

Theoretically Good Distributed CDMA/OVSF Code Assignment for Wireless Ad Hoc Networks

Xiang-Yang Li^{1,*} and Peng-Jun Wan^{1,2,*}

¹ Illinois Institute of Technology, Chicago, IL, USA
{xli,wan}@cs.iit.edu

² Hong Kong City University, Hong Kong, China

Abstract. We present several distributed CDMA/OVSF code assignment algorithms for wireless ad hoc networks modelled by unit disk graph (UDG). We first give a distributed code assignment whose total throughput is within a constant factor of the optimum. Then we give a distributed method such that the minimum rate achieved is within a constant factor of the optimum. A distributed method that can approximate both the minimum rate and total throughput is also presented. All our methods use only $O(n)$ total messages (each with $O(\log n)$ bits) for an ad hoc wireless network of n nodes modelled by UDG.

Keywords: CDMA code assignment, coloring, throughput, bottleneck, interference, wireless networks.

1 Introduction

We consider a static wireless ad hoc network consisting of a set V of n nodes distributed in a two-dimensional plane. Assume all nodes have the same transmission radius r , thus, wireless ad hoc networks are modelled by unit disk graphs (UDG), in which two nodes are connected iff their Euclidean distance is no more than r . We assume that the omnidirectional antenna is used by all wireless nodes: the signal sent by a node will be received by all nodes inside its transmission region. The transmission region of a node u is thus modelled as a disk $D(u, r)$ centered at u with radius r . To increase the capacity of the network, frequency spectrum has to be reused as it is one of the scarcest resources available. Same channel is not assigned to two nodes if it causes either *primary interference* or *secondary interference*. Primary interference occurs if two nodes use the same channel and one is inside the transmission region of the other. The secondary interference occurs if a third node is within the common transmission regions of two nodes using the same frequency channel. The *interference graph* $G = (V, E)$ has an edge uv if two nodes u and v will generate interference when they are assigned the same channel. Assigning frequency channel efficiently in UDG has been well-studied [6, 10] but little is known about assigning CDMA/OVSF code for wireless ad hoc networks while achieving some global quality such as maximizing the total throughput or the bottleneck of the networks.

* The research was supported in part by NSF under Grant CCR-0311174.

In a CDMA system, the channels are defined by the pseudo-random codewords. For simplification, we represent each CDMA/OVSF codeword by a binary string (called colors hereafter). Two colors are said to be *prefix-free* if neither is a prefix of the other, which is equivalent to that the corresponding codewords are orthogonal. The rate of an ℓ -bit color is equal to $2^{-\ell+1}$, which is equal to the rate of the corresponding codeword. We also say that an ℓ -bit color is in the ℓ -th layer of the CDMA/OVSF code tree structure. The root has layer 1.

A (proper) vertex coloring is to assign each vertex a color such that two adjacent vertices receive different colors. The CDMA code assignment is to assign colors to nodes such that adjacent nodes in the interference graph receive prefix-free colors, which is called *prefix-free* vertex coloring. The minimum vertex coloring of the interference graph has been studied in the context of channel assignment in wireless ad hoc networks channelized by FDMA, TDMA or CDMA/OVSF [2–5, 7, 11, 12, 14, 15]. The majority of these CDMA code assignment methods simply presented networking protocols to obtain a proper vertex coloring without addressing the computational complexity and/or the optimization. Sen and Huson [13] gave a proof of the NP-hardness of the vertex coloring in interference graph even when all nodes are located in a plane and have the same transmission radii. A problem related to the (prefix-free) vertex coloring of the interference graphs is the *distance-2 vertex coloring* [8]. A *distance-2 vertex coloring* of a graph H is a proper vertex coloring of H^2 , the *square graph* of H , which is the graph obtained by creating an edge between each pair of vertices of H separated by at most two hops in H . Notice that the colors assigned to two adjacent nodes in H^2 should only be different for a vertex coloring problem, while these two colors should further be prefix-free for CDMA/OVSF code assignment.

