A Simple Improved Distributed Algorithm for Minimum CDS in Unit Disk Graphs

Stefan Funke*, Alexander Kesselman†, Ulrich Meyer†, Michael Segal‡
* Computer Science Department, Stanford University, Stanford, U.S.A.
Email: sfunke@stanford.edu
†Max Planck Institut für Informatik, Saarbrücken, Germany
Email: akessel,umeyer@mpi-sb.mpg.de
‡ Communication Systems Engineering Dept., Ben Gurion University, Beer-Sheva, Israel
Email: segal@cse.bgu.ac.il.

***Abstract*— Several routing schemes in ad hoc networks first establish a virtual backbone and then route messages via backbone nodes. One common way of constructing such a backbone is based on the construction of a minimum connected dominating set (CDS). In this paper we present a very simple distributed algorithm for computing a small CDS. Our algorithm has an approximation factor of at most 6.91, improving upon the previous best known approximation factor of 8 due to Wan et al. [INFOCOM'02]. The improvement relies on a refined analysis of the relationship between the size of a maximal independent set and a minimum CDS in a unit disk graph. This subresult also implies improved approximation factors for many existing algorithm.**

I. Introduction

Wireless ad hoc networks appear in a wide variety of applications, including military battle-field, disaster relief, sensing and monitoring. Unlike wired networks, no physical backbone infrastructure is installed in wireless ad hoc networks. Instead, the nodes communicate either directly or via intermediate nodes. In this paper we assume that all nodes are located in a Euclidean plane and have an equal transmission range of 2. The topology of such a network can be modeled as a unit-disk graph $G = (V, E)$. Two nodes are adjacent if the unit disks centered at them intersect, i.e., their inter-distance is at most two. (Note: We use this definition instead of the commonly used one based on a transmission range of 1 or equivalently intersecting disks of radius 0.5 to simplify calculations in the proofs. It should be obvious that by scaling our results also hold for any other definition of the unit-disk graph, be the 'unit' 1,2, or any other constant.)

Although a wireless ad-hoc network has no physical backbone infrastructure, a virtual backbone can be formed by nodes in a connected dominating set (CDS) of G. A CDS of G is a subset $S \subseteq V$ such that each node in $V \setminus S$ is adjacent to some node in S and the communication graph induced by S is connected. We denote by OPT a minimum CDS in G.

The problem of finding a minimum CDS in a unit disk graph has been shown to be NP-hard [4]. The work in [7] proposes a 10-approximation centralized algorithm for this problem. The work in [3] presents a polynomial-time approximation scheme that guarantees an approximation factor of $(1 + 1/s)$ with running time of $n^{O((s \log s)^2)}$. However, *centralized* algorithms cannot be applied to real networks. Recently, the *distributed* construction of a small CDS has attracted a great deal of attention. The currently best known distributed algorithm due to [8] has an approximation factor of 8 and running time $O(n)$. However, the analysis of [8] ignores delays incurred by interference. An algorithm recently presented in [6] with high probability computes a 192-approximation in $O(n \log n)$ time and explicitly handles interference. The algorithm of [6] is based on a distance-2-coloring (D2-coloring), where no two nodes at 2-hop distance can have the same color.

In this paper we present a very simple 6.91-approximation algorithm for computing a minimum CDS in unit disk graphs. That improves upon the previous best known approximation factor of 8 due to [8]. Ignoring interference, our algorithm matches the $O(n)$ running time of the algorithm in [8], but we provide a more detailed analysis also taking care of interference by using the D2-coloring of [6]. The main contribution of this paper is an improved analysis of the relationship between the size of a maximal independent set and a minimum CDS in a unit disk graph, which yields better bounds for many previous algorithms [1], [2], [6], [7], [8]. Note that a maximal independent set is also a dominating set, which only needs to be connected to obtain a CDS.

The rest of the paper is organized as follows. The distributed CDS algorithm is presented in Section II. In Section III we bound the size of any independent set in terms of the minimum CDS size. Section IV contains some concluding remarks.

II. A Simple Distributed Algorithm for CDS

In this section we present a very simple distributed algorithm computing a CDS of G. We assume that there exists an assignment of time slots to the nodes such that no interference occurs, i.e. no two nodes transmit in the same time slot.

