



ELSEVIER

Contents lists available at ScienceDirect

## Information Processing Letters

www.elsevier.com/locate/ipl



# Approximation algorithms for maximum independent set of a unit disk graph



Gautam K. Das<sup>a</sup>, Minati De<sup>b</sup>, Sudeshna Kolay<sup>c</sup>, Subhas C. Nandy<sup>d,\*</sup>,  
Susmita Sur-Kolay<sup>d</sup>

<sup>a</sup> Dept. of Mathematics, Indian Institute of Technology Guwahati, Guwahati, India

<sup>b</sup> Dept. of Computer Science, The Technion—Israel Institute of Technology, Israel

<sup>c</sup> The Institute of Mathematical Sciences, Chennai, India

<sup>d</sup> Indian Statistical Institute, Kolkata, India

## ARTICLE INFO

### Article history:

Received 21 November 2013

Received in revised form 5 November 2014

Accepted 6 November 2014

Available online 15 November 2014

Communicated by R. Uehara

### Keywords:

Maximum independent set

Unit disk graph

Approximation algorithms

PTAS

Computational geometry

## ABSTRACT

We propose a 2-approximation algorithm for the maximum independent set problem for a unit disk graph. The time and space complexities are  $O(n^3)$  and  $O(n^2)$ , respectively. For a penny graph, our proposed 2-approximation algorithm works in  $O(n \log n)$  time using  $O(n)$  space. We also propose a polynomial-time approximation scheme (PTAS) for the maximum independent set problem for a unit disk graph. Given an integer  $k > 1$ , it produces a solution of size  $\frac{1}{(1+\frac{1}{k})^2}|OPT|$  in  $O(k^4 n^{\sigma_k} \log k + n \log n)$  time and  $O(n + k \log k)$  space, where  $OPT$  is the subset of disks in an optimal solution and  $\sigma_k \leq \frac{7k}{3} + 2$ . For a penny graph, the proposed PTAS produces a solution of size  $\frac{1}{(1+\frac{1}{k})}|OPT|$  in  $O(2^{2\sigma_k} nk + n \log n)$  time using  $O(2^{\sigma_k} + n)$  space.

© 2014 Elsevier B.V. All rights reserved.

## 1. Introduction

Unit disk graphs play an important role in formulating several problems in mobile ad hoc networks. A *unit disk graph*  $G = (V, E)$  is the intersection graph of a set of circular disks  $\mathcal{C} = \{C_1, C_2, \dots, C_n\}$ , placed in  $\mathbb{R}^2$ , each having diameter 1. The center of disk  $C_i$  is denoted by  $c_i$ . Each vertex  $v_i \in V$  corresponds to a disk  $C_i$ , and an edge  $(v_i, v_j) \in E$  indicates that the corresponding pair of unit disks  $C_i$  and  $C_j$  intersect, i.e.,  $\delta(c_i, c_j) \leq 1$ , where  $\delta(a, b)$  is the Euclidean distance between a pair of points  $a, b \in \mathbb{R}^2$ . In a mobile network, if all its base stations have the same range of transmission, then these can be viewed as the vertices of a unit disk graph. Various practical problems on this network can be formulated in terms of a unit disk

graph. In this paper, we consider the problem of finding a maximum independent set (MIS) in a given unit disk graph, where the co-ordinates  $(x_i, y_i)$  of the center  $c_i$  of each disk  $C_i$  are given.

The MIS problem for unit disk graph is known to be NP-complete [2]. Thus, research on this topic is concentrated on designing efficient approximation algorithms. Most of the related works assume that the geometric representation, i.e., the layout of the set of unit disks is given. In such an environment, the MIS for unit disk graph is defined as follows:

*Given a set  $\mathcal{C}$  of  $n$  circular disks, each of diameter 1, placed arbitrarily in  $\mathbb{R}^2$ , find a subset  $OPT$  of non-intersecting disks in  $\mathcal{C}$  such that  $OPT$  has the maximum cardinality among all possible subsets of non-intersecting disks in  $\mathcal{C}$ .*

**Previous works:** A dynamic programming based shift-invariant strategy was used by Erlebach et al. [7] to design a

\* Corresponding author.

E-mail address: [nandysc@isical.ac.in](mailto:nandysc@isical.ac.in) (S.C. Nandy).

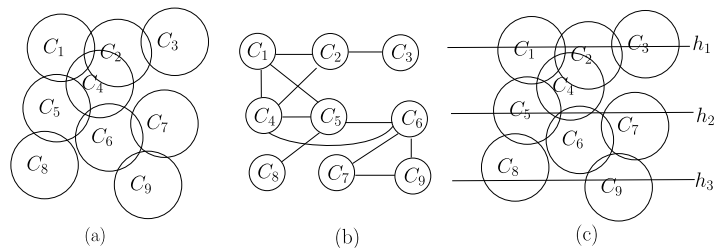


Fig. 1. (a) A layout of unit disks, (b) corresponding unit disk graph, and (c) horizontal lines at distance 1 dividing the region into strips.

polynomial-time approximation scheme (PTAS) for finding a maximum weighted independent set (disjoint disks of maximum total weight) in an intersection graph of  $n$  disks of arbitrary radii. They proposed a  $(1 + \frac{1}{k-1})^2$ -approximation algorithm<sup>1</sup> in  $n^{O(k^4)}$  time, where  $k$  is an integer greater than 1. Matsui [12] showed that if the centers of a set of  $n$  unit disks lie inside a region bounded by a pair of parallel lines at a distance at most  $k$  apart, then an optimal MIS can be obtained in  $O(n^{\lceil \frac{2k}{\sqrt{3}} \rceil})$  time. He also gave an approximation algorithm for the MIS problem for unit disk graphs, that produces a solution of size  $(1 - \frac{1}{r})|OPT|$ , and runs in  $O(rn^{\lceil \frac{2(r-1)}{\sqrt{3}} \rceil})$  time using  $O(n^{2r})$  space, for any positive integer  $r \geq 2$ . Here  $OPT$  is the subset of disks in an optimum solution.

The concept of *thickness* was introduced by van Leeuwen [11] to propose a fixed parameter tractable algorithm for the MIS problem for a unit disk graph. An instance of a unit disk layout is said to have thickness  $\tau$  if the region containing the disks can be split into a set of strips of width 1 such that each strip contains at most  $\tau$  disk centers. Further, he showed that an instance of the MIS problem with thickness  $\tau$  can be solved in  $O(\tau^2 2^{2\tau} n)$  time.

