A New Approximation Algorithm for Maximal Independent Set in Wireless Sensor Networks

M. Ghanbaria[1], D.A. Mojdeh[b], M. Ramezani[a]

[a] Department of Mathematics, University of Tafresh, Tafresh, Iran

[b] Department of mathematics, University of Mazandaran, Babolsar, Iran

Abstract

The unit disk graph is a mathematical model for wireless sensor networks when all sensors have the same communication radius. There are two classical optimization problems on graphs, relevant to unit disk graph, models of mobile ad hoc networks, maximal independent set of nodes, that is also a dominating set, and minimum connected dominating set.

We propose decomposition of connected unit disk graph into 1-by-1 boxes and two new matrices corresponding to this graph (Packing matrix and Independent vertex matrix) for approximating maximal independent set. If the graph is bounded, then the considered problems can be solved in polynomial time. We prove this fact indirectly by presenting dynamic programming algorithm and show that these results are optimal.

2010 AMS Subject Classifications: 68R10, 05C69

Keywords and Phrases:

Wireless network, independent set, dominating set, algorithm.

1. Introduction and preliminaries

- As it will be shown, a mobile ad hoc network can be naturally modeled as a unit disk graph. Each node in such a graph has a disk around it containing all points reachable by that node. The intersections of these disks then determinate the edges of the graph. Unit disk graphs are special kind of geometric intersection graphs.

Definition 1.1.

Let S be a set of geometric objects. Then the graph G=(V,E), where each vertex corresponds to an object in S and two vertices are connected by an edge if and only if the two corresponding objects intersect, is called an intersection graph. The graph G is said to be realized by S. Tangent objects are assumed to intersect.

Definition 1.2.

A graph G is a disk graph if and only if there exists a set of disks $D = \{D_i | i = 1,..., n\}$ *such that G is the intersection graph of D. The set of disks is called a disk representation of G.*

Definition 1.3.

A graph G is a unit disk graph (UDG) if and only if G is a disk graph and the radius of a set of disks realizing G are equal.

Usually the common radius is 1, but often it is assumed to be $\frac{1}{2}$. Note that any common radius can be obtained by scaling *D* appropriately.

Definition 1.4.

Let G=(V,E) be a graph. A set S⊆V is an independent set, if there are no u,v ∈S, such that (u,v) ∈E. A set S⊆V is a vertex cover, if for each (u,v) ∈E it holds that u ∈S or v ∈ S.

Definition 1.5.

Let G=(V,E) be a graph. A set S⊆V is a dominating set, if for each vertex v either v ∈S or there exists a vertex u ∈S for which (u,v) ∈E.

Definition 1.6.

Let G=(V,E) be a graph. A set S ⊆ V is a connected dominating set, if S is a dominating set and the subgraph of G induced by S (G[S]) is connected.

1.1 Problems and previous works

- In wireless ad hoc networks, a connected dominating set (CDS) has been extensively used as a virtual backbone for routing. The majority of approximation algorithms for constructing a minimum connected dominating Set (MCDS) in wireless ad hoc networks follow a general two-phased approach. The first phase is to construct a dominating set (DS) and the second phase is to connect the nodes in it. Generally, in the first phase a maximal independent set (MIS) is used as the DS. The relation between the size of MIS and MCDS plays the key role in

