Beyond Interference Avoidance: Distributed Sub-network Scheduling in Wireless Networks with Local Views

Pedro E. Santacruz, Vaneet Aggarwal, and Ashutosh Sabharwal

***Abstract*—In most wireless networks, nodes have only limited local information about the network state, which includes connectivity and channel state information. With limited local information about the network, each node's knowledge is mismatched, therefore they must make distributed decisions. In this paper, we pose the following question - if every node has network state information only about a small neighborhood, how and when should nodes choose to transmit? While scheduling answers the above question for point-to-point physical layers which are designed for an interference-avoidance paradigm, we look for answers in cases when interference can be embraced by advanced code design, as suggested by results in network information theory. To make progress on this challenging problem, we propose a distributed algorithm which achieves rates higher than interference-avoidance based link scheduling, especially if each node knows more than one hop of network state information.**

I. Introduction

The shared nature of wireless communication networks results in the fundamental problem of dealing with interference from simultaneous transmissions by co-located flows. The most commonly used technique of managing interference is to avoid it by scheduling transmissions such that the co-located flows do not transmit simultaneously. Link scheduling inherently assumes that the underlying physical layer architecture is to decode a single packet. Link scheduling, both centralized and distributed, has a rich history and continues to be an active area of research, [1]-[5] (and references therein). In this paper, we pose and study the scheduling problem for the case when the physical layer architecture can embrace interference by using advanced coding methods.

While interference avoidance continues to be the near de-facto strategy in wireless networks, it has been known for some time that avoiding interference is not a capacity maximizing strategy for many networks. For example, techniques like multi-user detection, Han-Kobayashi coding for 2-user interference channel and interference alignment for general interference networks are known to yield higher capacity by embracing, not avoiding, interference; see the book-length exposition [6]. These new ideas have also inspired new standardization activity, like CoMP [7], which uses network MIMO to improve capacity at the edge of the cells. However, almost all such advanced techniques assume extensive knowledge about the network topology, channel statistics, and, in many cases, instantaneous channel information, to achieve capacity gains from embracing interference [6]. A direct impact of requiring such extensive knowledge at each node is that the resulting network architecture is not scalable.

In this paper, we pose the following problem: if each node in the network has limited information about the network state (connectivity and channel states), say it only knows the network state information within $h \geq 1$ hops away from it, then what is the capacity maximizing transmission scheme? Note that limited local information problems have been extensively studied in distributed scheduling [2]-[5]. However, as mentioned above, all of them assume interference avoidance as their underlying architecture. In this paper, the physical layer architecture is not restricted a priori and is allowed to be any feasible scheme like in network information theory analyses [6]. However, unlike network information theory formulations, we are explicitly studying only scalable architectures by limiting network state information at each node.

The new posed problem turns out to be extremely hard, and is the generalized version of the distributed capacity problem studied recently in [8], [9]. The formulation in [8], [9] shared full network connectivity information with all nodes but only h-hop information about the channel state. The key result in [8] was that a generalized form of scheduling is information-theoretically optimal for many networks. The general scheduling, labeled Maximum Independent Graph (MIG) Scheduling, schedules *connected sub-networks* larger than a single link. This is especially true if $h > 1$, i.e., nodes know more than one hop of channel information.

In this paper, no global connectivity information is available at any node and, hence, all decisions must be truly distributed. To make progress on the challenging new network capacity problem, we use MIG Scheduling as our starting point and focus on how sub-networks can be *identified*, *selected* and *scheduled* in a distributed manner. Thus, our contribution in this paper is a distributed sub-network scheduling algorithm that achieves higher network sum-rate compared to interference-avoidance, especially when more than one-hop of network state information is made available at each node.