The first author was supported by the Max Planck Center for Visual Computing and Communication (MPC-VCC) funded by the Federal Ministry of Education and Research of the Federal Republic of Germany (FKZ 01IMC01).

The second author has done a part of this work while visiting the Dipartimento di Scienze dell'Informazione at Universita di Roma La Sapienza, Italy. Supported in part by AvH-Stiftung, EYES project and MIUR grant FIRB WebMinds.

Such an assignment can be determined using the D2-coloring algorithm from [6]. Let us denote by $q \leq n$ be the number of different time slots in this assignment.

In the course of our algorithm, we construct a connected set S and an independent set $I \subseteq S$. In a nutshell, we color a node (without connection to D2-coloring) with the following colors: *black* – the node is a part of I; *blue* – it is not in S but adjacent to a node in I; *grey* – it is in S but not in I, *red* – it is neither black, grey, nor blue, but a neighbor to a grey or blue node; and *white* – it is neither black nor grey nor blue, nor a neighbor to a grey or blue node. Initially, one node is colored red (this node can be chosen by running a leader election algorithm) and all other nodes are colored white. Each red node u (except the first one) keeps its parent grey node.

The execution of our algorithm is divided into *rounds*. Each round consists of three *phases* and in each phase we use a conflict-free time slots assignment so that each node is able to transmit once. Basically, in a round each red node with minimum ID among its red neighbors joins I and its blue parent joins S. Then the colors of the relevant nodes are updated accordingly. The algorithm is presented on Figure 1.

A. Analysis

The algorithm terminates when there remain no white or red nodes. Next we state the main theorem.

Theorem 2.1: Our algorithm computes a connected dominating set S in G with $|S| \leq 6.91 \cdot |\mathrm{OPT}| + 16.58$ and has running time $O(|OPT| \cdot q)$ and message complexity $O(|OPT| \cdot n)$.

Proof: The fact that the final set S (black and grey nodes) is indeed a CDS, can be easily established by verifying the following invariants maintained throughout the execution of our algorithm:

1) The set of black nodes I form an independent set in G and dominate the set of grey and blue nodes;
2) The subgraph induced by black and grey nodes is connected;
3) If the set of black nodes does not form a dominating set of G, there is at least one red node;
4) In each round at least one red node turns black.

Note that $|S| \leq 2|I|$. That is due to the fact that each grey node u can be associated with a unique black node v s.t. u was parent of v. As we show in Section III, $|I| \leq 3.453 \cdot |\mathrm{OPT}| + 8.291$, which implies the bound on $|S|$.

The running time of our algorithm is $O(|\mathrm{OPT}| \cdot q)$, since each phase lasts q time steps (according to the number of time-slots obtained from the D2-coloring) and we have $O(|OPT|)$ rounds (with three phases each) as in each round at least one node gets colored *red*. The message complexity is $O(|\mathrm{OPT}| \cdot n)$ because during each phase at most n messages are sent (the message size is $O(\log n)$ bits). ■

This bound *includes* delays due to interference between nearby nodes, as this is treated *explicitely* by the D2-coloring. Note that $q = O(n)$ and $|OPT| = O(n)$. Ignoring interference as in the model of [8], we get the following corollary:

Corollary 2.2: Ignoring interference our algorithm computes a connected dominating set S in G with $|S| \leq 6.91 \cdot |\mathrm{OPT}| + 16.58$ and has running time $O(n)$ and message complexity $O(n^2)$.

Furthermore, if the density of the nodes is high (i.e. large value of q), typically $|OPT|$ is rather small, and vice versa, if the optimal $|OPT|$ is large, the node density is typically small. So even *including* the explicit treatment of interference we expect the running time of our algorithm to be very competetive.

III. Bounding the Size of an Independent Set

In this section we bound the size of any independent set in G with respect to the size of OPT. For that we first bound the area covered by the union of unit disks in G.

Theorem 3.1: The area covered by the union of unit disks in G is at most $|\mathrm{OPT}| \cdot 11.774 + 9\pi$.