Manuscript received June 5, 2017
Manuscript revised June 20, 2017

the performance analysis of these two-phased algorithms. Islam et al. [3], proposed a distributed algorithm with constant performance ratio of 38 for CDS. This algorithm does not find MIS of the whole network. Firstly it constructs a small connected subgraph of the network, and then finds MIS of the subgraph, finally, connects the MIS through other nodes. Bourgeois et al. [1], presented a new bound on the number of MIS for a given size in triangle-free and bipartite graphs. Surendran et al. [8], employed a distributed algorithm to compute CDS in a wireless network with the help of spectral network. Das et al. [2], proposed an algorithm with three phases, in the first phase, an algorithm is proposed for finding MCDS using MIS. In the second phase, an algorithm is written for data gathering and in the third phase, they tried to minimize the data transmission power consumption among the sensor nodes. Mohanty et al. [6], proposed a new degree-based greedy approximation algorithm named as connected pseudo dominating set using two hop information, which reduces the CDS size as much as possible. Rai et al. [7], studied an energy efficient MCDS construction algorithm. The algorithm is divided into three phases. Firstly, it finds DS and selects a node with the maximum degree as a dominator. Then connects the DS through a Steiner tree. Finally, it prunes the obtained CDS to get MCDS. Yu et al. [11], introduced the diameter of CDS as a new quality factor for CDS construction algorithms. These two algorithms first find MIS of the graph, and then connect the nodes in MIS to form CDS. Yu et al. [10], proposed and analyzed a distributed synchronous algorithm for constructing CDS. Wightman and Labrador [9], presented a family of distributed topology construction algorithms based on CDS. Kamei et al. [4], studied a self-stabilizing fully distributed algorithm with a safe convergence for MCDS in the networks modeled by UDG. Leeuwen et al. [5] presented two new notions for unit disk graphs, i.e., thickness and density. The thickness of a graph is the number of disk centers in any width 1 slab. If the thickness of a graph is bounded, then the considered problems can be solved in polynomial time. They proved this both indirectly by presenting a relation between unit disk graphs of bounded thickness and the pathwidth of such graphs, and directly by giving dynamic programming algorithms. They then considered unit disk graphs of bounded density. The density of a graph is the number of disk centers in any 1-by-1 box. They presented a new approximation scheme for the considered problems, which uses the bounded thickness results mentioned above and the so called shifting technique.

2 Approximation of maximal independent set

- Throughout this paper, we assume, a unit disk graph G=(V,E) is given with a known unit disk representation D={D(vi)| its radius is 12 and vi=(xi,yi) for i=1,¼,n}

- Since the given unit disk graph is bounded and finite, we let

- m=ëmax(xi)-min(xi)û+1; i=1,2,…,n

- k=ëmax(yi)-min(yi)û+1; i=1,2,…,n

- Therefore unit disk graph is contained in a k-by-m box or in boundaries of this box (see Figure. 1).

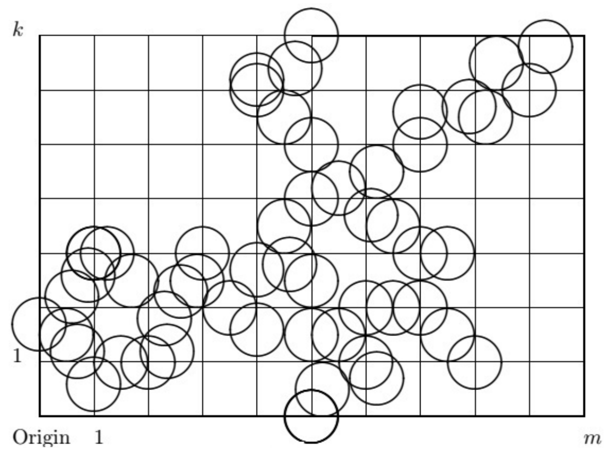
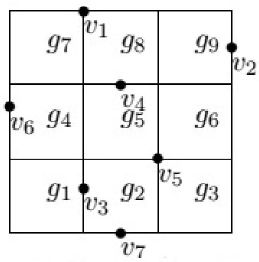


Fig. 1 The k-by-m box

- Now we transform the origin of coordinate to left bottom of k-by-m box and rewrite coordinate of vertices in the new system.

- Now we partite the plane with finite vertical and horizontal lines. The vertical lines intersect the x-axis at 1,2,3,…,m and the horizontal lines intersect the y-axis at 1,2,3,…,k. Therefore the side of every grid square is equal to 1. The lines are called grid boundaries. A disk $D(v_i)$ is a member of a grid which its center is in the grid (see Figure. 1).