Given a prefix-free CDMA code assignment $\{c_v \mid v \in V\}$, its *throughput* and *bottleneck* are defined as $\sum_{v \in V} 2^{-|c_v|+1}$ and $\min_{v \in V} 2^{-|c_v|+1}$ respectively, where $|c_v|$ denotes the number of bits of the color c_v . The *throughput of an interference graph* G , denoted by $\tau(G)$, is then the maximum of the throughput over all prefix-free code assignments of G . Similarly, the *bottleneck of an interference graph* G , denoted by $\beta(G)$, is then the maximum of the bottleneck over all prefix-free code assignments of G . We will study various optimization problems on prefix-free vertex coloring of the interference graphs: maximize the total throughput, the minimum rate, and both at the same time.

The main contributions of this paper are as follows. We propose several efficient distributed CDMA/OVSF code assignment algorithms for wireless ad hoc networks modelled by UDG. We first study how to assign CDMA/OVSF code such that the total throughput achieved is within a constant factor of the optimum. Then we give a method such that the minimum rate achieved is within a constant factor of the optimum. A method that can approximate both the minimum rate and total throughput simultaneously is also presented. All our methods use only $O(n)$ total messages (each with $O(\log n)$ bits) for an ad hoc wireless network of n nodes modelled by UDG.

2 Distributed Code Assignment

Let $N_k(u)$ be the set of all wireless nodes that are at most k hops away from node u in UDG, $d_k(u)$ be the cardinality of $N_k(u)$. Obviously, nodes that can have primary interference with u are $N_1(u)$ only; nodes that can have either primary interference or secondary interference with u are $N_2(u)$ only. If every node knows its exact geometry location, a communication efficient protocol [1] is known to find all two-hop neighbors of all nodes using at most $O(n)$ communications.

2.1 Maximize Throughput $\tau(G)$

First-fit coloring is a class of greedy algorithms for vertex coloring. Assume that there is a (partial) ordering of all nodes. We then assign code to nodes sequentially according to the associated ordering by assigning each node the shortest possible code. Thus, in any first-fit coloring, all nodes receiving the same code form a *maximal* independent set (MIS). Intuitively, such MIS should be a small constant approximation of a *maximum* independent set to maximize the throughput. Clearly, the performance of a first-fit code assignment depends on the ordering used. Indeed, there always exists an ordering in which the first-fit coloring generates an optimal code assignment. However, such ordering is unlikely to be found in polynomial time due to the expected NP-hardness of the max-throughput code assignment. So we seek some node ordering that produces a code assignment approximating $\tau(G)$; such node ordering should be generated efficiently. We propose several different node orderings and show that all of them produce a code assignment with total throughput $O(\tau(G))$ and use total communications $O(n)$. Hereafter, we assume that each message has $O(\log n)$ bits. All node orderings used here are just partial ordering computed locally.

We will first construct the interference graph and then construct an MIS based on a rank (*e.g.*, ID, or degree, or the node's geometry position). Nodes in the MIS are assigned the shortest code 10. For the remaining nodes, we assign code using the first fit heuristics based on a partial ordering (*e.g.*, ID, or degree, or the node's geometry position). Here we assume that every node has a distinctive ID and knows its geometry position if a communication efficient protocol is needed. Algorithm 1 presents our method (run by every node u) of assigning CDMA/OVSF code based on ordering by ID to maximize the throughput.

We find an MIS in a distributed manner as follows: initially all nodes' status are *White*; a node becomes *InIS* if it has a *rank* smaller than all its neighbors with status *White*; a node becomes *NotInIS* if it has an *InIS* neighbor. A node could be either *White*, or *InIS*, or *NotInIS*. The nodes with status *InIS* form an MIS. We will show that its size is within a constant factor of the maximum independent set for an interference graph G . See [9, 17] for more details.