Proof: Consider the set L of disks with radius 3 around the centers of unit disks from OPT. Clearly, all unit disks corresponding to the nodes of G must be contained in the union of the disks in L and this union is connected. To bound the area covered by L, we mimic the growth of a spanning tree of OPT. In iteration i, we add a new disk l_i whose center has distance at most 2 to a center of the already added disks $l_1, \ldots, l_{i-1}$. We consider the area 'newly covered' by l_i, i.e., the area not covered by the union of disks $l_1, \ldots, l_i$.

Note that the center of l_i is at distance at most 2 from the center of a l_j s.t. $j < i$. The newly covered area is thus at most the hatched area on Figure 2, where $|c_j m| = |c_i m| = 1$. Let $\alpha = \angle i_1 c_j m$ be the angle spanned by i_1 and m at c_j. We have $\cos\alpha = 1/3$ and $|m i_1 = |m i_2| = \sqrt{8} = 2\sqrt{2}$. The hatched area can then be computed by considering a $2\pi - 2\alpha$ sector of D_i, subtracting a 2α sector of D_j and adding the area of the diamond $i_1 c_j i_2 c_i$. Hence, we get:

$$
\begin{aligned}
A_h &= \frac{2\pi - 2\alpha}{2\pi} 9\pi - \frac{2\alpha}{2\pi} 9\pi + 4 \cdot \frac{1}{2} \cdot 1 \cdot 2\sqrt{2} \\
&= (\pi - 2\arccos(1/3)) \cdot 9 + 4 \cdot \sqrt{2} \\
&\leq 11.774
\end{aligned}
$$

Therefore, the total area covered by L is at most $|\mathrm{OPT}| \cdot 11.774 + 9\pi$. ■

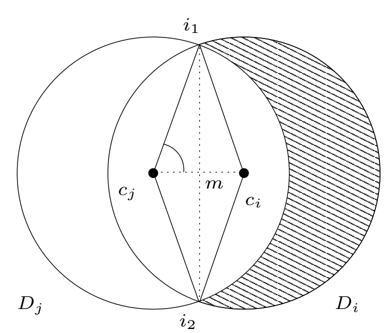

Fig. 2. Arguing about the covered area.

I. APPLY PHASE: Each *red* node sends an APPLY-MSG with its ID.
II. CONFIRM PHASE: Each *red* node that during the first phase received only APPLY-MSG's of nodes with larger ID if any, colors itself *black* and sends a CONFIRM-MSG(black) with its ID and its parent's ID. Each *blue* node that receives a CONFIRM-MSG(black) with parent ID equal to its own ID, colors itself *grey*.
III. UPDATE PHASE: Each *red* or *white* node that during the second phase received one or more CONFIRM-MSG(black), colors itself *blue* and sends an UPDATE-MSG(blue) with its ID. Each *white* node that receives an UPDATE-MSG(blue) from node v colors itself *red* and sets its parent to be v.

Fig. 1. The distributed CDS algorithm.

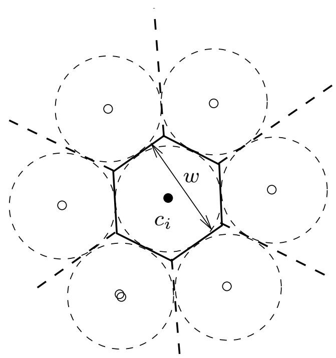

Fig. 3. The smallest Voronoi cell for point c_i

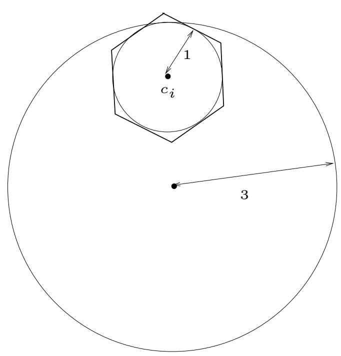

Fig. 4. How much of a Voronoi cell remains inside

We note that this bound immediately implies that the size of any independent set is bounded by $3.748 \cdot |\mathrm{OPT}| + 9$ since it consists of disjoint disks, which are contained in the area covered by L. Next we will derive an even better bound by making use of the fact that any placement of a unit-disk necessarily 'wastes' some area besides the area π covered by the unit-disk. Let us consider the Voronoi diagram of the centers c_i of the disks in L. How small can the Voronoi cell of a point c_i be ? It is not hard to see that since all c_i's have pairwise distance of at least 2, the smallest possible Voronoi cell is a regular hexagon of width $w = 2$. This follows immediately from the well-known result by Fejes Tóth [5] which proves that the densest packing of unit-disks in the plane is attained by a hexagonal lattice. See Figure 3, left. The area of one hexagon is $\frac{\sqrt{3}}{2} w^2 = 2\sqrt{3}$.