- If a vertex lies on the right boundary of a grid, then the disk is considered to belong to the grid to the right of the boundary unless there is not a grid to the right of the boundary. If a vertex lies on bottom boundary of a grid, then the disk is considered to belong to the grid to the bottom of the boundary unless there is not a grid to the bottom of boundary. If a vertex lies on right bottom corner of the current grid, then the disk is considered to belong to the grid to the right bottom corner of the current grid unless there is not a grid to the right down corner. For example, see Figure 2.



- $v_3, v_7 \in g_2, v_5 \in g_3, v_6 \in g_4, v_4 \in g_5,\ v_1 \in g_8,\ v_2 \in g_9$

Fig. 2

Now we are ready to define "packing matrix $T_{k'm}$", "independent vertex matrix $InD_{k'm}$" and "neighborhood matrix $N_{k'm}$". First $N_{k'm}$, $T_{k'm}$ and $InD_{k'm}$ are initialized to zero matrices and start from the origin (left bottom corner of k-by-m box). For every i=1,2, ,n, if $D(v_i)$ lies in pth row and qth column of k-by-m box, and $N[p,q] \le deg(v_i)$,then let $T[p,q]$=1, $InD[p,q]=v_i$ and $N[p,q]=deg(v_i)$ .Therefore if there are some vertices $v_{i_1}, v_{i_2}, \cdots, v_{i_s}, i_1<i_2<\cdots<i_s$ on the grid of pth row and qth column of k-by-m box, then finally we have $T[p,q]$=1 and $InD[p,q]=v_{i_t}, (1 \le t \le s)$ such that $deg(v_{i_t})$ is greater than or equal to $deg(v_{i_j}), 1 \le j \le s$. Corresponding to all grids that include no vertices, zero in packing matrix and independent vertex matrix are inserted. Now the matrix $InD_{k'm}$ presents the thinly scattered graph G' of initial graph. Next stage, we continue to find a maximal independent set of G.

If v_{i_p} be a vertex of G' such that $deg_{G'}(v_{i_p})$=1 we select the v_{i_p} as a member of MIS and delete the vertex v_{i_q} of G' such that $D(v_{i_p}) \cap D(v_{i_q}) \neq \phi$. Then we select vertexes v_{i_p} of G' such that $deg_{G'}(v_{i_p})$=2 and remove their neighborhoods. Since maximum degree of G' is 8, by continuing this process, finally $InD_{k'm}$ presents a maximal independent set of G. It is obvious that, the approximation of independence number is calculated too. These lead to algorithm 2.1 for computing a maximal independent set of G.

Algorithm 2.1. *Grid Decomposition MIS(G)*

Input: A unit disk graph G with the coordinates of the vertices $v_i = (x_i, y_i), 1 \le i \le n$.

Output: A maximal independent set of G

1\. Set $k = \lfloor max(y_i) - min(y_i) \rfloor + 1,\ m = \lfloor max(x_i) - min(x_i) \rfloor + 1$

2\. Set $T(k,m) = O_{k\times m},\ InD(k,m) = O_{k\times m},\ N(k,m) = O_{k\times m}$

3\. $x' = min(x_i), i = 1,2,\cdots,n, \quad y' = min(y_i), i = 1,2,\cdots,n$

4\. For $i = 1,2,\cdots,n$

4.1 $\quad x_i = x_i - x' \quad y_i = y_i - y'$

4.2 End For

5\. For $i = 1,2,\cdots,n$

6\. If $\lfloor y_i \rfloor \neq y_i$ then

6.1. If $\lfloor x_i \rfloor + 1 \le m$ then If $N[\lfloor y_i \rfloor + 1, \lfloor x_i \rfloor + 1] \le deg(v_i)$ then

6.2. $T[\lfloor y_i \rfloor + 1, \lfloor x_i \rfloor + 1] = 1,\ InD[\lfloor y_i \rfloor + 1, \lfloor x_i \rfloor + 1] = v_i,\ N[\lfloor y_i \rfloor + 1, \lfloor x_i \rfloor + 1] = deg(v_i)$