Obviously Algorithm 1 generates a prefix-free code assignment since, for each pair of neighboring nodes u and v in the interference graph, the node with larger ID can only assign code after it gets the code of the other node. The total communication cost is $O(n)$ since we use communication efficient protocol to

Algorithm 1 Max-Throughput Using ID-ordering by u

-
- 1: Node u sends a message to tell its ID to all nodes $N_1(u)$. If secondary interference is not permitted, node u finds $N_2(u)$ using a communication efficient method [1].
 - 2: All nodes collectively find an MIS based on a rank ID.
 - 3: Node u assigns a CDMA code represented by binary 10 if u is in the MIS. It then informs its neighbors in G about its code.
 - 4: If node u receives a CDMA code from its neighbor in G , u marks the corresponding code *used* in the CDMA/OVSF tree structure stored locally.
 - 5: Then assign code to the remaining nodes. If node u has an ID smaller than all its neighbors in G without a code, then node u finds the smallest layer $h > 0$ in the local CDMA/OVSF code tree such that layer h has at least 2 free codes not marked used. Node u picks the first unused code in layer h and informs its neighbors in G about its code. The picked code is called the *first fit* code for node u .
-

collect $N_2(u)$ for all nodes and to inform the assigned code to its neighbors in G . We can also use the node degree (or position) to find an MIS in Algorithm 1.

We then show that the above methods indeed approximate the optimum throughput $\tau(G)$. To do so, we first study the structure of some optimum CDMA code assignment, called *canonical coloring*. In [16], we defined the *canonical coloring* as follows. Given a graph $G = (V, E)$, partition the vertex set V into independent sets V_1, V_2, \dots, V_k with $|V_1| \geq |V_2| \geq \dots \geq |V_k|$. Let $G_0 = G$ and G_i be the graph of removing the vertices V_i and the incident edges from graph G_{i-1} , for $1 \leq i \leq k$. Vertex set V_i is a maximum independent set of graph G_{i-1} . For $1 \leq i \leq k-1$, all nodes in V_i receive the code $1^i 0$, and all nodes in V_k receive the code 1^k . Obviously, the throughput of such canonical coloring is $\sum_{i=1}^{k-1} \frac{|V_i|}{2^i} + \frac{|V_k|}{2^{k-1}}$. Notice that, If there are multiple maximum independent sets V_1 , we have to choose the one that produces the largest maximum independent set V_2 . Similarly, the selection of the first i maximum independent sets V_1, V_2, \dots, V_i produces the largest maximum independent set V_{i+1} , for $1 \leq i < k$. Call such sequence of maximum independent set as *canonical maximum independent set decomposition* and the corresponding coloring *canonical coloring*.

Theorem 1. [16] *The canonical coloring maximizes the throughput.*

This theorem implies that the maximum throughput of any code assignment is at most the independence number $\alpha(G)$ of the interference graph G . Based on this observation, we can assign the code as follows. First, compute an MIS that approximates the maximum independent set (with approximation ratio ρ). Then assign the nodes in the MIS a code 10 (its rate is $1/2$). For the remaining nodes, we can recursively find the MIS and assign code $1^i 0$ for the MIS retrieved in the i th iteration but the messages of this approach could be very large. To optimize the message complexity, Algorithms 1 used a different approach for the remaining nodes (actually any prefix-free code assignment for the remaining nodes works here). Obviously, the throughput generated by assigning nodes in MIS a code 10 is at least $\rho \cdot \alpha(G)/2$. This implies the following theorem (see appendix for the proof).

Theorem 2. *Algorithm 1 generates a code assignment whose throughput is at least $\varrho/2$, where $\varrho = 1/5$ if only primary interference is concerned and $\varrho = 1/13$ if secondary interference is also concerned.*

When every node knows its position, we can further improve the theoretical lower bounds on the throughput of the assigned codes as follows. We still construct an MIS first, but instead of using the node ID or the degree as selection criterion, we select a node u to the MIS if all unassigned neighboring nodes are inside one half of the disk centered at u . Notice that such node u always exists since the most left undecided node trivially satisfies this condition.

