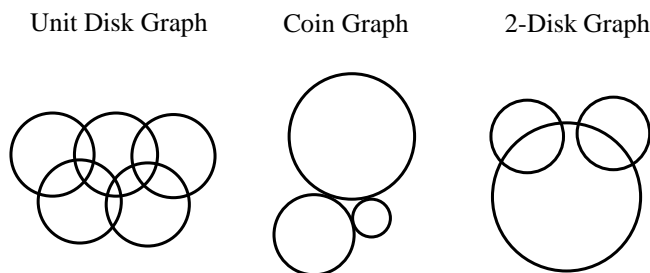


Figure 1: Subclasses of Disk Graphs

Obviously given a disk graph, there is no unique way to place disks on the plane. In some applications, coordinates of disks could be available. This additional information can allow us to develop better algorithms. Given a disk graph with the coordinates of disks will be referred as a graph with given representation. All of our analysis will be done for both cases. In other words, complexity results as well as the performance of algorithms will be analyzed on graphs with and without representation.

This report is structured in the following way. First, we will enumerate some important properties of disk graphs. These properties will provide us some tools to make our analysis for the approximation algorithms. Secondly, we will discuss about complexity of some graph theoretical problems. We will show that almost all problems are NP-hard in DG. Therefore, next section will cover approximations algorithms. Starting by defining what an approximation algorithm is, we will state approximation algorithms for various problems in DG. Moreover, we will proof the performance guarantee of these algorithms. Finally, we will finish by mentioning about some open questions on this topic.

## 1.1 Characteristics

In the beginning, we will do some basic observations that we will make use of in our analysis. Therefore, in this section we will specify important properties of DG. [Marathe et al., 1995, see]

**Property 1.1.** *Unit disk graphs are not perfect.*

*Proof.* Any odd cycle of length of five or greater is a UDG. □

**Property 1.2.** *Unit disk graphs are not planar.*

*Proof.* Any clique size of five or greater is a UDG. □

**Property 1.3.** *Coin graphs are planar.*

Obviously these properties are true also for DG. It means that DG are not planar nor perfect. Therefore, unfortunately none of the polynomial time algorithms for perfect graphs will be available for UDG and clearly for DG.

**Property 1.4.** *Let  $C$  be a disk of radius  $r$  and and let  $S$  be a set disks of radius  $r$  such that every disk in  $S$  intersects  $C$  and no two disks in  $S$  intersect each other. Then  $\text{card}(S) \leq 5$ .*

*Proof.* Suppose  $|S| \geq 6$ . Let  $s_i, 1 \leq i \leq 6$ , denote the centers of any six disks in  $S$ . Let  $c$  denote the center of  $C$ . Denote the ray  $\overrightarrow{cs_i}$  by  $r_i$  ( $1 \leq i \leq 6$ ). Since there are six rays emanating from  $c$ , there must at least on pair of rays  $r_j$  and  $r_k$  such that the angle between them is at most  $\frac{\pi}{3}$ . Now, it can be verified that the distance between  $s_j$  and  $s_k$  is at most  $2r$ , which implies that disks centered at  $s_j$  and  $s_k$  intersect, contradicting our assumption. □

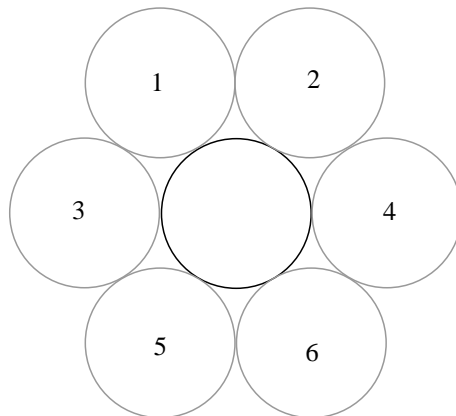


Figure 2: An unit disk having six neighborhood disks

Problem	Complexity
Recognition	NP-hard
Independent Set	NP-hard
Dominating Set	NP-hard
Clique	Polynomial
Coloring	NP-hard

Table 1: Complexity results for Unit Disk Graphs

**Property 1.5.** *Let  $G$  be a unit disk graph, and let  $v$  be a vertex such that the unit disk corresponding to  $v$  (in some model for  $G$ ) has the smallest  $X$ -coordinate. The size of a maximum independent set in  $G(N(v))$  is at most 3.*

Property 1.5 is a natural extension of property 1.4. It can be easily seen on figure 2 that if a unit disk has a smallest  $x$ -coordinate, it cannot have more than 3 independent neighborhood.

