Theoretical Computer Science 607 (2015) 363–380

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

A better approximation for constructing virtual backbone in 3D wireless ad-hoc networks

Xiaofeng Gao *, Jun Li, Guihai Chen

Shanghai Key Laboratory of Scalable Computing and Systems, Department of Computer Science and Engineering, Shanghai Jiao Tong University, 200240, Shanghai, PR China

ARTICLE INFO

Article history:
Received 31 March 2015
Received in revised form 22 July 2015
Accepted 31 July 2015
Available online 7 August 2015

Keywords:
Wireless ad-hoc network
Virtual backbone
Connected dominating set (CDS)
Unit ball graph

ABSTRACT

Wireless ad hoc networks have been widely used in many areas. In order to improve network performance, we often select a connected dominating set (CDS) as its virtual backbone to deal with routing-related tasks. The problem of finding a minimum CDS (MCDS) for 2-dimensional networks has been widely studied, whereas finding an MCDS in 3-dimensional networks draws more attention recently, because it can formulate the network environment more precisely. Since MCDS problem is proved to be NP-complete, lots of approximations were proposed in literature. Among those, the best approximation for MCDS in 3D network is 14.937 in [1]. However, their projection method during the approximation deduction process is incorrect, which overthrows its final bound completely. As a consequence, in this paper we will first propose a new projection method to overcome their problem, illustrate the cardinality upper bound of independent points in a graph (which will be used to analyze the approximation ratio), and then optimize the algorithms to select MCDS with prune techniques. The major technique we use is an adaptive jitter scheme, which solves the open question in this area.

1. Introduction

Wireless ad hoc networks consist of lots of wireless nodes, which serve not only as mobile hosts but also as routers. Because of such characteristics, wireless ad hoc networks can be widely used in lots of applications such as sensor monitoring, traffic control, mobile computing, etc. [2–5]. However, those networks do not have physical infrastructures. When any two nodes in a wireless ad hoc network want to communicate with each other, they must forward messages to intermediate nodes to construct route between them. Consequently, this will cause unnecessary energy consumption and even broadcast storm when routing.

To overcome such shortcomings, we usually construct a virtual backbone to response for routing related tasks. A virtual backbone consists of a subset of all nodes in a wireless ad hoc network. Every node in the wireless ad hoc network is either in this subset or adjacent to at least one node of the subset. With virtual backbone, each ordinary node only need to send messages to its surrounding nodes which are in virtual backbone. And the nodes in virtual backbone will help it to forward such messages to its destination. As a result, virtual backbone can greatly reduce forwarding processes in wireless ad hoc networks. In literature, virtual backbone has been widely studied [6–9].

* Corresponding author.
E-mail addresses: gao-xf@cs.sjtu.edu.cn (X. Gao), lijun2009@sjtu.edu.cn (J. Li), gchen@cs.sjtu.edu.cn (G. Chen).

http://dx.doi.org/10.1016/j.tcs.2015.07.061

For any given wireless ad hoc network, we can represent it by an undirected graph $G = (V, E)$ as follows: any vertex $v \in V$ corresponds to a node in the original network, while any edge $(u, v) \in E$ represents that the nodes corresponding to vertices u and v can communicate with each other. Moreover, it is widely accepted that a connected dominating set (CDS) of the given graph is often the first choice to construct a virtual backbone for the corresponding wireless ad hoc network [10,11]. A CDS is defined to be a subset of V in a given graph $G = (V, E)$, such that every vertex of V is either in this subset or adjacent to a vertex in this subset and this subset can induce a connected subgraph.

Most literature discussed CDS in two-dimensional space, and use a unit disk graph (UDG) to model the network. However, such model cannot precisely describe the non-flat area such as mountainous region [12] or underwater environment [13]. Correspondingly, we can use a unit ball graph (UBG) to model such a network in 3-dimensional space. In a UBG $G = (V, E)$, any two vertices are adjacent (or connected) if and only if the Euclidean distance between them is at most 1.

Since UBG can formulate a network environment more precisely than UDG, CDS in UBG can represent more applications than that in UDG. For instance, Wang et al. [14] constructed 3D landmark maps with vision data extracted from camera images, and then used 3D-CDS to improve data association in application of simultaneous localization and mapping (SLAM). Yang [15] implemented 3D-CDS as clusters to find an optimal topology control strategy in 3D wireless sensor networks. In all, it is significant to design fast algorithms for selecting an appropriate CDS set from a given network and analyze their performance. Typically, a CDS with minimum cardinality is the most efficient choice for practical use, and we refer it as MCDS.