6.3. else If $N[\lfloor y_i \rfloor + 1, \lfloor x_i \rfloor] \le deg(v_i)$ then

6.4 $T[\lfloor y_i \rfloor + 1, \lfloor x_i \rfloor] = 1,\ InD[\lfloor y_i \rfloor + 1, \lfloor x_i \rfloor] = v_i,\ N[\lfloor y_i \rfloor + 1, \lfloor x_i \rfloor] = deg(v_i)$

6.5. End If

7\. else If $\lfloor y_i \rfloor = y_i$ then

7.1. If $\lfloor x_i \rfloor + 1 \le m$ then If $N[y_i, \lfloor x_i \rfloor + 1] \le deg(v_i)$ then

7.2 $T[y_i, \lfloor x_i \rfloor + 1] = 1,\ InD[y_i, \lfloor x_i \rfloor + 1] = v_i,\ N[y_i, \lfloor x_i \rfloor + 1] = deg(v_i)$

7.3. else If $N[y_i + 1, \lfloor x_i \rfloor] \le deg(v_i)$ then

7.4 $T[y_i + 1, \lfloor x_i \rfloor] = 1,\ InD[y_i + 1, \lfloor x_i \rfloor] = v_i,\ N[y_i + 1, \lfloor x_i \rfloor] = deg(v_i)$

7.3. *End If*

8. *End For*

9. *For* $i = 1, 2, \cdots, k$

10. *For* $j = 1, 2, \cdots, m$

11. *If* $InD[i, j] \neq 0$

12. $N[i, j] = deg_{G'}(InD[i, j])$ *else* $N[i, j] = -1$

13. *End For*

14. *End For*

15. *For* $l = 1, 2, \cdots, 8$

16. *For* $i = 1, 2, \cdots, k$

17. *For* $j = 1, 2, \cdots, m$

18. *If* $N[i, j] = l$ *then*

19. *If* $j + 1 \leq m$ *and* $T[i, j + 1] = 1$ *and* $D(InD[i, j]) \cap D(InD[i, j + 1]) \neq \emptyset$

19.1. $T[i, j + 1] = -1, \quad InD[i, j + 1] = -1, \quad N[i, j + 1] = -N[i, j + 1]$

19.2. *End If*

20. *If* $i + 1 \leq k$ *and* $j + 1 \leq m$ *and* $T[i + 1, j + 1] = 1$ *and* $D(InD[i, j]) \cap D(InD[i + 1, j + 1]) \neq \emptyset$

20.1. $T[i + 1, j + 1] = -1, \quad InD[i + 1, j + 1] = -1, \quad N[i + 1, j + 1] = -N[i + 1, j + 1]$

20.2. *End If*

21. *If* $i + 1 \leq k$ *and* $T[i + 1, j] = 1$ *and* $D(InD[i, j]) \cap D(InD[i + 1, j]) \neq \emptyset$

21.1. $T[i + 1, j] = -1, \quad InD[i + 1, j] = -1, \quad N[i + 1, j] = -N[i + 1, j]$

21.2. *End If*

22. *If* $i + 1 \leq k$ *and* $j - 1 \geq 1$ *and* $T[i + 1, j - 1] = 1$ *and* $D(InD[i, j]) \cap D(InD[i + 1, j - 1]) \neq \emptyset$

22.1. $T[i + 1, j - 1] = -1, InD[i + 1, j - 1] = -1, \quad N[i + 1, j - 1] = -N[i + 1, j - 1]$

22.2. *End If*

23. *If* $j - 1 \geq 1$ *and* $T[i, j - 1] = 1$ *and* $D(InD[i, j]) \cap D(InD[i, j - 1]) \neq \emptyset$

23.1. $T[i, j - 1] = -1, InD[i, j - 1] = -1, \quad N[i, j - 1] = -N[i, j - 1]$

23.2. *End If*

24. *If* $i - 1 \geq 1$ *and* $j - 1 \geq 1$ *and* $T[i - 1, j - 1] = 1$ *and* $D(InD[i, j]) \cap D(InD[i - 1, j - 1]) \neq \emptyset$