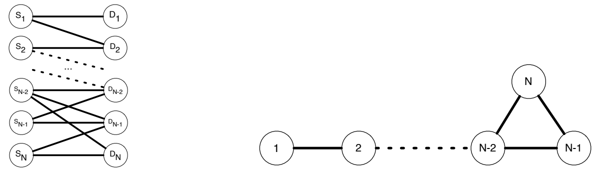
II. Problem Formulation

We consider a wireless network in an interference network model scenario. The interference network model consists of N source-destination pairs (users). The source nodes, labeled S_i, are connected to a subset of the destination nodes, labeled D_j, if the received power at destination D_j from S_i is above some threshold. We assume there is always a connection between S_i and D_j, for all $i = j$. The channel gain between S_i and D_j is denoted H_{ij}. The received signal at receiver D_j is $Y_j = \sum_{i:S_i \text{ is connected to } D_j} H_{ij} X_i + W$, where X_i is the transmit signal from S_i subject to its average power constraint P_i and W is complex Gaussian noise, $\mathcal{CN}(0,1)$.

Associated with the interference network is a *conflict graph*, $G(V,E)$. In this conflict graph, a vertex $v \in V$ represents a user in the interference network and an edge $e \in E$ represents

P. E. Santacruz and A. Sabharwal are with the Department of Electrical and Computer Engineering, Rice University, Houston TX 77005 (email: {pesantacruz,ashu}@rice.edu). V. Aggarwal is with AT&T Labs-Research, Florham Park NJ 07932 (email: vaneet@research.att.com).

interference between two users. Figure 1 depicts an example interference network and its corresponding conflict graph. It is important to note that, since our conflict graph is undirected several interference networks result in the same conflict graph.



(a) Interference Network

(b) Conflict Graph, G

Fig. 1: Example N-node Line-clique

A user is said to have τ hops of connectivity information if it knows all vertices and edges τ hops away from it in the conflict graph G. Similarly, a user has η hops of channel information if it has knowledge of all channel gains in the interference network for all users η hops away in the conflict graph. Notice that η hops of channel knowledge in conflict graph equals $h = 2\eta + 1$ hops of channel knowledge in the interference network.

Our performance metric will be *normalized sum-rate*, α, introduced in [8], which is the information-theoretic sum-rate achieved normalized by the sum-capacity with full network state information. More precisely, a normalized sum-rate of $\alpha(\eta)$ with η hops of channel state information in the conflict graph is achievable if there exists a strategy that allows transmission at rates R_i for each user $i \in \{1, 2, ..., N\}$ with error probabilities going to zero and satisfying

$$\sum_{i=1}^{N} R_i \geq \alpha(\eta) C_{sum} - \epsilon \qquad (1)$$

for all topologies consistent with the local view information, regardless of the realization of the channel gains. Here C_{sum} is the sum capacity of the network with full information and ϵ is some constant independent of channel gains.

III. Related Work

A. Independent set scheduling

In [1], the capacity of a constrained queueing system for an interference-avoiding PHY was derived and characterized. The problem was shown to be equivalent to finding independent sets in the network graph. The original work has been extended significantly over the years with many important contributions; see e.g. [2]-[5] and references therein. In the absence of queueing constraints, independent set scheduling is equivalent to a graph coloring problem. The problem of minimizing the number of required colors to color a graph has been a widely studied, see [10] and references therein. In [11] it was shown that coloring a graph with the optimal number of colors, defined as the *chromatic number*, is an NP-hard problem, even for a centralized algorithm with full connectivity information. In this paper, we use an algorithm by Kuhn [10], which is a distributed scheduling algorithm that gives each node in a graph several colors and ensures that each node receives at least a fraction $1/(\Delta + 1)$ of the total number of colors, where Δ is the maximum degree of the graph.

B. Independent graph scheduling

In [8], the authors examine optimal schemes with local channel state and global connectivity knowledge, and propose MIG Scheduling which is a strategy that is information-theoretically optimal for various classes of networks. Information-theoretic optimality means that there exists no other physical layer coding strategy which can achieve higher sum-rates given the amount of knowledge available. In MIG Scheduling, the network is separated into sub-networks, where each sub-network can achieve a normalized sum-capacity of 1 (i.e. into sub-networks with enough local knowledge to simultaneously transmit in an optimal way).