Algorithm 2 Max-Throughput Using Position-ordering by u

- 1: Every node finds its neighbors in G using a communication efficient protocol in [1].
 - 2: All nodes together compute an MIS based on rank $(x(u), y(u), ID(u))$, where $x(u)$, and $y(u)$ are the x -coordinate and y -coordinate of u .
 - 3: Node u gets code 10 if it is in the computed MIS.
 - 4: All nodes not in MIS get the *first fit* code in an *increasing* ordering of $(x(u), y(u), ID(u))$ using method similar to the last step of Algorithm 1.
-

Theorem 3. *Algorithm 2 generates a code assignment whose throughput is at least $\varrho/2$, where $\varrho = 1/3$ ($\varrho = 1/7$ resp.) if primary interference (secondary interference resp.) is concerned. It uses $O(n)$ messages, each with $O(\log n)$ bits.*

The proof is similar to Theorem 2 and is omitted. The approximation ratio could be further improved to be better than $\varrho/2$, which is analyzed as follows. The new approach will compute an MIS V'_1 , and then compute an MIS V'_2 for the remaining nodes. Clearly, the number of messages is still $O(n)$. The nodes in V'_1 will receive a code 10 and the nodes in V'_2 will receive a code 110. We assign codes to other nodes using a method similar to the last step of Algorithm 1.

Theorem 4. *An ϱ -approximation algorithm for the maximum independent set gives a $\frac{5}{8}\varrho$ -approximation algorithm for the maximum throughput code assignment.*

PROOF. Consider a canonical maximum independent decomposition V_1, V_2, \dots, V_k of all nodes V . Here $|V'_1| \geq \varrho \cdot |V_1|$. Let $t_{i,j} = \frac{|V'_i \cap V_j|}{|V_j|}$, i.e., the portion of V_j is used in V'_i . After V'_1 is generated, we know that the maximum independent set in the remaining graph has size at least $\max((1 - t_{1,1}) \cdot |V_1|, (1 - t_{1,2}) \cdot |V_2|)$, since $V_1 - V'_1 \cap V_1$ and $V_2 - V'_1 \cap V_2$ are still independent sets. Notice that $t_{1,1} \cdot |V_1| + t_{1,2} \cdot |V_2| \leq |V_1|$. Then $(1 - t_{1,1}) \cdot |V_1| + (1 - t_{1,2}) \cdot |V_2| \geq |V_2|$. It implies that V'_2 has size at least $\varrho \cdot |V_2|/2$. Consequently, the throughput τ' generated by partition $V'_1, V'_2, \dots, V'_k, \dots, V'_{k_2}$ is at least $\varrho \cdot (\frac{|V_1|}{2} + \frac{|V_2|}{2 \cdot 2^2})$. Remember that the canonical coloring has throughput τ at most $\frac{|V_1|}{2} + 2 \cdot \frac{|V_2|}{2^2}$ using fact $|V_i| \leq |V_2|$. From $|V_2| \leq |V_1|$, it is easy to show that $\tau' \geq \frac{3}{8}\varrho \cdot \tau$. This finishes the proof. \square

Theorem 5. *If node position is known, we can produce a code assignment, using $O(n)$ total messages, whose total throughput is at least $5/24$ (resp. $5/56$) of the optimum when primary interference (reps. secondary interference) is concerned.*

2.2 Maximize Bottleneck $\beta(G)$

We continue to study how to assign codes to maximize the minimum rate. Intuitively, to maximize the throughput, from the canonical code assignment discussion, the assigned codes should be imbalanced. However, to maximize the minimum rate, the assigned codes should be as balanced as possible. Clearly, the previous greedy methods do not generate a balanced code assignment. In this section, we present a novel distributed method to assign a balanced code.

Our method is based on the following observation. Consider a node u and all its neighbors in the interference graph G . If all such neighbors and u form a clique, then the minimum rate of these nodes is approximately $1/d$, where d is the size of the clique. This is achieved when all nodes use the code in level $\log d$. In other words, to maximize the minimum rate assigned, node u cannot choose the first fit code; it has to use a code in level close to $\log d$. Putting in other way, node u cannot be too greedy and it has to leave good codes for its neighbors. The following Algorithm 3 details our method.