24.1. $T[i - 1, j - 1] = -1, InD[i - 1, j - 1] = -1, \quad N[i - 1, j - 1] = -N[i - 1, j - 1]$

24.2. *End If*

25. *If* $i - 1 \geq 1$ *and* $T[i - 1, j] = 1$ *and* $D(InD[i, j]) \cap D(InD[i - 1, j]) \neq \emptyset$

25.1. $T[i - 1, j] = -1, InD[i - 1, j] = -1, \quad N[i - 1, j] = -N[i - 1, j]$

25.2. *End If*

26. *If* $i-1 \geq 1$ *and* $j+1 \leq m$ *and* $T[i-1,j+1]=1$ *and* $D(InD[i,j]) \cap D(InD[i-1,j+1]) \neq \emptyset$

26.1. $T[i-1,j+1]=-1,\ InD[i-1,j+1]=-1,\quad N[i-1,j+1]=-N[i-1,j+1]$

26.2. *End If*

27. *End For*

28. *End For*

29. End For

30.Return $InD_{k\times m}$

Theorem 2.2. *The Algorithm 2.1 returns a maximal independent set of G.*

Proof. In the output of Algorithm 2.1, matrix $InD_{k´m}$ has elements 0 or 1 or v_{i_p} for an index $1£i_p£n$.

(1) If $InD[i,j]=0$, in fact there exists no vertex $v_p=D(v_p),v_p=(x_p,y_p)$ such that $i<y_p£i+1$ and $j£x_p<j+1$.

(2) If $InD[i,j]=-1$, in fact for its corresponding vertex, there exists another vertex such that their intersection is not empty.

(3) $InD[i,j]=v_{i_p}, 1£i_p£n$, and intersection between such vertices are empty. Then output is an independent set of G.

For proof of maximality of output, let the algorithm 2.1 return $I=\{v_{i_1},v_{i_2},\cdots,v_{i_q}\}$ and there exists a vertex v_{i_t} such that v_{i_t} is not member of I and $I \cup \{v_{i_t}\}$ is independent set of G. Let $v_{i_t}=D(v_{i_t}),v_{i_t}=(x_{i_t},y_{i_t})$, then there exists a grid g_{i_t} such that $v_{i_t} \in g_{i_t}$. In order to the definition of packing matrix and that v_{i_t} is not a member of I, its corresponding element in packing matrix is only □ 1, in fact there exists another vertex such that their intersection is not empty. Therefore the Algorithm 2.1 returns a maximal independent set of G. ■

Theorem 2.3 *Let the initial graph be order n. Then maximum the time complexity of Algorithm 2.1 is O(max{8k.m,n}).*

Proof. The time complexity of Algorithm 2.1 is depends on how sparse and distanced the distribution of disks in the plane. In other hand there exist three nested loops. Therefore maximum the time complexity of Algorithm 2.1 is $O(max\{8k.m,n\})$. For instance, take two disks of radius $\frac{1}{2}$ whose centers are given by $v_1=(0,0)$ and $v_2=(10,10)$. In this case, $k=m=11$ and $n=2$. The time complexity of Algorithm 2.1 for this particular case is $O(8k.m)$. However, if all disks lie on the first grid ([1,1]) and we have 100 disks, then $k=m=1$ and $n=100$, and as a consequence, the time complexity of Algorithm 2.1 for this particular case is $O(n)$. ■

3 Two Examples

Example 1 *Suppose G=(V,E) be a unit disk graph such that*

$D=\{v_i=D(v_i),v_i=(x_i,y_i)|i=1,2,3\}$

and theirs representations are given by following coordinates:

v1=(0.5,0.5),v2=(1.5,0.5),v3=(0.5,1.5)

In this case k=m=2. Algorithm 2.1 yields the following results for lines between 1 and 8.

$$T=\begin{bmatrix}1 & 1\\ 1 & 0\end{bmatrix}, \qquad InD=\begin{bmatrix}v_1 & v_2\\ v_3 & 0\end{bmatrix},$$

Finally, Algorithm 2.1 provides the following result for remaining lines.