Theorem 3.2: The size of any independent set in a unit disk graph G is at most $3.453 \cdot |\mathrm{OPT}| + 8.291$.

Proof: First observe that any point in the Voronoi Cell of c_i is either covered by the unit-disk around c_i or not covered at all (if it was not covered by the unit disk centered at c_i but by another unit disk, it would not be in the Voronoi cell of c_i). So basically each placed unit-disk 'uses' up an area of at least $2\sqrt{3}$ (and not only π) from the area of the region covered by L with the only exception of disks near the boundary. If the center c_i of such a disk is close to the boundary, part of its Voronoi cell might lie outside the region covered by L. So we need to give a lower bound on the area z of the intersection of the Voronoi cell of c_i with the region covered by L. For that, let us allow other points c_j to be to be placed arbitrarily, in particular also outside the region covered by L (this will give only a smaller lower bound). The area z is then again minimized when there are 6 centers c_j placed regularly at distance 2 around c_i. How much of c_i's Voronoi cell can then lie outside the region covered by L ? As it is illustrated on Figure 4, at most $(2\sqrt{3} - \pi)/6$, that is one 'ear' of the regular hexagon (remember that the unit disk around c_i is contained in a union of disks of radius 3). Hence, we can uniquely assign each c_i an area of

$$2\sqrt{3} - (2\sqrt{3} - \pi)/6 \geq 3.410$$

from the area covered by L. Therefore, the number of disjoint unit disks that can be placed in the region covered by L is at most

$$\frac{|\mathrm{OPT}| \cdot 11.774 + 9\pi}{3.410} \leq 3.453 \cdot |\mathrm{OPT}| + 8.291.$$

■

We note that the same technique can be used to improve the number of non-adjacent D2/D3-Neighbors to 22 and 44 respectively improving upon the previously best bounds of 23 and 47. This has a direct impact on the size of several CDS constructions, which also have bounded geometric and topological dilation, like in [2], [6].

IV. Concluding Remarks

In this work we have proposed an improved distributed 6.91-approximation algorithm for computing a connected dominating set in unit disk graphs. The algorithm is very simple and can be easily implemented in wireless ad hoc networks. As main contribution of this paper, we have shown an improved analysis of the relationship between the size of a maximal independent set and a minimum CDS in a unit disk graph, which yields better bounds for many other algorithms.

REFERENCES

[1] K.M. Alzoubi, ”Connected Dominating Set and its Induced Position-less Sparse Spanner For Mobile Ad Hoc Networks,” *Proc. of 8th IEEE Int. Symp. on Computers and Communication (ISCC’03)*

[2] K.M. Alzoubi, Peng-Jun Wan, Ophir Frieder, ”Weakly-Connected Dominating Sets and Sparse Spanners in Wireless Ad Hoc Networks,” *Proc. of 23rd Int.Conf. on Distributed Computing Systems (IEEE, ICDCS), 2003*

[3] X. Cheng, X. Huang, D. Li and D.-Z. Du, “Polynomial-time approximation scheme for minimum connected dominating set in ad hoc wireless networks,” *Networks*, to appear.

[4] B. Clark, C. Colbourn and D. Johnson, “Unit Disk Graphs,” *Discrete Mathematics*, Vol. 86, pp. 165-177, 1990.

[5] L. Fejes Tóth, ”Über die dichteste Kugellagerung”, *Math.Zeit. 48 (1942 1943), 676–684*

[6] R. Gandhi and S. Parthasarathy, ”Fast Distributed Well Connected Dominating Sets for Ad Hoc Networks,” *CS-TR-4559*, UM Computer Science Department.

[7] M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi and D. J. Rosenkrantz, ”Simple Heuristics for Unit Disk Graphs,” *Networks*, Vol. 25, pp. 59-68, 1995.

[8] P. J. Wan, K. Alzoubi and O. Frieder, “Distributed Construction of Connected Dominating Set in Wireless ad hoc networks,” *Proc. of INFOCOM 2002*.