Algorithm 3 Max-Bottleneck by Degree-ordering by a node u

- 1: All nodes together compute the interference graph G . Assume that each node u knows its degree $d(u)$ in G . Each node u informs its neighbors in G its degree $d(u)$.
 - 2: Node u constructs a local binary code tree T .
 - 3: If node u has the *largest* degree $d(u)$ among all neighbors in G without a code, where ties are broken by smaller ID, node u picks the first unmarked code in the code tree T stored locally from layer ℓ , where $2^{\ell-2} < d(u) + 1 \leq 2^{\ell-1}$. Here a code is marked if it is either marked as *used* or *conflicted*.
Node u informs its neighbors in G the selected code of u efficiently.
 - 4: If a node u receives a code message from its neighbor in G , u marks the corresponding code *used* in T , and marks all prefix-codes of this code *conflicted* in T .
-

Theorem 6. *Algorithm 3 generates a prefix-free code assignment whose minimum rate is within a constant factor of optimum.*

PROOF. It is easy to show that it generates a prefix-free code assignment (thus the proof is omitted due to space limit). Consider a node u with the largest degree $d(u)$ in G . If primary interference is concerned, we partition the disk $D(u, 1)$ into 6 equal-sized sectors. If secondary interference is also concerned, we partition the disk $D(u, 2)$ into 13 equal-sized sectors. We already showed that all neighbors of u inside one sector form a complete subgraph in G . Using the pigeonhole principle, it is easy to show that among the neighbors of u in G and u , the minimum clique size is at least $c \cdot d(u) + 1$, where $c = 1/6$ for primary interference graph, and $c = 1/13$ for secondary interference graph. For a clique

of size q , the minimum rate of nodes in the clique is obviously at most $2^{-\lceil \log_2 q \rceil}$. Thus, for any assignment, the minimum rate among neighbors of u and node u is at most $2^{-\lceil \log_2 (c \cdot d(u)+1) \rceil}$. Obviously, the rate by our approach is $2^{-\lceil \log_2 (d(u)+1) \rceil}$. Then $2^{-\lceil \log_2 (d(u)+1) \rceil} \geq 2^{-\lceil \log_2 c \rceil} \cdot 2^{-\lceil \log_2 (c \cdot d(u)+1) \rceil}$ implies that the minimum rate achieved by Algorithm 3 is at least $1/8$ ($1/16$ resp.) of the optimum if the primary interference (the secondary interference resp.) is concerned. \square

2.3 Maximize $\tau(G)$ and $\beta(G)$

As we discussed before, to maximize the throughput, the assigned codes should be as imbalanced as possible, while to maximize the bottleneck rate, the assigned codes should be as balanced as possible. It seems impossible to have a code assignment that approximates both the total throughput and the bottleneck rate. In this subsection, we show that by retreating little bit on both requirements, we can achieve this. Our method is almost a straightforward combination of previous methods. We first assign the shortest code to the nodes in an MIS. For the remaining nodes, we assign a balanced code.

Algorithm 4 Max-Throughput and Bottleneck by a node u

- 1: All nodes together compute the interference graph G . Each node u computes its degree $d(u)$ in G and informs its neighbors in G about its degree $d(u)$.
 - 2: All nodes together compute an MIS based on the rank by degree. Node ID or $(x(u), y(u), ID(u))$ can also be used as the rank criterion. Node u gets CDMA/OVSF code 10 if it is in the computed MIS. The remaining steps will assign code for other nodes.
 - 3: Each node u constructs a binary code tree T .
 - 4: If node u is not assigned and has the largest degree $d(u)$ among all its neighbors in G without a CDMA code, node u picks the first unmarked code from layer ℓ in T , where $2^{\ell-3} < d(u) \leq 2^{\ell-2}$. Node u informs all its neighbors in G the selected code.
 - 5: If node u receives a message from its neighbor v informing the code of v , u marks this code *used*, and marks all prefix-codes of this code *conflicted* in tree T .
-

