

Max-Cut and Max-Bisection are NP-hard on unit disk graphs*

Josep Díaz¹ and Marcin Kamiński²

¹Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya
08034 Barcelona
diaz@lsi.upc.edu

² RUTCOR, Rutgers University
640 Bartholomew Road
Piscataway, NJ 08854, USA
mkaminski@rutcor.rutgers.edu

January 21, 2007

Abstract

We prove that the MAX-CUT and MAX-BISECTION problems are NP-hard on unit disk graphs. We also show that λ -precision graphs are planar for $\lambda > 1/\sqrt{2}$ and give a dichotomy theorem for MAX-CUT computational complexity on λ -precision unit disk graphs.

Keywords: computational complexity, Max-Cut, Max-Bisection, NP-hard, unit disk graphs.

1 Introduction

Given a simple graph G , a *cut* in G is a partition of its vertex set into two parts. A cut whose parts have the same cardinality is called a *bisection*. The size of a cut is the number of edges that have their endpoints into two different parts of the cut. A *maximum cut* is a cut with maximum size and the algorithmic MAX-CUT problem consists in finding one given the input graph. Similarly, a *maximum bisection* is a bisection of maximum size and the MAX-BISECTION problem is to find one. Let $mc(G)$ denote the maximum size of a cut in G .

The MAX-CUT problem is known to be NP-hard for general graphs ([11]) and remains NP-hard even when the input graph is restricted to be a split or 3-colorable graph ([3]). As shown in [15], the problem is NP-hard also for the class of graphs with bounded maximum degree Δ , if $\Delta \geq 3$. On the other hand, the MAX-CUT problem can be solved in polynomial time for planar graphs ([8], [13]) or graphs with bounded treewidth ([3]).

The MAX-BISECTION problem is NP-hard for general graphs ([6]). However, contrary to the MAX-CUT problem, MAX-BISECTION remains NP-hard on planar graphs (result of Jerrum presented in [10]).

Unit disk graphs are intersection graphs of unit diameter disks in the plane. Place n disks of diameter one in the plane so that the centers of disks do not coincide. An undirected graph is said to be a *unit disk graph* if there exists a one-to-one correspondence between its vertices and disks

*Research was partially supported by the EC Research Training Network HPRN-CT-2002-00278 (COMBSTRU) and the Spanish CYCIT: TIN2004-07925-C03-01 (GRAMMARS). The first author was also supported by *La distinció* per a la promoció de la recerca de la Generalitat de Catalunya, 2002. The work was done when the second author was visiting Universitat Politècnica de Catalunya.

in such a way that two vertices are adjacent if and only if the corresponding disks intersect. (We assume tangent disks do intersect, however the classes of unit disk graphs with open or closed disks coincide [12].) Each configuration of disks that defines a unit disk graph is called its *intersection model*. This can be easily translated into a *proximity model* which is a collection of distinct points on the plane in one-to-one correspondence with the vertices of the graph in such a way that two vertices are adjacent if and only if two points are at distance at most one. Notice that unit disk graphs are simple and loopless. Often it is convenient to identify points in the proximity model and vertices of the graph and we will do so.

In recent years there has been an increasing interest in the study of unit disk graphs and their randomized version, the random geometric graphs, due to their use as models of wireless communication networks (see for ex. [1]).

Sometimes an additional assumption on the proximity model is imposed and points are required to be at distance greater or equal λ from each other. Graphs which have such a proximity model are called λ -precision unit disk graphs. Notice that the class of λ_1 -precision unit disk graphs is contained in the class of λ_2 -precision unit disk graphs, for every $\lambda_1 \geq \lambda_2$.

In [4], many algorithmic problems – CHROMATIC NUMBER, MAXIMUM INDEPENDENT SET, MINIMUM DOMINATING SET – have been proved to be NP-hard on unit disk graphs. The authors of that paper also provide a polynomial-time algorithm for MAX-CLIQUE. They conclude the paper pointing out that the computational complexity of all considered problems is the same for planar and unit disk graphs.

The computational complexity of the MAX-CUT and MAX-BISECTION problem on unit disk graphs has not been known, even though several authors offered approximation algorithms for MAX-CUT and MAX-BISECTION. In [9], among other approximation results, the authors present a polynomial-time approximation scheme for λ -precision unit disk graphs. A polynomial-time approximation scheme for the MAX-BISECTION problem on unit disk graphs was developed in [10]. It is also worth mentioning that the problem of recognizing unit disk graphs is NP-hard ([2]) so approximation algorithms require providing an intersection model for the input graph.