$$InD=\begin{bmatrix}-1 & v_2\\ v_3 & 0\end{bmatrix}$$

Example 2 *Suppose G=(V,E) be a unit disk graph such that*

$D=\{v_i=D(v_i),v_i=(x_i,y_i)|i=1,...,12\}$

and theirs representations are given by following coordinates:

$$v_1=(0.5,3.2),\quad v_2=(1.3,3.4),\quad v_3=(0.7,3.2),\quad v_4=(1.6,2.7),\quad v_5=(0.4,2.7),$$
$$v_6=(1.3,2.2),\quad v_7=(0.3,1.7),\quad v_8=(0.5,1.9),\quad v_9=(2,1.9),\quad v_{10}=(0.8,1.3),$$
$$v_{11}=(1.5,1.3),\quad v_{12}=(1.4,0.5),$$

Now in terms of the lines 1,2,…,8 of Algorithm 2.1, the following results are obtained:

k=3,m=2

$$T = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad N = \begin{bmatrix} 4 & 4 \\ 4 & 4 \\ 3 & 3 \end{bmatrix}, \quad InD = \begin{bmatrix} v_{10} & v_{11} \\ v_8 & v_6 \\ v_5 & v_4 \end{bmatrix}$$

In terms of the lines 9,10,…,14 of Algorithm 2.1, we have the following result:

$$N = \begin{bmatrix} 2 & 2 \\ 3 & 3 \\ 1 & 1 \end{bmatrix}$$

Finally Algorithm 2.1 provides the following result for remaining lines.

$$InD = \begin{bmatrix} v_{10} & -1 \\ -1 & -1 \\ v_5 & v_4 \end{bmatrix}$$

References

[1] N. Bourgeois, F. Della Croce, B. Escoffier, V.Th. Paschos, Fast algorithms for min independent dominating set, Discrete Applied Mathematics 161, (2013), pp. 558–-572.

[2] S. Das, S. Barman, J. D. Sinha, Energy efficient routing in wireless sensor network, Procedia Technology 6, (2012), pp. 731–738.

[3] K. Islam, S. Akl, H. Meijer, A constant factor distributed algorithm for computing connected dominating sets in wireless sensor networks, in: Proceedings of the 14th IEEE International Conference on Parallel and Distributed Systems (ICPADS'08), December (2008), pp. 559–566.

[4] S. Kamei, H. Kakugawa, A self-stabilizing 6-approximation for the minimum connected dominating set with safe convergence in unit disk graphs, Theoretical Computer Science 428, (2012), pp. 80–90.

[5] M. Tahmassebpour, "A New Method for Time-Series Big Data Effective Storage," in IEEE Access, vol. 5, pp. 10694-10699, 2017.

[6] J. P. Mohantya, C. Mandal, C. Reade, A. Das, Construction of minimum connected dominating set in wireless sensor networks using pseudo dominating set, Ad Hoc Networks 42, (2016), pp. 61–73.

[7] M. Esfahani, M. Emami, H. Tajnesaei, (2013). The investigation of the relation between job involvement and organizational commitment. Management Science Letters, 3(2), 511-518.

[8] S. Surendran, S. Vijayan, Distributed Computation of Connected Dominating Set for Multihop Wireless Networks, Procedia Computer Science 63, (2015), pp. 482–487.

[9] P. M. Wightman, M.A. Labrador, A family of simple distributed minimum connected dominating set-based topology construction algorithm, Journal of Network and Computer Applications 24, (2011), pp. 1997–2010.

[10] J. Yu , L. Jia, D. Yu, G. Li, X. Cheng, Minimum connected dominating set construction in wireless networks under the beeping model, IEEE Conference on Computer Communications, (2015), pp. 972–980.

[11] J. Yu, N. Wang, G. Wang, Heuristic algorithms for constructing connected dominating sets with minimum size and bounded diameter in wireless networks, in: Proceedings of Wireless Algorithms, Systems, and Applications (WASA2010), Lecture Notes in Computer Science, vol. 6221, (2010), pp. 11–20.