Theorem 7. *Algorithm 4 generates a prefix-free code assignment whose total throughput is within $\varrho/2$ of the optimum, and whose minimum rate is within $2^{-\lceil \log_2 c \rceil - 1}$ factor of the optimum, where ϱ is the approximation ratio of the maximum independent set algorithm, $c = 1/6$ for primary interference and $c = 1/13$ for secondary interference.*

3 Conclusion

We presented several efficient distributed CDMA/OVSF code assignment algorithms. Notice that our theoretical analysis is pessimistic. In [18], we presented methods to further improve the performance. We conducted extensive simulations and we found that the practical performances of our methods are much

better than these pessimistic analysis. Our methods can also be used to generate prefix-free code assignment for wireless ad hoc networks that are not modelled by UDGs. The UDG network model only enables us to prove that our methods have constant approximation ratios. However, it is unclear how to bound the communications in no-UDG model or without position information.

This paper is not intended to solve all critical issues in CDMA based wireless ad hoc networks. There are several other important issues that should be addressed, *e.g.*, the mobility of wireless nodes; and the time synchronization among the mobile wireless nodes. See [18] for more discussions.

References

1. CĂLINESCU, G. Computing 2-hop neighborhoods in ad hoc wireless networks, 2003. AdHocNow.
2. CHLAMTAC, I., AND FARAGO, A. Making transmission schedules immune to topology changes in multi-hop packet radio networks. *IEEE/ACM Transactions on Networking* 2, 1 (1994), 23–29.
3. CHLAMTAC, I., AND KUTTEN, S. A spatial reuse TDMA/FDMA for mobile multihop radio networks. In *IEEE INFOCOM* (1985), pp. 389–394.
4. GARCIA-LUNA-ACEVES, J. AND RAJU, J. Distributed assignment of codes for multihop packet-radio networks. *IEEE MILCOM*, volume 1, pages 450–454, 1997.
5. GOLDBERG, A., AND RAO, S. Flows in undirected unit capacity networks. Tech. Rep. 97-103, NEC Research Institute, Inc, 1997.
6. GRAF, A., STUMPF, M., AND WEISENFELS, G. On coloring unit disk graphs. *Algorithmica* 20, 3 (1998), 277–293.
7. HU, L. Distributed code assignments for CDMA packet radio networks. *IEEE/ACM Transactions on Networking*, 1:668, Dec. 1993.
8. KRUMKE, S., MARATHE, M., AND RAVI, S. Models and approximation algorithms for channel assignment in radio networks. *Wireless Networks* 7 (2001), 567–574.
9. LI, X.-Y., AND WANG, Y. Simple heuristics and PTASs for intersection graphs in wireless ad hoc networks. In *ACM DialM* (workshop of ACM MobiCom) (2002).
10. MCDIARMID, C., AND REED, B. Colouring proximity graphs in the plane. *Discrete Mathematics* 199 (1999), 123–137.
11. NELSON, R., AND KLEINROCK, L. Spatial-TDMA: A collision-free multihop channel access protocol. *IEEE Transactions on Communications* 33, 9 (1985), 934–944.
12. RAMANATHAN, S., AND LLOYD, E. Scheduling algorithms for multi-hop radio networks. *IEEE/ACM Transactions on Networking* 1 (April 1993), 166–172.
13. SEN, A., AND HUSON, M. L. A new model for scheduling packet radio networks. *ACM/Baltzer Journal Wireless Networks* 3 (1997), 71–82.
14. SEN, A., AND MALESINSKA, E. Approximation algorithms for radio network scheduling. In *Allerton Conf. on Comm., Contr. and Comp.* (1997), pp. 573–582.
15. STEVENS, D., AND AMMAR, M. Evaluation of slot allocation strategies for TDMA protocols in packet radio networks. In *IEEE MILCOM* (1990), pp. 835–839.
16. WAN, P.-J., LI, X.-Y. AND FRIDER, O. OVSF-CDMA code assignment for wireless ad hoc networks. ACM DialM (workshop of ACM MobiCom), 2004.
17. WANG, Y., AND LI, X.-Y. Geometric spanners for wireless ad hoc networks. In *Proc. of 22nd IEEE ICDCS* (2002).
18. LI, X.-Y., WAN, P.-J. AND SONG, W.-Z. Theoretically Good Distributed CDMA/OVSF Code Assignment for Wireless Ad Hoc Networks. Tech Report, IIT, 2004, <http://www.cs.iit.edu/~xli/publications-select.htm>