Agarwal et al. [1] proposed a 2-approximation algorithm for the MIS problem for the rectangle intersection graph corresponding to a given set of rectangles of fixed size. They also proposed a PTAS for this MIS problem which produces a  $(1 + \frac{1}{k})$ -approximation result in  $O(n \log n + n^{2k-1})$  time for any positive integer  $k$ . For a set of arbitrary squares or rectangles of bounded aspect ratio in  $\mathbb{R}^d$ , Chan [4] proposed a PTAS that runs in  $O(n^{1/\epsilon^{d-1}})$  time and space with  $0 < \epsilon \ll 1$ . They improved the space complexity to  $O(n)$  by sacrificing the running time to  $O(n^{1/\epsilon^d})$ . Recently, Chan and Har-Peled [5] addressed the MIS problem for pseudo-disks in the plane. In the unweighted case, for a set of  $n$  pseudo-disks, their algorithm produces a solution of size  $(1 - \epsilon)|OPT|$  in  $O(n^{O(1/\epsilon^2)})$  time; this result is similar to Erlebach's results [7] for  $n$  disks of arbitrary radii. For the weighted case, they proposed an  $O(n^3)$  time algorithm to produce an independent set of total weight  $\Omega(|OPT|)$ , where  $OPT$  is a subset of independent disks that produces maximum weight over all possible independent sets, provided the set of pseudo-disks has linear union complexity.

<sup>1</sup> An approximation algorithm for the MIS problem is said to be an  $\alpha$ -approximation algorithm if it produces a solution of size  $\frac{1}{\alpha}|OPT|$ , where  $OPT$  is the subset of disks in an optimal solution of the said problem.

**Our main results:** First, we propose a 2-approximation algorithm for the MIS problem for a given unit disk graph. The time and space complexities of our proposed algorithm are  $O(n^3)$  and  $O(n^2)$ , respectively. The perspective of this algorithm is that (i) *the best known constant factor approximation algorithm* available for this problem achieves an approximation factor 3 with time complexity  $O(n^2)$  [13], and (ii) by plugging  $r = 2$  in Matsui's PTAS result for the MIS problem of unit disk graph [12], we get a 2-approximation algorithm that runs in  $O(n^8)$  time using  $O(n^4)$  space. Hence, our 2-approximation algorithm is much faster than the best known 2-approximation algorithm for the MIS problem for unit disk graphs.

Second, we consider the MIS problem for a *penny graph*, which is a special case of unit disk graph where the corresponding unit disks do not overlap, i.e., the distance between any two disk centers is at least 1. Here an intersection between a pair of disks implies that the distance between corresponding centers is exactly 1. The MIS problem for the penny graph is also NP-complete [3]. Our algorithm produces a 2-approximation result for a penny graph in  $O(n \log n)$  time.

Next, we propose a PTAS for the MIS problem for the unit disk graph. It needs to optimally solve a subproblem of the MIS problem where the centers of a set of unit disks lie inside a square region of size  $k \times k$ . We show that this problem can be solved in  $O(k^3 m^{\sigma_k \log k})$  time using  $O(k \log k + \min(\varphi_k, m))$  space, where  $m$  is the input size of the subproblem,  $\sigma_k \leq \frac{7k}{3} + 2$  and  $\varphi_k \leq 9(\lceil \frac{k}{2} \rceil)^2$ . We then apply the two level shifting strategy of Hochbaum and Maass [10] to get a solution of the original MIS problem, that is of size at least  $\frac{1}{(1 + \frac{1}{k})^2} |OPT|$ , in  $O(k^4 n^{\sigma_k \log k})$  time using  $O(n + k \log k)$  space, where  $n$  is the input size of the original MIS problem. The time complexity of our algorithm is comparable with that of [12]; but its space complexity is  $O(n + k \log k)$  compared to  $O(n^{2k})$  of [12]. For penny graphs, the time and space complexities of our PTAS are  $O(2^{2\sigma_k} nk + n \log n)$  and  $O(2^{\sigma_k} + n)$ , respectively.

## 2. MIS problem for unit disk graphs

A layout of a set  $C$  of unit disks and the corresponding unit disk graph are shown in Figs. 1(a) and 1(b), respectively. An independent set in this unit disk graph consists of a set of disks  $I \subseteq C$  which are mutually non-intersecting. The objective of the MIS problem is to find the largest subset of disks in  $C$  which are mutually non-intersecting.

As in the 2-approximation algorithm for the MIS problem of fixed width rectangle intersection graph [1], we

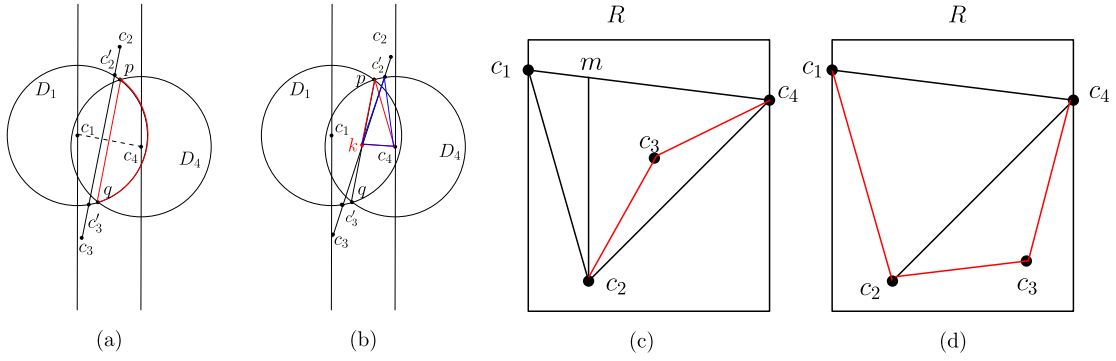


Fig. 2. Proof of Lemma 1.

split the region containing the members in  $\mathcal{C}$  into a set of  $s$  ( $\leq n$ ) disjoint strips  $\{H_1, H_2, \dots, H_s\}$  of unit height, separated by horizontal lines at  $y$ -coordinates  $\{h_1, h_2, \dots, h_{s+1}\}$ . The  $j$ -th strip  $H_j$  contains the set of disks  $\mathcal{Q}_j = \{C_t | C_t \in \mathcal{C} \text{ and } h_j < y_t \leq h_{j+1}\}$  (see Fig. 1(c)). Finally, we consider only those strips which contain at least one disk center.

