Coloring of Double Disk Graphs

HONGWEI DU¹, XIAOHUA JIA², DEYING LI² and WEILI WU³
¹Department of Computer Science, Huazhong Normal University, Wuhan, China;
²Department of Computer Science, City University of Hong Kong, Kowloon Tong, Hong Kong,
(E-mail: {jia,dyli}@cs.cityu.edu.hk);
³Department of Computer Science, University of Texas at Dallas, Richardson, TX 75083, USA,
(E-mail: weiliwu@utdallas.edu).
(Received 8 May 2003; accepted 15 May 2003)

Abstract. We study the problem of minimizing the number of colors for vertex-coloring of double
disk graphs and in this note, show a polynomial-time 31-approximation for the problem, which
improves an existing result.

1. Introduction
Consider \( n \) points \( v_1, \ldots, v_n \) on the Euclidean plane, each with two disks centered
at the point. Let \( r_i \) and \( R_i \) with \( r_i \leq R_i \) be radiuses of the two disks at point \( v_i \).
Define a graph \( G \) with vertex set \( \{v_1, \ldots, v_n\} \) and edge \( (v_i, v_j) \) exists if and only
if \( d(v_i, v_j) \leq \max(R_i + r_j, r_i + R_j) \), that is, either outer disk of \( v_i \) intersects
with inner disk of \( v_j \) or inner disk of \( v_i \) intersects with outer disk of \( v_j \). Such
a graph \( G \) is called a double disk graph. The double disk graph has application
in wireless communication [3]. Motivated from frequency assignment problem in
wireless communication, Malesinska et al. [3] studied the problem of minimizing
the number of colors for vertex-coloring of double disk graphs. The problem is
NP-hard since the problem on unit disk graphs is NP-hard [3] and the unit disk
graph is a special case of the double disk graph with \( R_i = r_i = 1 \). They showed a
polynomial-time 33-approximation for the problem. In this short note, we improve
the result by reducing the performance ratio from 33 to 31.

2. Main Results
Consider the following greedy algorithm for vertex-coloring of double disk graph
\( G \):

Step 1. Put all vertices in a list as follows: At each iteration, choose a vertex
with lowest degree and put it at the head of the list; then delete it from the graph.
That is, all vertices of \( G \) are put in a list \( v_1, \ldots, v_n \) such that for every \( i, 1 \leq i \leq n, \)
\( v_i \) has the least degree in subgraph induced by \( \{v_1, \ldots, v_i\} \).
Step 2. Color all vertices as follows: At each iteration, color the head in the list with a smallest color (note: each color is represented by an integer) not appearing in its neighbors. Then delete it from the list. Therefore, vertex \( v_i \) is in color not bigger than one plus the degree of \( v_i \) in subgraph induced by \( \{v_1, ..., v_j\} \).

Malesinska et al. [3] showed that this greedy algorithm has performance ratio 33. We improve it to 31 as follows.

THEOREM 1. The above greedy algorithm is a polynomial-time 31-approximation for vertex-coloring of double disk graphs.

To show this result, we first prove a lemma.

LEMMA 2. For any double disk graph \( G \), there exists a vertex with degree \( \leq 31\omega(G) - 1 \) where \( \omega(G) \) is the size of maximum clique in \( G \).

Proof. Choose a vertex \( v_i \) with smallest \( R_i \). Without loss of generality, we may assume that \( R_i = 2 \). Taking \( v_i \) as the center, draw 19 regular hexagons with edge length one as shown in Figure 1. For every two vertices \( v_j \) and \( v_k \) lying in the same hexagon, edge \( (v_j, v_k) \) must exist since \( d(v_j, v_k) \leq 2 \leq R_j \). This means that all vertices lying in the same hexagon form a clique. To make the clique possibly larger for each outer hexagon, we use arc of outscribing circle to replace some edges. Meanwhile, we divide remaining area into 12 equal areas (see Figure. 1). Next, we show that for each of those 12 areas, all vertices adjacent to \( v_i \), lying into the area, also form a clique.

To do so, we consider two points \( v_j \) and \( v_k \) lying in the same one of the 12 areas and assume \( d(v_j, v_k) \leq d(v_k, v_i) \). We claim that \( d(v_j, v_k) \leq \max(2, d(v_k, v_i) - 2) \).

To prove our claim, we first show three facts as follows:

FACT 1. Suppose \( v_h \) is on the extension of segment \( (v_j, v_k) \). If the claim is true for \( v_j \) and \( v_h \), then it is true for \( v_j \) and \( v_k \).

In fact, \( d(v_h, v_k) \geq d(v_h, v_j) - d(v_j, v_k) \). Hence,

\[
    d(v_j, v_k) = d(v_j, v_h) - d(v_k, v_h)
    \leq \max(2, d(v_j, v_h) - 2) - d(v_j, v_k) + d(v_j, v_k)
    \leq \max(2, d(v_k, v_i) - 2)
\]

FACT 2. Suppose \( d(v_j', v_k) \geq d(v_j, v_k) \) and the claim is true for \( v_j' \) and \( v_k \). Then it is true for \( v_j \) and \( v_k \).

This fact is trivial. However, it has two important special cases:

(a) \( v_j' \) is on the extension of segment \( (v_k, v_j) \).
(b) \( v_j \) is on the segment \( (v_j', v_k) \) and the claim is true for \( v_j' \) and \( v_k \), and also true for \( v_j' \) and \( v_k \). In this case, we have either \( d(v_j, v_k) \leq d(v_j', v_k) \) or \( d(v_j, v_k) \leq d(v_j', v_k) \).