Appendix (Proof of Theorem 2)

PROOF. Let's consider all nodes, denoted by V_1 , that receive code 10. Clearly, V_1 is independent. We will show that $|V_1|$ is within ϱ factor of $\alpha(G)$.

If only primary interference is concerned, G is the original UDG and it is well-known that the greedy method generates an MIS whose size is at least $1/5$ of the maximum independent set. Obviously, the total throughput generated by our approach is at least $|V_1|/2$ and the optimum throughput is at most $\alpha(G) \leq 5|V_1|$.

If the secondary interference is concerned, we will prove that $|V_1|$ has size at least $1/13$ of $\alpha(G)$ by showing that, $\forall u \in V_1$, there are at most 13 independent nodes in G . Let $D(x, r)$ be the disk centered at a point x with radius r hereafter. Consider a disk $D(u, 2)$ centered at node u with radius 2. Then all its neighbors $N_2(u)$ are inside the disk $D(u, 2)$. Partition this disk into 13 equal-sized sectors, each with angle $2\pi/13$. It is easy to show that the chord ab defined by the sector $\angle aub$ has length $4 \sin(\pi/13) < 1$. We will show that all neighboring nodes in one sector are connected. Consider any two nodes x and y from $N_2(u)$. We actually will prove a stronger result: any two neighbors of u in the sector $\angle aub$ with $\|ab\| = 1$ are connected in the interference graph.

If x and y are inside $D(u, 1)$, then obviously $\|xy\| < 1$. Thus, x and y are connected in G . If y is inside $D(u, 1)$ but x is not, then there exists a node w connected to both x and u . Clearly, y and w are all inside $D(u, 1)$ now, thus, edge yw exists in the original unit disk graph. Thus, node w is inside the common transmission range of nodes y and x . It implies that x, y are connected in G (concerning the secondary interference).

Finally, we consider the case when both x and $y \notin D(u, 1)$. Assume that node w is connected to both x and u , and node v is connected to both y and u . See Figure 1 (a) and (d) for an illustration. We will then show that either $\|yx\| \leq 1$, or $\|yw\| \leq 1$, or $\|vx\| \leq 1$. Notice that, if any one is true, then x, y are connected in G . For the sake of contradiction, assume that $\|yx\| > 1$, $\|yw\| > 1$, and $\|vx\| > 1$. We partition the region $\angle aub - \angle cud$ into 6 regions. Figure 1 (b) illustrates such six partitions. Here segments ca, db, ab, eb, am have length 1.

We then prove that any two nodes in region $efa \cup mfb$ have distance at most 1 and any two nodes in region $efbha$ have distance at most 1. Consider any two nodes x and y in the region $efa \cup mfb$. If both are in the same triangle, then clearly $\|xy\| < 1$ since the triangles have side-length less than 1. Otherwise, let x' and y' be the intersection point of line xy with segment ea and segment mb respectively. Figure 1 (b) and (e) illustrate the proof that follows. Obviously, $\|xy\| \leq \|x'y'\| \leq \min(\|ey'\|, \|ay'\|)$. Note that $\|ey'\| \leq \min(\|em\|, \|eb\|) < 1$ and similarly $\|ay'\| \leq \min(\|am\|, \|ab\|) = 1$. Thus, $\|xy\| \leq 1$. Similar proof reveals that any two nodes in region $efbha$ have distance at most 1.