Matsui [12] showed that if the width of a strip is  $\frac{\sqrt{3}}{2}$ , then the intersection graph of unit disks centered inside a strip is a co-comparability graph [8]. Thus, the optimum solution of the MIS problem for such a graph can be computed in  $O(m^2)$  time [12], where  $m$  is the number of disk-centers inside that strip. We show that for a strip of width 1, the optimum solution of the MIS problem inside a strip can be obtained in  $O(m^3)$  time and  $O(m^2)$  space using dynamic programming. Finally, we use this algorithm for designing our 2-approximation algorithm for the MIS problem for the given unit disk graph.

We compute the MIS of disks in each strip separately. Let  $I_j$  denote a maximum independent set of disks whose centers are in the strip  $H_j$ . Now observe the following.

**Observation 1.** If  $|j - k| > 1$ , then the disks in  $I_j$  and  $I_k$  are non-intersecting.

However, disks in  $I_j$  and  $I_{j+1}$  may intersect.

So, both  $IS_{\text{odd}} = \{I_1 \cup I_3 \cup \dots\}$  and  $IS_{\text{even}} = \{I_2 \cup I_4 \cup \dots\}$  are independent sets. We report the solution  $IS$  for the MIS problem as the one among  $IS_{\text{odd}}$  and  $IS_{\text{even}}$  having the larger cardinality. In Theorem 1, we show that  $|IS| \geq \frac{1}{2}|OPT|$ , where  $OPT$  is the optimum solution of the MIS problem for the set of disks  $\mathcal{C}$ .

Now, we describe the method of computing the MIS of unit disks whose centers lie in a strip  $H$  of width 1. The following result is pertinent to our method.

**Lemma 1.** Let  $C_1, C_2, C_3$  and  $C_4$  be four disks of unit diameter with centers  $c_1 = (x_1, y_1), c_2 = (x_2, y_2), c_3 = (x_3, y_3)$  and  $c_4 = (x_4, y_4)$ , respectively, lying inside a horizontal strip  $H$  of width 1, and  $x_1 < x_2 < x_3 < x_4$ . If  $C_1, C_2, C_3$  are pairwise non-intersecting and  $C_2, C_3, C_4$  are also pairwise non-intersecting, then  $C_1$  and  $C_4$  are also non-intersecting.

**Proof.** Suppose that (i)  $C_1, C_2, C_3$  are mutually non-intersecting and (ii)  $C_2, C_3, C_4$  are also mutually non-inter-

secting. As a contradiction, let us assume that  $C_1$  and  $C_4$  are intersecting, i.e.,  $\delta(c_1, c_4) \leq 1$ .

Consider the vertical strip  $V$  whose left and right boundaries are at  $c_1$  and  $c_4$ , respectively (see Fig. 2(a)). As  $x_1 < x_2 < x_3 < x_4$ , both  $c_2$  and  $c_3$  are inside the vertical strip  $V$ . Next, consider two circles  $D_1$  and  $D_4$  of unit radius centered at  $c_1$  and  $c_4$ , respectively. As the unit disks  $C_2$  and  $C_3$  are mutually non-intersecting to both  $C_1$  and  $C_4$ , their centers  $c_2$  and  $c_3$  must be outside the region  $D_1 \cup D_4$ . So,  $c_2$  and  $c_3$  are inside the vertical strip  $V$  but outside the region  $D_1 \cup D_4$ . Now, there are the following two possibilities for the positions of  $c_2$  and  $c_3$ :

**Case 1:**  $c_2$  and  $c_3$  are in different sides of the line segment  $\overline{c_1c_4}$ ,

**Case 2:** both  $c_2$  and  $c_3$  are in the same side of the line segment  $\overline{c_1c_4}$ .

In Case 1, let  $c'_2$  and  $c'_3$  be the extreme two intersection points of the line segment  $\overline{c_2c_3}$  with the region  $D_1 \cup D_4$  (see Fig. 2(a)). So,  $\delta(c_2, c_3) \geq \delta(c'_2, c'_3)$ . Here, we can have the following two sub-cases.

**Case 1.1:**  $c'_2$  and  $c'_3$  are on the same circle (see Fig. 2(a)),

**Case 1.2:**  $c'_2$  and  $c'_3$  are on the different circles (see Fig. 2(b)).

In Case 1.1, without loss of generality, assume that  $c'_2$  and  $c'_3$  are on the circle  $D_1$ . Here, the line segments  $\overline{p\overline{q}}$  and  $\overline{c'_2c'_3}$  do not intersect with each other but both of them intersect with the line segment  $\overline{c_1c_4}$ . Thus the minor arc  $\widehat{c'_2c'_3}$  of  $D_1$  always contains the minor arc  $\widehat{pq}$  of  $D_1$ . Thus  $\delta(c_2, c_3) \geq \delta(c'_2, c'_3) > \delta(p, q) = 2 \times \sqrt{1 - (\delta(c_1, c_4)/2)^2} \geq \sqrt{3}$  as  $\delta(c_1, c_4) \leq 1$ .

In Case 1.2, the line segments  $\overline{c'_2c'_3}$  and  $\overline{p\overline{q}}$  must be intersecting. Let  $k$  be this intersection point (see Fig. 2(b)). Consider the triangles  $\Delta(c'_2c_4k)$  and  $\Delta(pc_4k)$  having common side  $\overline{c_4k}$ ,  $\delta(c_4, c'_2) = \delta(c_4, p)$  and  $\angle c'_2c_4k \geq \angle pc_4k$ . So,  $\delta(c'_2, k) \geq \delta(p, k)$ . Similarly, from the two triangles  $\Delta(kc_1c'_3)$  and  $\Delta(kc_1q)$ , we can show that  $\delta(k, c'_3) \geq \delta(k, q)$ . Thus, here also  $\delta(c_2, c_3) \geq \delta(c'_2, c'_3) = \delta(c'_2, k) + \delta(k, c'_3) \geq \delta(p, k) + \delta(k, q) = \delta(p, q) \geq \sqrt{3}$ .

Now, consider the rectangle  $R = H \cap V$  which contains  $c_1, c_2, c_3$  and  $c_4$ . As the longest line segment that can be accommodated inside the rectangle  $R$  is its diagonal, which is of length at most  $\sqrt{2}$ , so  $\delta(c_2, c_3) \leq \sqrt{2}$  which contradicts both Case 1.1 and Case 1.2. Thus, Case 1 is not possible.