**Property 1.6.** *Every induced subgraph of a unit disk graph is also a unit disk graph.*

Property 1.5 and 1.6 will be frequently used in analysis of approximation algorithms in following sections. An example of a graph that is not a unit disk graph is the star  $K_{1,6}$  with one central node connected to six leaves: if each of six unit disks touches a common unit disk, some two of the six disks must touch each other.

## 2 Complexity Results

Most of the graph problems remain NP-hard in unit disk graphs. Table 1 presents a summary of complexity results in this domain. All the problems stated except maximum clique problem is also NP-hard problem. There is no known result for the maximum clique problem in DG in general. In this section, we will give the sketch of the proofs of some NP-hard problems and the polynomial time algorithm for the maximum clique problem.

### 2.1 Recognition Problem

UDG recognition problem can be stated as given a graph  $G$  determining whether  $G$  is a unit disk graph or not. Breu and Kirkpatrick showed that this problem is NP-hard by reducing satisfiability problem to it. A graph that simulates SATISFIABILITY is constructed in the following way: Given an instance of SATIFIABILITY problem  $C$ , the vertices of the graph correspond the the clauses, variables, and negated variables of  $C$  and there is an edge between a literal vertex and a clause vertex if the literal appears in the clause. The relatively long proof of the reduction can be found in [Breu and Kirkpatrick, 1998].

### 2.2 Maximum Clique Problem

A clique of graph  $G=(V,E)$  is vertex subset  $V' \subseteq V$  such that for any pair of vertices in  $V'$ , there is an edge between them in  $E$ . The clique number  $\omega(G)$  is defined as the size of a maximum clique in  $G$ .

**Theorem 2.1.** [Clark et al., 1990] *There is a polynomial algorithm for the maximum clique problem in unit disk graphs with representation.*

*Proof.* We are given unit disks in the plane, consider two unit disk  $A$  and  $B$  which intersect. Let  $O_A$  and  $O_B$  be the centers of disks  $x$  and  $y$ , respectively. Since  $x$  and  $y$  intersect,  $d(O_A, O_B) \leq d$  where  $d$  is the diameter of a disk. We define  $R_{AB}$  denoting the intersection of two closed disk of radius  $d(O_A, O_B)$ , see figure 3. Let  $H_{AB}$  be the induced subgraph defined as follows  $H_{AB} = R_{AB} \cap V$ . We can partition  $R_{AB}$  into two partition  $R_{AB}^1$  and  $R_{AB}^0$  as seen in figure 3. We can observe that every vertex in one of the partition ( $R_{AB}^1$  or  $R_{AB}^0$ ) intersect with each other and possibly vertices on the other partition. Therefore,  $R_{AB}$  is a complement of

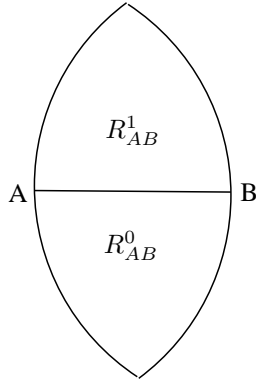


Figure 3: The intersection of two closed disk of radius  $d(O_A, O_B)$

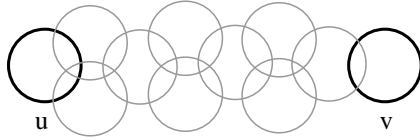


Figure 4: Transformation for the reduction from Planar Graphs 3-colorability to UDG 3-colorability

a bipartite graph. We know that the maximum independent set problem in a bipartite graph can be found in time  $O(|V|^{2.5})$ . It is also true that the maximum clique problem in  $G$  is equivalent the maximum clique problem in  $\overline{G}$ . So, we can find a maximum clique in  $R_{AB}$  in polynomial time. Finally we can find maximum clique in the whole graph by looking all pairs which intersect.  $\square$

### 2.3 Minimum Vertex Coloring

A coloring of a graph  $G=(V,E)$  is an assignment of colors to each vertex in  $V$  such that no edge in  $E$  connects two identically colored vertices. The chromatic number  $\chi(G)$  is defined as the minimum number of color used for coloring all the graph.