In this paper, we prove that the MAX-CUT and MAX-BISECTION problems are NP-hard for unit disk graphs, which to the knowledge of the authors were open problems (see [14, 10]). Also, it turns out that MAX-CUT is the first problem known whose computational complexities on planar and unit disk graphs do not coincide. In the last section, we show that the λ -precision unit disk graphs are planar for $\lambda > 1/\sqrt{2}$. An interesting open problem is to investigate the computational complexity of MIN-BISECTION on unit disk graphs ([5]).

2 Mesh drawings

A *drawing* of a graph $G = (V, E)$ is a mapping f which assigns to each vertex of G a distinct point in the plane and to each edge uv a continuous arc between $f(u)$ and $f(v)$, not passing through the image of any other vertex. We also allow interiors of images of two different edges to intersect only at a finite number of points. Each such intersection is called a *crossing point*. A graph which can be drawn in the plane without any crossing points is called a *planar graph*. Below, if it does not lead to misunderstanding, we often do not distinguish between vertex/edge and its image.

The *mesh* $\mathcal{M}$ is the set of points in the plane which have at least one integral coordinate. These points of the mesh that have two integral coordinates are called *mesh crosses*. The distance between two points of $v = (x_1, y_1)$ and $w = (x_2, y_2)$ in $\mathcal{M}$ is the Manhattan distance $d(v, w) = \{|x_1 - x_2| + |y_1 - y_2|\}$.

A *mesh drawing* of a graph is a drawing in which the images of all vertices are mesh crosses and the images of edges belong to the mesh. Notice that in a mesh drawing only two edges can intersect

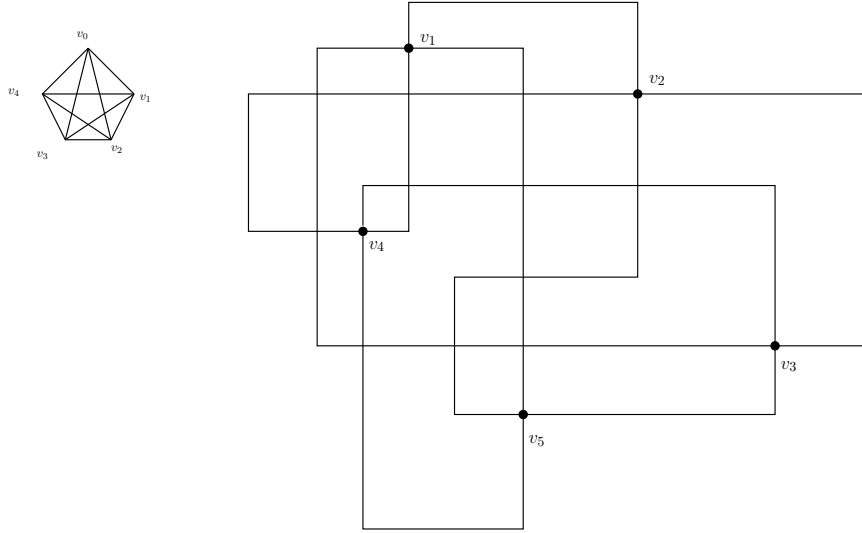


Figure 1: K_5 and its mesh drawing.

at a crossing point and a crossing point always occurs at a mesh cross.

A necessary condition for a graph to have a mesh drawing is to be of maximum degree 4. In fact that condition is also sufficient, as shown by the following argument. Let G be any graph of maximum degree 4. Place all vertices of G at distinct mesh crosses in such a way that the distance between any pair of vertices is at least 5. For a vertex of G whose corresponding mesh cross has coordinates (a, b) we define four “corridors”: $x = a - 1$ or $y = b + 2$; $x = a - 2$ or $y = b + 1$; $x = a + 1$ or $y = b - 2$; $x = a + 2$ or $y = b - 1$.

Then an edge between any pair of vertices in $\mathcal{M}$ can be drawn entirely within the corridors of its endpoints. Hence, we can construct a mesh drawing of G and the construction can be done in polynomial time.

Let us say that a mesh drawing is *standard* if the distance between any two crossing points is at least 10, the distance between any two vertices is at least 10, the distance between any vertex and any crossing point is at least 10, and if interior points of two edges belong to two different parallel vertical or horizontal lines, the distance between these two lines is at least 10.