In Case 2, we have the following two sub-cases:

**Case 2.1:**  $c_1$  and  $c_3$  are in the same side of the line segment  $\overline{c_2c_4}$  (see Fig. 2(c)),

**Case 2.2:**  $c_1$  and  $c_3$  are in different sides of the line segment  $\overline{c_2c_4}$  (see Fig. 2(d)).

In Case 2.1, since  $x_2 < x_3$ ,  $c_3$  must be in the triangle  $\Delta(c_2, c_4, m)$ , where  $\overline{c_2m}$  is a vertical line (see Fig. 2(c)). Here,  $\delta(c_2, c_3) + \delta(c_3, c_4) \leq \delta(c_2, m) + \delta(m, c_4) \leq 2$ . So, the unit disk  $C_3$  intersects either the unit disk  $C_2$  or  $C_4$ , which is a contradiction. So, the possibility of case 2.1 is ruled out.

In Case 2.2, as the maximum distance between  $c_1$  and  $c_4$  along the boundary of  $R = H \cap V$  is 3, we have  $\delta(c_1, c_2) + \delta(c_2, c_3) + \delta(c_3, c_4) \leq 3$ . This implies that  $\min\{\delta(c_1, c_2), \delta(c_2, c_3), \delta(c_3, c_4)\} \leq 1$  (see Fig. 2(d)). Thus, at least one of the corresponding pair of disks are intersecting, which contradicts one of the two premises (i) or (ii) above for the lemma. Thus, our initial assumption that  $C_1$  and  $C_4$  are intersecting is wrong. This proves that the statement of the lemma is correct.  $\square$

Let  $Q = \{C_1, C_2, \dots, C_m\}$  be a set of  $m$  unit disks whose centers lie in the strip  $H$ . We assume that the centers of no two disks have the same  $x$ -coordinate. In our algorithm, we consider the disks in  $Q$  are in increasing order of the  $x$ -coordinates of their centers.

We compute the MIS for  $Q$  by maintaining an upper triangular matrix  $\mu$  for the dynamic programming. Its  $(j, k)$ -th element corresponds to two non-intersecting disks  $C_j, C_k$ , where  $j < k$ . The indices of the matrix  $\mu$  are  $[0, \dots, m-1] \times [1, \dots, m]$ , and each element  $\mu[j, k]$  contains the size of the largest independent set containing  $C_j$  and  $C_k$  as the rightmost two elements.

Initially, all the matrix elements contain 0. The disks are processed in increasing order of the  $x$ -coordinates of their centers. When a disk  $C_k$  is being processed, we set  $\mu[0, k] = 1$  indicating that a singleton element  $C_k$  forms an independent set. In other words, we assume a virtual left most unit disk  $C_0$  such that  $C_0 \cap C_k = \emptyset$  where  $k \in \{1, 2, \dots, m\}$ . Lemma 1 suggests to compute the  $j$ -th row of the matrix  $\mu$  as follows:

$$\mu[j, k] = \begin{cases} \max_{i=0}^{j-1} \{\mu[i, j] + 1 \mid C_i \cap C_k = \emptyset\} & \text{if } C_j \cap C_k = \emptyset, \\ 0 & \text{otherwise.} \end{cases}$$

The matrix element  $\mu[j^*, k^*]$  having the maximum value is the size of the maximum independent set of  $Q$ . The corresponding independent set is reported by executing a backward pass starting from  $\mu[j^*, k^*]$ . In each step, from  $\mu[j, k]$  it moves to  $\mu[i, j]$  (to report  $C_i$ ) if  $\mu[i, j] = \max_{\alpha=0}^{j-1} \mu[\alpha, j]$ . If there are multiple choices, any one can

be chosen for reporting, and the search continues from there. The process continues until  $i = 0$  is reached.

**Lemma 2.** *If the centers of a set of  $m$  unit disks lie inside a strip of width 1, then the dynamic programming procedure, presented above, correctly computes the optimum solution of the MIS problem for the corresponding unit disk graph in  $O(m^3)$  time using  $O(m^2)$  space.*

**Proof.** The correctness of our dynamic programming algorithm follows from Lemma 1.

In order to compute each element of the matrix, one needs to consider at most  $m$  entries. Since the number of elements in the matrix  $\mu$  is  $O(m^2)$ , the time and space complexity results follow. The backward pass for reporting the independent set needs  $O(m^2)$  time, since reporting of each entry needs scanning a row of the matrix  $\mu$ .  $\square$

**Theorem 1.** *Given a set  $\mathcal{C}$  of  $n$  unit disks in  $\mathbb{R}^2$ , a subset of at least  $\frac{1}{2}|OPT|$  non-intersecting disks can be obtained in  $O(n^3)$  time using  $O(n^2)$  space, where  $|OPT|$  is a largest subset of mutually non-intersecting disks in the set.*

**Proof.** Let  $OPT_{odd}$  (resp.  $OPT_{even}$ ) be the set of centers of the unit disks in an optimum solution  $OPT$  that lie in odd (resp. even) numbered strips. Therefore,  $|IS_{odd}| \geq |OPT_{odd}|$ , and  $|IS_{even}| \geq |OPT_{even}|$ . Thus,  $|IS_{odd}| + |IS_{even}| \geq |OPT|$ , and hence we have  $2 \max\{|IS_{odd}|, |IS_{even}|\} \geq |OPT|$ . This implies that the size of the reported answer  $|IS| = \max\{|IS_{odd}|, |IS_{even}|\} \geq \frac{1}{2}|OPT|$ .

Since the two sets of unit disks in every pair of odd numbered strips are disjoint, the total time required for computing  $IS_{odd}$  (MIS for all the odd numbered strips) is  $O(n^3)$ . The same result holds for computing  $IS_{even}$  (MIS for all the even numbered strips). Thus the time complexity result follows.

The space complexity follows from the size of the matrix  $\mu$  to be maintained for solving the subproblem corresponding to each strip.  $\square$

### 3. MIS problem for penny graphs

If unit disks do not overlap, then the corresponding unit disk graph is referred as penny graph. In other words, a unit disk graph is a penny graph if it has a unit disk representation in which the distance of each pair of two centers is at least 1. We refer the unit disks corresponding to the penny graph as pennies. Here an edge in the graph indicates that the corresponding two pennies touch each other.