Finding a minimum CDS (MCDS) is a well-known NP-complete problem, and lots of approximation algorithms were proposed during last decade. Those algorithms often include two phases. Firstly, they choose a maximal independent set (MIS) from G. Second, they add some extra nodes from G to connect this MIS, usually by Steiner trees. An MIS in a graph $G = (V, E)$ is a subset $M \subseteq V$ such that any two vertices from M are not connected and we cannot add another vertex from $M \backslash V$ to form a bigger MIS. Easy to see, in UDG or UBG, the distance between any two vertices in M should be more than 1.

In 2-dimensional situation, the approximation ratio of such algorithms has been widely studied. Based on the fact that the neighborhood area of any node can contain at most five independent points, Wan et al. [16] proposed that $mis(G) \leq 4mcds(G) + 1$. Later, Wu et al. [17] improved this ratio to 3.8 by proving that the neighborhood of any two adjacent nodes can contain at most 8 nodes. In [18], Gao et al. showed the bound can be at most 3.453 and Li et al. improved the ratio into 3.4305 in [19]. Recently, Du et al. [20] showed that $mis(G) \leq 3.399mcds(G) + 4.874$, which is the best result up to now.

To analyze the performance of those approximations, we need to decompose the algorithm selection and compare each part separately to an optimal solution. If $alg(G)$ is the size of selected MCDS by those algorithms, then the approximation ratio of these algorithms can be calculated by

$$\frac{alg(G)}{mcds(G)} = \frac{mis(G)}{mcds(G)} + \frac{connector(G)}{mcds(G)},$$

where $mis(G)$ is the size of MIS the algorithm selected, $connector(G)$ is the number of nodes used to connect such MIS, and $mcds(G)$ is the size of an optimal MCDS. Generally, $connector(G)$ highly depends on the value of $mis(G)$. Thus, the ratio $mis(G)/mcds(G)$ plays an important role when analyzing the performance of those approximations.

An interesting geometric property between an MIS and a MCDS will be helpful for us to calculate $mis(G)/mcds(G)$. For example, in UDG, if we shrink the radius of a disk from 1 to 0.5 for any vertices in the graph (denoted as small disks), and enlarge the radius of a disk from 1 to 1.5 for any vertices in an optimal MCDS (denoted as large disks), then any small disks will locate in the region formed by the union of MCDS large disks, and any two disks formed by two independent vertices will not intersect with each other. Fig. 1 shows such a scenario, where the red points denote a vertex in MCDS and blue points denote independent vertices. Hence, we can apply disk packing, namely how many disks with radius 0.5 the dominating areas of an MCDS can contain, to estimate the ratio of $mis(G)/mcds(G)$.

Although finding a minimum CDS in UBG is very similar as in UDG, the approximation analysis for UBG is much harder since the geometric properties in UBG are more complicated to analyze than in UDG. In UDG, the problem of "how many disks of the same size can a disk touch" is a foundation for disk packing [18–20]. Easy to see, when a disk touch two disks of the same size, the angle of those two touched disks to this touching disk is at least $\pi/3$. Therefore, we can easily figure out that a disk can touch $\frac{2\pi}{\pi/3} = 6$ disks of the same size. However, when it comes to three-dimension, the corresponding discussion is not that easy. In UBG, we can use the same ideas in UDG and replace those disks with corresponding spheres. Then, we can use sphere packing to estimate the ratio of $mis(G)/mcds(G)$ in UBG. Similarly, we should first solve the corresponding extended problem, "how many spheres of the same size can a sphere touch", which is well known as Gregory–Newton problem. However, this problem is so difficult that it has been a puzzle in literature until 1953 [21].

To the best of our knowledge, few papers studied the approximation ratio for MCDS problem in UBG. In the earlier stage, Hansen et al. [22] discussed the expected size of a CDS in a random UBG and compared the performance of existing algorithms. Later, Butenko and Ursulenko [23] proved that the ratio of $mis(G)/mcds(G)$ in UBG is at most 11 by using the well-known fact that a sphere can touch at most twelve spheres of the same size, which induced an approximation ratio of 22 for MCDS in UBG. Zhong et al. [24] claimed that such ratio could be reduced to 16. However, Kim et al. [1] pointed out and proved that both the algorithm and approximation analysis in [24] have problems. Zou et al. [25] further reduced this ratio to $13 + \ln 10$. Recently, Kim et al. [1] referred the idea in [17] and tried to answer how many independent points can be contained in two adjacent unit balls. Finally, they improved the ratio of $mis(G)/mcds(G)$ into 10.917 by showing