**Theorem 2.2.** [Clark et al., 1990] *Minimum vertex coloring problem in UDG is NP-hard.*

*Proof.* It is already shown that planar graph 3-colorability is NP-complete. We will reduce this problem to our case UDG 3-colorability. We transform a planar graph  $G = (V, E)$  with maximum degree 4 into a unit disk graph  $G'$  such that  $G$  is 3-colorable if and only if  $G'$  is 3-colorable.  $G'$  is constructed by making use of the following result.

**Lemma 2.3.** [Valiant, 1981] *A planar graph  $G$  with maximum degree 4 can be embedded in the plane using  $O(|V|)$  area such a way that its vertices are at integer coordinates and its edges are drawn so that they are made up of line segments of the form  $x=i$  or  $y=j$ , for integers  $i$  and  $j$ .*

We transform  $G$  into  $G'$  by the following way: (1) The vertices of  $G$  are modeled by disks of radius  $\frac{1}{2}$  centered at the location of the vertices in the embedding, (2) the edges of  $G$  are replaced by chain of radius  $\frac{1}{2}$  disks, see figure 4. Let  $c[e_i]$  be the chain of disks in  $G'$  that correspond the edge  $e_i$  in graph  $G$ .

We can observe that any proper 3-coloring of  $c[e_i] \cup \{u, v\}$  assigns  $u$  and  $v$  different colors. On the other hand if we assign different colors to  $u$  and  $v$ , there exists a proper 3-coloring of  $c[e_i] \cup \{u, v\}$ . Therefore the reductions is complete.  $\square$

### 2.4 Maximum Independent Set Problem

An independent set of a graph  $G=(V,E)$  is a vertex subset  $V' \subseteq V$  such that for any pair of vertices in  $V'$ , there is no edge between them in  $E$ . The independence number  $\alpha(G)$  is defined as the size of a maximum

independent set in  $G$ . It can be easily seen that if  $V'$  is a vertex cover of  $G$ , then  $V-V'$  is an independent set. In other words this two problems (minimum vertex cover and maximum independent set) are polynomially equivalent. We will reduce Min Vertex Cover in Planar Graphs problem which is shown to be NP-complete into Vertex Cover Problem in UDG.

**Theorem 2.4.** [Clark et al., 1990] *Minimum cover problem in UDG is NP-hard.*

*Proof.* We will again use the same construction strategy in proof of theorem 2.2. Let  $G$  be a planar graph with maximum degree 3 and having a vertex cover  $S$  with  $|S| \leq k$  and let  $G'$  be a unit disk graph having a vertex cover  $S'$  with  $|S'| \leq k'$ . We construct  $G$  in the plane using lemma 2.3. Vertices are modeled as unit disks and each edge  $\{u, v\}$  by a chain having an even number of  $2k_{uv}$  of disks. We can verify that  $G$  has a vertex cover  $S$  such that  $|S| \leq k$  if and only if  $G'$  has a vertex cover  $S'$  such that  $|S'| \leq k + \sum_{uv \in E(G)} k_{uv}$   $\square$

### 3 Approximation Algorithms

In some situations there is no polynomial algorithms to solve the problem. In this case, we have three options: (a) we can work in a subcase where a polynomial time algorithm exist, (b) we can develop a heuristic method, however there is no guarantee for the optimality or (c) we can find an approximation algorithm. Approximation algorithms cannot guarantee the optimality but at least can give a performance guarantee. We know that the solutions generated from an approximation algorithm lies on an error bound. This is not true for an ordinary heuristic method. Formally, an approximation algorithm for an optimization problem  $\pi$  provides a performance guarantee of  $\rho$  if for every instance  $I$  of  $\pi$ , the solution value returned by the approximation algorithm is within the factor  $\rho$  of the optimal value for  $I$ . In this section, we will state approximations to some graph problems.

#### 3.1 Max Independent Set

We already show that maximum independent set problem in UDG is NP-hard. So, we will suggest the following approximation algorithm, then we will proof its performance. [Fishkin, 2004]

1. Find a vertex whose neighborhood does not contain an IS of size larger than 3.
2. Add such a vertex to an IS.
3. Remove all its neighbors.