The MIS problem for the penny graph is NP-hard [3]. We show that a 2-approximation result of the MIS problem for such a graph can be computed in  $O(n \log n)$  time.

As in Section 2, we split the plane into strips of width 1. Considering each strip separately, we compute the maximum independent set among the pennies whose centers lie in that strip. We report the union of the solutions of either the even numbered strips or the odd numbered strips depending on whose size is maximum. Following the same argument as in Theorem 1, it can be shown that the size

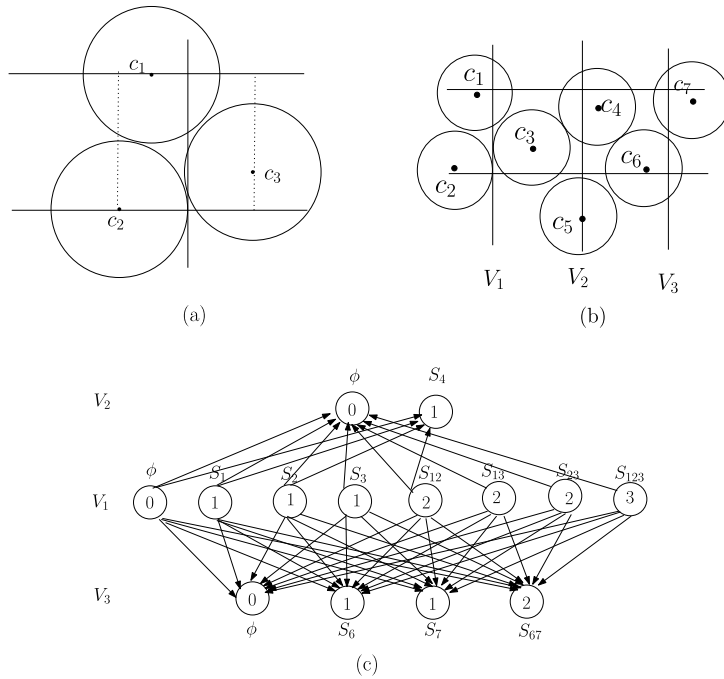


Fig. 3. A geometric layout of penny graph.

of this solution is  $\frac{1}{2}|OPT|$ , where  $OPT$  is a solution of the MIS problem for this penny graph of maximum cardinality.

In order to compute the maximum independent set among the pennies whose centers lie inside a strip, we draw vertical lines at unit distance apart, such that each penny is intersected by a vertical line. Next, we remove those vertical lines which does not intersect any penny.

**Lemma 3.** *A vertical line can intersect at most 4 pennies whose centers are inside a horizontal strip of unit width.*

**Proof.** We will prove the lemma by proving that a vertical line can intersect at most two pennies having their centers in the same side of a vertical line inside a strip of unit width.

For a contradiction, let us assume that there exists more than 2 such pennies which intersect a vertical line  $\ell$ , and whose centers lie in the same side of  $\ell$ . Thus, their centers lie in a rectangle of size  $\frac{1}{2} \times 1$  whose horizontal sides are aligned with the boundaries of the strip, and one of the vertical sides is aligned with the line  $\ell$ . Consider a horizontal line segment that splits this rectangle into two squares of size  $\frac{1}{2} \times \frac{1}{2}$ . By pigeonhole principle, one of these squares must contain the centers of at least two pennies. Thus the distance of the centers of these two pennies is at most  $\frac{1}{\sqrt{2}} < 1$ . This is impossible since two pennies cannot properly intersect.  $\square$

Fig. 3(a) demonstrates an example with 3 pennies whose centers lie in a strip and are intersected by a vertical line.

Suppose we have  $r$  vertical lines inside a strip as mentioned above. We then consider an  $(r + 2)$ -partite node

weighted digraph where the set of nodes in the  $i$ -th set, denoted by  $V_i$ , consists of all possible independent sets of disks among those intersected by the  $i$ -th vertical line,  $i = 1, 2, \dots, r$ . By Lemma 3, the number of nodes in any  $V_i$  can be at most 16. The weight of a node is equal to the number of non-intersecting disks it represents. Further,  $V_0$  and  $V_{r+1}$  respectively contain the dummy nodes  $s$  and  $t$ , each having weight 0. In Fig. 3(c), an instance is shown (without  $s$  and  $t$ ), where the number of nodes corresponding to the vertical lines  $V_1, V_2$  and  $V_3$  are 8, 2 and 4, respectively. The nodes of  $V_1$  are  $\{\phi, S_1, S_2, S_3, S_{12}, S_{13}, S_{23}, S_{123}\}$ , that of  $V_2$  are  $\{\phi, S_4\}$  and that of  $V_3$  are  $\{\phi, S_6, S_7, S_{67}\}$ . For a vertical line  $V_\alpha$ , a node  $S_{ijk\ell}$  represents the non-intersecting pennies  $\{C_i, C_j, C_k, C_\ell\}$  intersected by  $V_\alpha$ . Similarly, nodes  $S_{ijk}, S_{ij}$  and  $S_i$  for  $V_\alpha$  are defined. If there is no penny on  $V_\alpha$ , then we have  $\phi$  for  $V_\alpha$ . Next, there is a directed edge  $\vec{uv}$  between a pair of nodes  $u \in V_i$  and  $v \in V_j$  if  $i < j$  and the disks corresponding to  $u$  and  $v$  are all pairwise non-intersecting.<sup>2</sup> There is no directed edge between a pair of nodes in  $V_i, i = 1, 2, \dots, r$ . We add a directed edge from the node  $s$  to each node in  $\bigcup_{i=1}^r V_i$ , and also a directed edge from each node in  $\bigcup_{i=1}^r V_i$  to the node  $t$ . We also have two fields  $\pi(v)$  and  $\mu(v)$  with each node  $v$ , where  $\pi(v)$  is the predecessor of  $v$  on the longest weighted path from  $s$  to  $v$ , and  $\mu(v)$  indicates the weight of the aforesaid path from  $s$  to  $v$ . The maximum independent set of

<sup>2</sup> In Fig. 3, the edges from nodes in  $V_1$  to nodes in  $V_2$  and  $V_3$  are shown. For the sake of neatness, the edges (i) from nodes in  $V_2$  to those in  $V_3$ , from  $s$  to the nodes in  $V_1, V_2, V_3$  and from the nodes in  $V_1, V_2, V_3$  to  $t$  are not shown.