Given a mesh drawing of a graph, we can create its standard mesh drawing. Suppose that the distance between two vertices v, w is less than 10. Pick a line which separates these two vertices (i.e. at most one of v, w belongs to the line). Without loss of generality, we can assume it is a horizontal line $y = k$ for some $k \in \mathbb{Z}$ and v lies above that line. Now all the points of the drawing whose y -coordinate is at most k will be moved down by some constant c . The constant should be such that after w is moved down by c the distance between v and w is at least 10. All the edges that were broken by that operation should be extended (by adding a vertical segment of length c) in a way that makes a new graph isomorphic to the original one. This way we decreased the number of pairs of vertices that were at distance at most 10 from each other. Similar techniques can be used to satisfy all the conditions of the standard mesh drawing. Moreover, given a graph G , its standard mesh drawing can be found in polynomial time.

Lemma 1. *Every graph of maximum degree 4 has a standard mesh drawing and the drawing can be found in time polynomial in the number of vertices of the graph.*

Before we proceed to the next section, we want to note a useful fact. Given a graph G , let G'

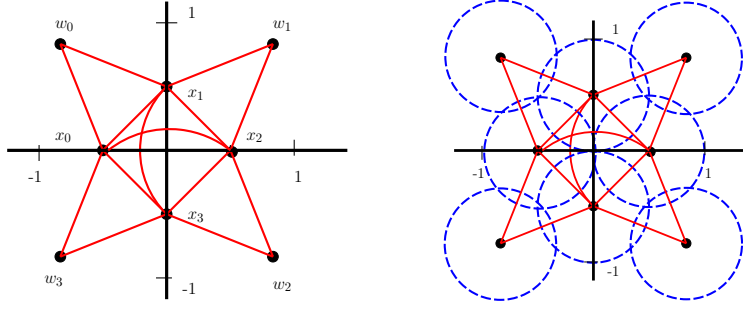


Figure 2: Gadget H and its intersection model.

be the graph obtained from G by subdividing one of the edges of G twice. Then,

$$mc(G') = mc(G) + 2. \quad (1)$$

3 Reduction

In this section we give a polynomial-time reduction from the class of graphs with maximum degree $\Delta \leq 4$ to the class of unit disk graphs. First we define a gadget that will be used in the reduction.

Let H be the graph on vertices: $x_0, x_1, x_2, x_3, w_0, w_1, w_2, w_3$ such that $H[x_0, x_1, x_2, x_3]$ is a K_4 , $H[w_i, x_i, x_{i+1}]$ is a K_3 , for $i = 0, 1, 2, 3$, where $i + 1$ is taken modulo 4, and H has no other edges than those needed to satisfy the two previous conditions.

Lemma 2. H is a $(1/\sqrt{2})$ -precision unit disk graph.

Proof. Let us consider the following proximity model for H . Place vertices x_0, x_1, x_2, x_3 at points $(1/2, 0)$, $(0, 1/2)$, $(-1/2, 0)$ and $(0, -1/2)$, respectively. Vertices w_0, w_1, w_2, w_3 should be put at points $(4/5, 4/5)$, $(-4/5, 4/5)$, $(-4/5, -4/5)$ and $(4/5, -4/5)$, respectively. Now it is easy to verify that this is indeed a proximity model of H and all distances are at least $1/\sqrt{2}$. $\square$

Given a graph G and the gadget H just defined, let $(x_0x_2), (x_1x_3)$ be two edges in G not incident to each other. We define the graph G^* to be the graph obtained from G by adding the new vertices w_0, w_1, w_2, w_3 and edges necessary to make $G^*[x_0, x_1, x_2, x_3, w_0, w_1, w_2, w_3]$ isomorphic to H . We say that H was constructed in G on $(x_0x_2), (x_1x_3)$.

The following lemma is straightforward by considering that the contribution of each triangle w_i, v_i, v_{i+1} to the maximum cut in G^* is exactly 2.

Lemma 3. Let G^* be the graph obtained from G by constructing H on two non-incident edges of G . Then, $mc(G^*) = mc(G) + 8$

Let us state and prove the main theorem.

Theorem 4. MAX-CUT is NP-hard on unit disk graphs.