We start by finding a vertex whose neighborhood does not contain an independent set of size larger than 3. This vertex always exists because of property 1.5 in UDG. Moreover this vertex can be found in polynomial time  $O(|V|^5)$ . We keep this vertex on a list, then we remove with all its neighborhoods. We know that from property 1.6 that any induced subgraph of a unit disk graph  $G$  is also a unit disk graph. Therefore we can keep doing this iterations until we get an empty graph.

**Lemma 3.1.** *The performance guarantee of this algorithm is 3.*

*Proof.* Let's define  $r$ -neighborhood of a vertex  $v \in IS(G)$  is  $N(v) \cup \{v\}$  where  $N(v)$  is the neighborhood of  $v$ . By construction, every vertex in  $OPT(G)$  is in the  $r$ -neighborhood of at least one vertex in  $IS(G)$ , if not we can add it to  $OPT(G)$  which contradicts with the optimality. Also by construction, the size of a maximum independent set in every  $r$ -neighborhood is 3. Therefore,  $r$ -neighborhood of a vertex  $v$  contains at most 3 vertices from  $OPT(G)$ . We have  $|OPT(G)| \leq 3|IS(G)|$ .  $\square$

#### 3.2 Minimum Vertex Coloring

Minimum vertex coloring is also NP-hard problem in UDG. We will consider first-fit algorithm with a  $y$ -coordinate descending order. We start with the unit disk at the top of the representation and color it with the first available color and consider remaining graph and repeat until we color all the graph. We claim that this algorithm uses at most  $3\omega(G) - 2$  colors for a given UDG with the representation.

1. Consider a disk  $D$  at the time it is assigned its color
2. Let  $d(D)$  be previously colored disks that intersect  $D$  have a  $y$ -coordinate that is not smaller than  $y$ -coordinate of  $D$ .
3. The previously colored neighbors can be covered by at most three cliques.
4. The disks with centers in the same region delimited by an angle of  $\frac{\pi}{3}$  must form a clique (together with  $D$ ).
5.  $d(D)$  is at most  $3(\omega(G) - 1)$
6. The algorithm uses at most  $3\omega(G) - 2$

It is always true that  $\omega(G) \leq \chi(G)$ . Therefore this algorithm has an approximation ratio of 3. Actually, we can generalize this method and find a first-fit algorithm which does not require the representation. On the other hand, approximation ratio for DG is 5 with and without representation.

## 4 Conclusion

In this report, we tried to introduce a new class of graph called disk graphs. We have seen that there are many applications on this domain. Therefore, UDG are one of the well studied graphs in the literature. We started by giving some important properties in UDG. Then we stated complexity results. We remarked that almost all problems are NP-hard, therefore approximation algorithms are suggested in the literature. Some properties of DG(UDG) allow us to derive constant approximation ratios which is not possible for the general graphs.

There are still some open questions in this domain for example the complexity of max clique problem in disk graphs in general. Besides, since all the algorithms presented are simple in a sense, we can try to improve approximation ratios by developing new complex but still polynomial algorithms. As we stated before, additional graph representation information can also be a way to reduce the approximation ratios.

## References

- K. I. Aardal, C. P. M. van Hoesel, A. M. C. A. Koster, C. Mannino, and A. Sassano. Models and solution techniques for the frequency assignment problem. ZIB-report 01-40, Konrad-Zuse-Zentrum für Informationstechnik Berlin, Berlin, Germany, 2001. URL <http://www.zib.de/Publications/abstracts/ZR-01-40/>.
- Heinz Breu and David G. Kirkpatrick. Unit disk graph recognition is np-hard. *Computational Geometry. Theory and Applications*, 9:9-1, 1998.
- Brent N. Clark, Charles J. Colbourn, and David S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86(1-3):165-177, December 1990.
- Aleksei V. Fishkin. *Approximation and Online Algorithms*, chapter Disk Graphs: A Short Survey. Springer Berlin / Heidelberg, 2004.
- W. K. Hale. Frequency assignment: Theory and applications. *Proceedings of the IEEE*, 68:1497-1514, 1980.
- M. V. Marathe, H. Breu, H. B. Hunt Iii, S. S. Ravi, and D. J. Rosenkrantz. Simple heuristics for unit disk graphs. *Networks*, 25:59-68, 1995.
- Leslie G. Valiant. Universality considerations in vlsi circuits. *IEEE Trans. Computers*, 30(2):135-140, 1981.