The center  $c_5$  of the disk  $C_5$  lies outside the strip; so it does not participate in the graph for the current strip.



the penny graph is the longest weighted path from  $s$  to  $t$  in this digraph. This can be computed in polynomial time since the digraph is acyclic [6].

Note that we need not explicitly construct the graph. We start from  $s$  and process the vertices of  $V_i$ ,  $i = 1, 2, \dots, r$  in order. While considering the vertices in  $V_i$ , we assume that each node of  $V_i$  contains the length of the longest path from  $s$  to that node, and we extend the paths from the nodes in  $V_i$  to those in  $V_{i+1}$  and  $V_{i+2}$ . Two points need to be noted here:

- while processing the nodes in  $V_i$ , the reason for choosing the next two consecutive levels  $V_{i+1}$  and  $V_{i+2}$  is that there may not exist an edge among any pair of nodes  $(u, v)$ , where  $u \in V_i$  and  $v \in V_{i+1}$ . But, there must exist edges from  $V_i$  to  $V_{i+2}$ .
- while processing the nodes in  $V_i$ , the reason for not putting any edge between nodes from  $V_i$  to the nodes in the levels  $V_{i+3}, \dots, V_r$  is that, we need not have to consider the transitively oriented edges for computing the longest path in a digraph.

After processing all the nodes in  $V_r$ , the length of the longest path is available at node  $t$ . The exact path also can be reported using the  $\pi$  field of the nodes along the longest path in reverse direction starting from the node  $t$ . Thus, the time complexity of computing the longest path of the graph corresponding to a strip is dominated by sorting of the centers of the pennies inside that strip. We have the following result.

**Theorem 2.** *Given a set of  $n$  pennies in  $\mathbb{R}^2$ , a set of mutually non-intersecting pennies of size at least  $\frac{1}{2}|OPT|$  can be computed in  $O(n \log n)$  time using  $O(n)$  space, where  $OPT$  is the largest subset of mutually non-intersecting pennies present in the plane.*

It needs to be mentioned that if the centers of the pennies have integer coordinates then an *optimum solution* of the MIS problem for the penny graph can be obtained in polynomial time using the following facts:

- As the centers of the unit disks are placed on the vertices of a grid, the corresponding penny graph is planar and bipartite; it can be written as  $G = (V_1 \cup V_2, E)$  where the set of vertices  $V_1$  and  $V_2$  represent the set of disks with centers in even and odd diagonals of the grid, respectively.
- A maximum independent set of this penny graph can be obtained from a maximum bipartite matching [9]; the time complexity is  $O(\sqrt{n}|E|)$  which is  $O(n^{1.5})$  since the graph is planar.

#### 4. Polynomial-time approximation scheme (PTAS)

We use two level shifting strategy of Hochbaum and Maass [10] to propose a PTAS for the MIS problem for a unit disk graph. As in the earlier sections, we split the region into horizontal strips each of width 1. In the first level of shifting strategy, we perform  $k + 1$  phases of execution. In the  $i$ -th phase ( $0 \leq i \leq k$ ), we partition the plane into

blocks; the first block consists of  $i$  consecutive strips from the beginning, and other blocks consist of  $k$  consecutive strips leaving one strip between two consecutive blocks so that solutions of different blocks are non-intersecting. The last block may contain fewer strips (see also [1]).

In a phase of the first level, we compute the solution of each block separately by applying the second level of shifting strategy (see subsection 4.2), add up the solutions of the blocks to get the solution of that phase. The phase producing a solution of maximum cardinality, is reported.

##### 4.1. Optimally computing MIS for unit disks centered in a $k \times k$ square

Let  $\varphi_k$  be the maximum number of non-overlapping unit disks whose centers lie in a  $k \times k$  square region.

**Lemma 4.**  $\varphi_k \leq 9(\lceil \frac{k}{2} \rceil)^2$ .

**Proof.** Follows from the fact that a  $2 \times 2$  square can contain at most 9 non-overlapping unit disks (divide  $2 \times 2$  square into 9 equal cells, the length of each diagonal of these cells is less than 1).  $\square$

Let  $\sigma_k$  be the maximum number of mutually non-overlapping disks whose centers lie in a strip of width  $k > 1$  and intersected by a vertical line  $\ell$ .

**Lemma 5.**  $\sigma_k \leq \frac{7k}{3} + 2$ .

**Proof.** The proof of this lemma is similar to Lemma 3. All the disks whose centers lie in the same side of the vertical line  $\ell$  of the strip and are intersected by the line are within a  $k \times \frac{1}{2}$  rectangle  $R$ . For  $k \leq 6$ , we can show that if we partition  $R$  into  $k + 1$  equal cells by  $k$  horizontal lines, then each of these cells cannot accommodate more than one points. The reason is that the maximum distance between two points residing in a  $\frac{k}{k+1} \times \frac{1}{2}$  rectangle is at most  $\sqrt{\frac{k^2}{(k+1)^2} + \frac{1}{4}} < 1$ , when  $k \leq 6$ . So, when  $k \leq 6$ , the number of mutually non-overlapping disks centered in a strip of width  $k$  and intersected by a vertical line  $\ell$  is at most  $2(k + 1)$ . For  $k > 6$ , we obtain the bound by splitting the  $k \times 2$  rectangle into  $\lfloor \frac{k}{6} \rfloor$  smaller rectangles each of size  $6 \times 2$  and remaining one of smaller size. Thus, in general, we have  $\sigma_k \leq 14 \times \lfloor \frac{k}{6} \rfloor + 2(k - 6 \times \lfloor \frac{k}{6} \rfloor + 1) = 2 \times \lfloor \frac{k}{6} \rfloor + 2k + 2 \leq (\frac{7k}{3} + 2)$ .  $\square$

Let  $m = |C'|$ , where  $C'$  is the set of unit disks whose centers lie in a  $k \times k$  square region. We choose all possible subsets of size  $1, 2, \dots, \varphi_k$  among the set of  $m$  disks. For each choice of size  $i$ , the checking of whether the chosen disks are mutually non-intersecting needs  $O(ik)$  time in the worst case.<sup>3</sup> Thus, the total time is  $O(k^3 m^{\varphi_k})$ . The extra

<sup>3</sup> We sort the chosen disks with respect to their  $x$ -coordinates in  $O(i \log i)$  time in the worst case. Now, the testing of whether a chosen disk  $c$  intersects any other chosen disk  $c'$  needs at most  $O(k)$  time. The reason is as follows: let  $c$  be intersected by a vertical line  $\ell'$ , now we need to check  $c$  with the chosen disks that are intersected by  $\ell'$ , and the

space required for generating the possible subsets is  $O(m)$ . We now describe a faster method.