In MIG Scheduling, the sub-networks are labeled as *independent sub-graphs*. MIG scheduling divides the network into t independent sub-graphs, $\mathcal{A}_1, ...\mathcal{A}_t$ (not all distinct), and each user i belongs to d_i independent sub-graphs and achieves a normalized sum-rate of $\min_{i \in 1,2,...N} d_i/t$. The set of independent sub-graphs, $\mathcal{A}_1, ...\mathcal{A}_t$, that maximizes this value is called the MIG schedule.

The problem of finding the MIG schedule for an arbitrary network is a difficult task, even with complete connectivity knowledge, and is particularly challenging with only local connectivity information at each node. In this paper, we will use independent graph scheduling as a starting point and focus on identification of independent sub-graphs in a distributed fashion with only (τ, η)-hops of knowledge about the network state.

IV. Overview of Proposed Algorithm

In this section, we describe our algorithm to distributedly find independent sub-graphs as required by MIG Scheduling. The algorithm consists of two phases. In the first phase, we divide the network into connected sub-networks of diameter at most ρ such that each sub-network independently achieves a normalized sum-capacity of 1. In the second phase, we arrange several of these connected sub-networks into independent sub-graphs that still achieve normalized sum-capacity of 1 by using a distributed coloring algorithm that assigns the same colors to whole sub-networks. Our algorithm is parametrized by ρ, the maximum diameter of the connected sub-networks being identified. Given a ρ, we assume that each node has at least $\eta = \rho + 1$ hops of channel knowledge and $\tau = 3\rho + 3$ hops of connectivity information; these quantities will be explained in detail in the following sections.

In Phase 1, we leverage the local knowledge available at each node by finding r-cliques for $r \leq \rho$. An r-clique in a graph $G = (V, E)$ is a subgraph, $G[S]$, induced by a subset of nodes $S \subset V$ that satisfies 3 conditions:

1) Every node in $G[S]$ is at most a distance r hops away from all other nodes in $G[S]$.
2) The diameter of $G[S]$ is exactly r.
3) There is no $S' \subset V$ that also satisfies conditions 1 and 2 and such that $S \subset S'$, i.e., $G[S]$ is a maximal subgraph.

Note that a single node is a graph by itself and a 0-clique according to the above definition.

Phase 1 consists of identifying and selecting r-cliques, for $r = 0, ..., \rho$, in the conflict graph, G. We consolidate the selected r-cliques into single vertices and generate a consolidated graph G_ρ where each vertex represents an r-clique,

$r = 0, ..., \rho$, from the conflict graph G and an edge exists between two vertices in the consolidated graph if there exists an edge between members of the two cliques in the original conflict graph.

Phase 2 of the general procedure is performed by applying the distributed multicoloring algorithm by Kuhn [10] to the consolidated graph, G_ρ, which results in the assignment of time slots to each one of the cliques. The set of cliques with the same color are defined as an Independent Clique Set. An Independent Clique Set achieves $\alpha(\rho+1) = 1$ because each clique achieves $\alpha(\rho+1) = 1$ and the cliques do not interfere with each other. When we assign a time slot to each one of the Independent Clique Sets we create a scheme for Independent Graph Scheduling. We have chosen Kuhn's multicoloring algorithm because it requires only one round of communication and ensures that each vertex in the graph being colored receives at least a fraction $1/(\Delta+1)$ of the total colors assigned. We note that our metric of normalized sum-rate is directly related to the time slots assigned to the worst-case user [8]. Thus, given a fixed number of cliques containing a specific user, it is desirable to use the multicoloring algorithm in consolidated graphs which have smaller Δ.

With the objective of finding the consolidated graph G_ρ with a smaller maximum degree, Δ, our major innovations emerge from Phase 1 in the conversion of G to G_ρ. Phase 1 can be separated into two different steps. In Step 1, each node identifies all the potential cliques it can belong to. This step is mainly governed by the amount of network information available so we identify r-cliques (for $r \leq \rho$) that achieve $\alpha(\rho+1) = 1$. Step 2 consists of selecting which of the potential cliques from Step 1 will become vertices in the consolidated graph G_ρ. This step is crucial to ensure that agreement in the distributed coloring process of Phase 2 and overall improvement in normalized sum-rate are possible.