Proof. To prove that MAX-CUT is NP-hard on unit disk graphs we are going to present a polynomial-time procedure that takes an arbitrary graph G of maximum degree 4 and produces a unit disk graph G' . Moreover, knowing $mc(G')$ we are able compute $mc(G)$ in polynomial time.

STEP 1 Let G be a graph of maximum degree 4. Let us consider its standard mesh drawing. According to Lemma 1, it does exist and can be found in polynomial time.

STEP 2 Subdivide edges of G putting new vertices at mesh crosses that are not crossing points. Now we have two types of vertices – those that were created in this step and those that correspond to original vertices of G .

STEP 3 For each crossing point (x, y) , subdivide only the edge between $(x, y - 1)$ and $(x, y + 1)$ (the vertical one). We remove two vertices placed at $(x - 1, y)$ and $(x + 1, y)$ and consequently all the edges they were incident to. We place new vertices at coordinates $(x - 1.5, y + 0.5)$, $(x - 0.5, y + 0.5)$, $(x + 0.5, y + 0.5)$, $(x + 1.5, y + 0.5)$ and create edges between pairs of vertices: $(x - 2, y)$ and $(x - 1.5, y + 0.5)$, $(x - 1.5, y + 0.5)$ and $(x - 0.5, y + 0.5)$, $(x - 0.5, y + 0.5)$ and $(x + 0.5, y + 0.5)$, $(x + 0.5, y + 0.5)$ and $(x + 1.5, y + 0.5)$, $(x + 1.5, y + 0.5)$ and $(x + 2, y)$. Notice that the drawing is not a mesh drawing anymore.

STEP 4 For each crossing point, construct a copy of graph H on the crossing edges. Place new vertices in the way described in the proof of Lemma 2 and create straight line edges in each copy of H .

STEP 5 If along one of the original edges of G there is an odd number of vertices, we need to subdivide one of the new edges once more. Pick an edge between points (x, y) and $(x + 1, y)$ whose both endpoints and the neighbors of the endpoints are of degree at most 2 and the endpoints belong to the original edge of G . Move vertex at (x, y) to a new position $(x - 1/4, y)$ and vertex at $(x + 1, y)$ to $(x + 5/4, y)$. Create a new vertex at $(x + 1/2, y + 1/2)$ and edges between pairs of vertices: $(x - 1/4, y)$ and $(x + 1/2, y)$, $(x + 5/4, y)$ and $(x + 1/2, y)$.

Let $U(G)$ be the graph whose drawing was constructed above. We will show that the graph is in fact a unit disk graph.

Claim 5. $U(G)$ is a $(1/\sqrt{2})$ -precision unit disk graph.

Proof of Claim. To prove that $U(G)$ is a unit disk graph we will show that placing vertices in the plane at the same coordinates as in the construction above gives a proximity model of $U(G)$.

Notice that after Step 2 all adjacent vertices, except the endpoints of crossing edges, are at distance exactly 1. Once crossing edges have been replaced by a construction described in Step 3, all the neighbors of a vertex are within distance 1 from it.

Notice that after constructing a copy of H at a crossing point (Step 4), “ w ” vertices are farther than a unit distance from all the vertices that they are not adjacent to. Also, the new edges created between endpoints belonging to two crossing edges are now connected. This together with Lemma 2 yields that the graph obtained after Step 4 is a unit disk graph.

Observe that if we start with a standard drawing, an edge whose both endpoints and the neighbors of the endpoints are of degree at most 2 can always be found. It is easy to see, that the construction described in Step 5 leaves a unit disk graph. $\square$

Let k be the number of crossing points in the standard mesh drawing of G from Step 1, and let t be the total number of subdivisions of all the original edges of G created at the end of executing Step 5. Notice that t must be even.

Claim 6. $mc(U(G)) = mc(G) + 8k + t$

Proof of Claim. A copy of H is constructed on each pair of crossing edges and each copy of H increases the value of the maximum cut by 8 (Lemma 3). Also, each double subdivision of an edge increases the value of maximum cut by 2 (Equation 1). Hence, $mc(U(G)) = mc(G) + 8k + t$. $\square$

We have shown that any graph G of maximum degree 4 can be transformed in a unit disk graph G' , and this reduction can be implemented in polynomial time. If there exist a polynomial-time algorithm solving MAX-CUT on unit disk graphs, then knowing the construction and $mc(G')$, the value of maximum cut of G' can be also computed in polynomial time. But as mentioned in section 1, MAX-CUT is NP-hard on graphs with maximum degree Δ , if $\Delta \geq 3$, and therefore it is also NP-hard on unit disk graphs. $\square$

Taking two disjoint copies of a unit disk graph G creates a unit disk graph whose maximum bisection is twice the value of maximum cut of G . The following fact is therefore a simple corollary of Theorem 4.