Consider the middle-most vertical line  $\ell$  in the region  $R$ . Let it intersect a subset  $S \subseteq \mathcal{C}'$ , where  $|S| \leq m$  in the worst case. As the maximum number of mutually non-intersecting members of  $S$  intersecting  $\ell$  is  $\sigma_k$ , we consider all possible mutually non-intersecting subsets of disks of sizes  $0, 1, 2, \dots, \sigma_k$ . The number of such subsets can be at most  $O(m^{\sigma_k})$ . For each subset  $S' \subseteq S$ , compute a non-intersecting set of disks of maximum cardinality among the members in  $\mathcal{C}'$  that contains the members of  $S'$  as stated below. Finally, the one having maximum size is reported.

While processing a subset  $S' \subseteq S$ , we remove the disks in  $S'$ , and those in  $\mathcal{C}'$  which are intersected by the members of  $S'$ . The remaining disks in  $\mathcal{C}'$  are partitioned into two disjoint subsets to the left and right of  $\ell$ . We compute the optimum solution of these two subsets independently using the same procedure recursively. The centers of each subset of disks is now contained in a rectangle of size  $k \times \lfloor \frac{k}{2} \rfloor$ .

We maintain an array containing the disks in  $\mathcal{C}'$  in sorted order of their  $x$ -coordinates. While executing a subproblem in the recursive call, we can identify the subset of disks  $S \in \mathcal{C}'$  that are intersected by the corresponding vertical line  $\ell$  using binary search. The centers of these disks are contiguous in the array  $\mathcal{C}'$ . We need an array of size  $\sigma_k$  to generate all possible combinations of disks of sizes  $1, 2, \dots, \sigma_k$  among the members in  $S$ . A valid subset is the one whose members (i) are mutually non-intersecting, and (ii) do not intersect with the disks in the solution considered in the levels of recursion prior to this level. Thus, in each level of recursion, we need to (i) preserve the array containing the permutation, and (ii) to pass the independent set of disks, say  $\chi$ , chosen till now to its next level, which can be at most  $m$  in number. Again by Lemma 4, this number can be at most  $\varphi_k$ . Thus, considering all the levels of recursion the total space required can be at most  $O(k \log k + \min(\varphi_k, m))$ . The first term corresponds to the extra space required for storing a binary sequence of  $k$  elements for all the levels of recursion.

We use  $T(m, q)$  to denote the time for computing the optimum solution of a rectangle of size  $k \times q$ . Thus, we have  $T(m, q) = O(\log m) + d_0 m^{\sigma_k} \times (d_1 k^2 + d_2 k^3 + d_3 k + 2T(m, \frac{q}{2}))$ , where  $d_0, d_1, d_2, d_3$  are constants. Here,  $O(\log m)$  time is required to identify a disk in  $S$  intersected by the corresponding vertical line  $\ell$ ; the first, second, third and fourth terms inside the parenthesis indicate the time for (i) checking whether the chosen disks are non-intersecting, (ii) checking whether the chosen disks are non-intersecting with the disks in set  $\chi$  chosen in the lower levels of recursion, (iii) inserting the chosen disks in the set  $\chi$ , and (iv) for solving the two subproblems in the next level of recursion. Thus, we have  $T(m, q) = \log m + c_1 k^3 m^{\sigma_k} + c_2 m^{\sigma_k} T(m, \frac{q}{2})$ , where  $c_1$  and  $c_2$  are constants. Solving this recursion, we get  $T(m, k) = O(k^3 m^{\sigma_k \log k})$ . Thus, we have the following results:

**Lemma 6.** *The worst case time and extra space required for computing an optimum solution of the MIS problem for a set of  $m$  disks whose centers lie in a square of size  $k \times k$  is  $O(k^3 m^{\sigma_k \log k})$  and  $O(k \log k + \min(\varphi_k, m))$ , respectively. Here  $\varphi_k \leq 9(\lceil \frac{k}{2} \rceil)^2$  and  $\sigma_k \leq \frac{7k}{3} + 2$ .*

#### 4.2. MIS for unit disk layout of a block of width $k$

Here the objective is to find a set of non-intersecting subset of disks among the set of disks in  $\mathcal{C}_B$  whose centers lie in a block  $B$  of width  $k$  (determined by  $k$  horizontal strips) where  $k > 1$ . We draw vertical lines at unit distance apart and execute  $k + 1$  phases. In the  $i$ -th phase, we consider the first rectangle of size  $i \times k$  and other consecutive  $k \times k$  squares leaving one vertical strip between each of them.<sup>4</sup> In Lemma 6, we show that the optimum solution of the MIS problem for a  $k \times k$  square region  $R$  can be solved in time  $O(k^3 |\mathcal{C}'|^{\sigma_k \log k})$ , where  $\mathcal{C}' \subseteq \mathcal{C}_B$  is the subset of disks whose centers lie inside the square and  $\sigma_k \leq \frac{7k}{3} + 2$ . The same result holds for a rectangle of size  $i \times k$ ,  $i < k$ . This leads to the following result:

**Lemma 7.** *Given a set  $\mathcal{C}_B$  of  $n_B$  disks whose centers lie inside a block  $B$  of width  $k$ , a non-intersecting subset  $IS_B$  of the disks in  $\mathcal{C}_B$  can be computed in  $O(k^4 n_B^{\sigma_k \log k})$  time, such that  $|IS_B| \geq \frac{1}{(1+\frac{1}{k})} |OPT_B|$ , where  $OPT_B$  is a maximum cardinality subset of non-intersecting disks in the sets  $\mathcal{C}_B$  and  $\sigma_k \leq \frac{7k}{3} + 2$ .*