If node x is in region 2, then node y cannot be in region 3, 5, and 6 since we can show that otherwise $\|xy\| \leq 1$. In other words, node y must be in region 1 or 4 in this case. Similarly, if node x is in region 3, 5, or 6, node y must be in region 1 or 4 in this case. Thus, we assume that either node x or y (say x w.l.o.g) is in region 1 by symmetry. Obviously, node y cannot be inside the disk $D(x, 1)$ since we assume that $xy \notin G$. Thus, we have to place node v inside the sector

$\angle cud$ but not inside the disk $D(x, 1)$ and place y inside region $efmbha$ but not inside the disk $D(x, 1)$ while still maintain $\|yv\| \leq 1$. We then show that this is impossible. Figure 1 (c) and (f) illustrate the proof that follows.

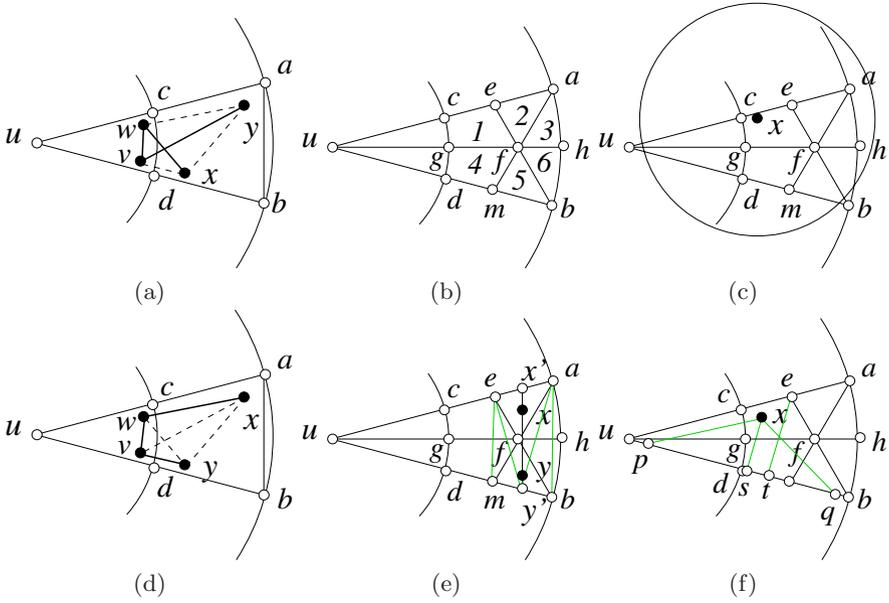


Fig. 1. All neighbors in the sector conflict with each other. Here $\|uc\| = \|ud\| = \|ab\| = 1$ and $\|ua\| = \|ub\| = 2$. (a): wx and vy intersect; (b) 6 regions to place node x or y ; (d): wx and vy don't intersect; (e): no two independent nodes in regions 2 and 5

If the disk $D(x, 1)$ contains the region $cgdbha = \angle aub - \angle cud$, then clearly node y is inside the disk $D(x, 1)$. It implies that xy is an edge in G . Let p and q be the points on line ub such that $\|xp\| = \|xq\| = 1$. Let s be the point on ub such that xs is perpendicular to segment pq and t be the point on ub such that et is perpendicular to segment pq . Clearly, $\|xs\| \leq \|et\|$ since x is inside the triangle $\triangle eub$. It is not difficult to show that $\|ce\| = \|ea\| = 1/2$. Then, $\|et\| = \|ue\| \cdot \sin(\angle aub) < \frac{3}{2} \sin(\frac{\pi}{6}) = 3/4 < \sqrt{3}/2$. It implies that $\angle xqp = \arcsin(\|xt\|/\|xq\|) < \frac{\pi}{3}$. Thus, edge pq is the longest in triangle $\triangle xpq$. Consequently, $\|pq\| > 1$. It is easy to show that, for any two-hop neighbor y of u connected through node v , $\|yv\| \geq \|pq\|$ if both v and y are not inside the disk $D(x, 1)$. This is a contradiction to $\|yv\| \leq 1$. This finishes the proof. \square