Corollary 7. *MAX-BISECTION is NP-hard on unit disk graphs.*

4 Precision and planarity

In this section we study the relation between λ -precision and planarity.

Theorem 8. *A λ -precision unit disk graph is planar, for every $\lambda > 1/\sqrt{2}$.*

Proof. Let G be a λ -precision unit disk graph, for some $\lambda > 1/\sqrt{2}$. Consider a drawing of G defined by its proximity model: put vertices in the plane at the same positions as in the proximity model and connect two of them by a straight line segment if and only if the distance between them is at most 1.

Let us consider an edge e of G of length x . The set of points at distance more than $1/\sqrt{2}$ from both endpoints of e consists of two disjoint regions; one on each side of the straight line containing e . It is easy to verify that the distance between these two regions is strictly greater than $2\sqrt{1/2 - (x/2)^2}$ and therefore always strictly greater than 1. Hence, there is no edge crossing e and the planarity of G follows. $\square$

Finally, we can state a dichotomy result on MAX-CUT for λ -precision unit disk graphs which is a consequence of Theorem 4 and Theorem 8.

Theorem 9. *The MAX-CUT problem is NP-hard in the class of λ -precision unit disk graphs if $\lambda \leq 1/\sqrt{2}$ and can be solved in polynomial time if $\lambda > 1/\sqrt{2}$*

Proof. Notice $U(G)$ constructed in the proof of Theorem 4 is a $(1/\sqrt{2})$ -precision unit disk. Hence, MAX-CUT is NP-hard in the class of λ -precision unit disk graphs for $\lambda \leq 1/\sqrt{2}$.

Since λ -precision unit disk graphs are planar for $\lambda > 1/\sqrt{2}$ and, as we mentioned, MAX-CUT can be solved in polynomial time on planar graphs, then the MAX-CUT problem can be solved on these graphs in polynomial time. $\square$

References

- [1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38:393–422, 2002.
- [2] H. Breu, and D.G. Kirkpatrick. Unit disk recognition is NP-hard. *Computational Geometry*, 9:3–24, 1998.
- [3] H. L. Bodlaender, and K. Jansen. On the complexity of the maximum cut problem. *Nordic Journal of Computing*, 7(1):14–31, 2000.

- [4] N.B. Clark, C.J. Colbourn, and D.S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86:165–177, 1990.
- [5] J. Díaz, M. Penrose, J. Petit, and M. Serna. Approximating layout problems on random geometric graphs. *Journal of Algorithms*, 39:78–116, 2001.
- [6] M. Garey and D. Johnson. *Computers and Intractability*. Freeman. N.Y., 1979.
- [7] M.C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs*. Academic Press, N.Y., 1980.
- [8] F. Hadlock. Finding a maximum cut of a planar graph in polynomial time. *SIAM J. Comput.* 4(3):221–225, 1975.
- [9] H.B. Hunt, M.V. Marathe, V. Radhakrishnan, S. Ravi, D. Rosenkratz, and R.E. Stearns. NC-approximation schemes for NP- and PSPACE-Hard problems for geometric graphs. *J. Algorithms*, 26:238–274, 1998.
- [10] K. Jansen, M. Karpinski, A. Lingas, and E. Seidel. Polynomial-time approximation schemes for Max-bisection on planar and geometric graphs. *SIAM J. Computing*, 35(1):110–119, 2005.
- [11] R.M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*. Plenum Press, 85–104, 1972.
- [12] E. J. van Leeuwen, J. van Leeuwen. On the representation of disk graphs *Technical Report UU-CS-2006-037*. Utrecht University, 2006.
- [13] G.I. Orlova and Y.G. Dorfman. Finding the maximal cut in a graph. *Engineering Cybernetics*, 10:502–506, 1972
- [14] J. Spinrad. *Efficient graphs representations*. Fields Institute Monographs 19. AMS, Providence, 2003.
- [15] M. Yannakakis. Node- and edge-deletion NP-complete problems. In *IEEE Symposium on Theory of Computing (STOC)*, 253–264, 1978.