**Proof.** The time complexity follows from the fact that (i) the shifting strategy consists of  $k + 1$  phases, and (ii) the  $k \times k$  squares considered in a phase are disjoint. The analysis of the approximation result is similar to that of the PTAS for fixed height rectangles proposed by Agarwal et al. [1]. Here, in the  $i$ -th phase, we have computed the optimum solution  $IS_B^i$  for the set of disks considered in this phase. If  $\alpha_i$  is the set of all disks in the optimum solution  $OPT_B$  that are intersected by the vertical lines numbered  $\{i + j * k, \forall j = 0, 1, 2, \dots\}$ , then  $|IS_B^i| \geq |OPT_B| - \alpha_i$ . Since  $\min_{i=0}^k \alpha_i \leq \frac{1}{k+1} |OPT_B|$ , we have  $|IS_B| = \max_{i=0}^k |IS_B^i| \geq |OPT_B| - \min_{i=0}^k \alpha_i \geq |OPT_B| - \frac{1}{k+1} |OPT_B| = \frac{1}{(1+\frac{1}{k})} |OPT_B|$ .  $\square$

#### 4.3. Approximation result and complexity

**Theorem 3.** *For a given integer  $k \geq 1$ , the two level nested shifting strategy for solving the MIS problem produces a solution of size  $\frac{1}{(1+\frac{1}{k})^2} |OPT|$ , where  $OPT$  is the subset of disks in an optimum solution. The worst case time and space complexities are  $O(k^4 n^{\sigma_k \log k})$  and  $O(n + k \log k)$ , respectively, where  $\sigma_k \leq \frac{7k}{3} + 2$ .*

**Proof.** The analysis of approximation result is similar to that in Lemma 7. In each phase, instead of computing the optimum solution, we are computing a solution of size

vertical lines previous and next to  $\ell'$  by traversing in the sorted list both towards the left and the right of  $c$ . The number of such disks is  $O(k)$  (see Lemma 5) even if  $i > k$ .

<sup>4</sup> In other words, the  $i$ -th phase consists of all the disks that are not intersected by the  $i + j * k$ -th vertical line, for all  $j = 0, 1, 2, \dots$

$\frac{1}{(1+\frac{1}{k})} |OPT|$ . Thus, after executing the  $k+1$  phases, the reported answer (the solution of maximum size among the  $k+1$  phases) achieves an approximation factor  $(1+\frac{1}{k})^2$ .

As the disks in different blocks are disjoint in a phase, the total time required for executing a phase is  $O(k^3 n^{\sigma_k} \log k)$ . Since we need to execute  $(k+1)$ -phases, the time complexity result follows. The space complexity is justified while explaining the recursion.  $\square$

A similar technique leads to a PTAS for the penny graph also. The complexity results are as follows:

**Theorem 4.** For a given integer  $k \geq 1$ , one can obtain a  $(1+\frac{1}{k})$ -approximation result for penny graph in  $O(2^{2\sigma_k} nk + n \log n)$  time using  $O(2^{\sigma_k} + n)$  space, where  $\sigma_k \leq \frac{7k}{3} + 2$ .

**Proof.** While processing a block  $B$  of width  $k$  containing a set  $C_B$  of disks ( $m = |C_B|$ ) whose centers lie inside the block  $B$ , the sorting of those disk-centers with respect to their  $x$ -coordinates needs  $O(m \log m)$  time. We use similar strategy as in Section 3. While solving this instance, the number of nodes in the graph corresponding to a vertical line can be at most  $2^{\sigma_k}$ . Here processing of a node needs comparing with the nodes on its predecessor and successor vertical line. Since the number of non-empty vertical strips in a block can be  $O(m)$  in the worst case, the time for computing the optimal solution in the block  $B$  of  $m$  vertical strips is  $O(2^{2\sigma_k} m)$ . Since the disks of different blocks are disjoint, the time complexity of a phase is  $O(2^{2\sigma_k} n)$ . Again, since we need to execute  $k+1$  phases, the overall time complexity is  $O(2^{2\sigma_k} nk)$ . Using the same argument as in Section 3, the space complexity is  $O(2^{\sigma_k} + n)$ .

In each phase, as we compute the optimum solution of each block, so the analysis of the approximation factor is similar to that in Lemma 7.  $\square$

## Acknowledgement

The authors wish to acknowledge the anonymous reviewer for the valuable comments that have improved the quality of the paper.

## References

- [1] P.K. Agarwal, M. van Kreveld, S. Suri, Label placement by maximum independent set in rectangles, *Comput. Geom.* 11 (1998) 209–218.
- [2] B.N. Clark, C.J. Colbourn, D.S. Johnson, Unit disk graph, *Discrete Math.* 86 (1990) 165–177.
- [3] M.R. Cerioli, L. Faria, T.O. Ferreira, F. Protti, On minimum clique partition and maximum independent set on unit disk graphs and penny graphs: complexity and approximation, *Electron. Notes Discrete Math.* 18 (2004) 73–79.
- [4] T.M. Chan, Polynomial-time approximation schemes for packing and piercing fat objects, *J. Algorithms* 46 (2003) 178–189.
- [5] T.M. Chan, S. Har-Peled, Approximation algorithms for maximum independent set of pseudo-disks, in: *Proceedings of the 25th Annual Symposium on Computational Geometry*, 2009, pp. 333–340.
- [6] T.H. Cormen, C.E. Lieserson, R.L. Rivest, *Introduction to Algorithms*, Prentice-Hall, 2007.
- [7] T. Erlebach, K. Jansen, E. Seidel, Polynomial-time approximation schemes for geometric intersection graphs, *SIAM J. Comput.* 34 (2005) 1302–1323.
- [8] M.C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Elsevier, 2004.
- [9] M.R. Garey, D.S. Johnson, *Computers and Intractability: a Guide to the Theory of NP-Completeness*, Freeman, 1979.
- [10] D.S. Hochbaum, W. Maass, Approximation schemes for covering and packing problems in image processing and VLSI, *J. ACM* 32 (1985) 130–136.
- [11] E.J. van Leeuwen, Approximation algorithms for unit disk graphs, Technical Report UU-CS-2004-066, Institute of Information and Computing Sciences, Utrecht University, 2004.
- [12] T. Matsui, Approximation algorithms for maximum independent set problems and fractional coloring problems on unit disk graphs, in: *Proceedings of the Japan Conference on Discrete and Computational Geometry*, JCDG 1998, in: LNCS, vol. 1763, 2000, pp. 194–200.
- [13] M. Marathe, H. Brey, H.B. Hunt III, S.S. Ravi, D.J. Rosenkrantz, Simple heuristics for unit disk graphs, *Networks* 25 (1995) 59–68.