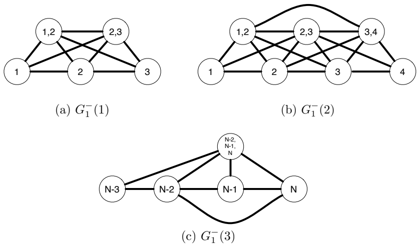
V. Step 1: Clique Identification ($G \to G_\rho^-(v)$)

In Step 1 of Phase 1, for a given ρ, we identify the r-cliques, $r = 0, ..., \rho$, which can be formed with $\rho + 1$ hops of connectivity information. We are interested in these r-cliques because with a parameter ρ we assume that enough knowledge is available such that each r-clique can achieve $\alpha(\rho+1) = 1$. These potential cliques are the candidates to ultimately be represented by a vertex in the consolidated graph G_ρ.

Since each node has a different local view of the conflict graph, G, the potential cliques discovered by each node will be different. Thus, in Step 1, each node will generate a temporary graph where the potential cliques it sees are turned into vertices. We will denote the temporary graph from the point to view of node v as $G_\rho^-(v) = (W^-(v), F^-(v))$. The set of vertices $W^-(v)$ represents all the r-cliques ($r \leq \rho$) in the part of the graph known to node v with $\rho + 1$ hops of connectivity information. Each node in $w \in W^-(v)$ maps to a set of nodes in the original conflict graph; we denote that set of nodes in the conflict graph represented by vertex w as $nodes(w)$. An edge exists between two vertices in $G_\rho^-(v)$, w_1 and w_2, if there is an edge between a member of $nodes(w_1)$ and a member of $nodes(w_2)$ in G or if a member of $nodes(w_1)$ is also a member of $nodes(w_2)$.

The following example shows the construction $G \to G_\rho^-(v)$ with the parameter $\rho = 1$ using the example original conflict graph G in Figure 1(b). Figure 2(a) shows the graph of all potential vertices from the point of view of node 1, which has 2 hops of connectivity knowledge. The vertices are labeled according to their corresponding set of nodes from the original conflict graph (in other words, the label of node w is $nodes(w)$). Node 1 observes 5 potential cliques, three 0-cliques ($\{1\}, \{2\}, \{3\}$) and two 1-cliques ($\{1,2\}, \{2,3\}$). There exists an edge between the vertices labeled $\{1\}$ and $\{1,2\}$ because 1 is present in both vertices and because there is an edge between 1 and 2 in the original conflict graph. Similarly, there are edges between $\{1,2\}$ and $\{2\}$, between $\{1\}$ and $\{2\}$, and so on.

Figure 2(b) depicts the graph $G_1^-(2)$ and Figure 2(c) describes the graph $G_1^-(N)$.



(a) $G_1^-(1)$

(b) $G_1^-(2)$

(c) $G_1^-(3)$

Fig. 2: Step 1 example for nodes 1, 2, and N with $\rho = 1$

The clique identification process can be extended for any $\rho > 1$ by identifying all r-cliques, for $r = 0, ..., \rho$. For example, if $\rho = 2$, $G_2^-(v)$ would consist of the full $G_1^-(v)$ plus all the 2-cliques in the 2-neighborhood of v, along with their respective edges. As we have mentioned, a larger ρ would increase the minimum amount of information required at each node. Also, finding maximal r-cliques is in general a hard problem, but since our goal is to leverage local information, we primarily concentrate on the cases of small r.

Once the potential optimally-transmitting r-cliques have been identified, a subset of them will be selected in Step 2 of Phase 1. Step 2 addresses two objectives. First, there is the issue that each node now has a graph with a maximum degree that is significantly higher than the maximum degree of the original conflict graph. As we have described before, the maximum graph degree and normalized sum-rate achieved by our scheme are intimately related, so our goal is a consolidated graph, G_ρ, with small degree. The second issue is the fact that we need to ensure that a distributed coloring algorithm does not lead to coloring conflicts, especially since the graphs seen by different nodes differ so much from each other.

VI. Step 2: Clique Selection ($G_\rho^-(v) \to G_\rho(v)$)

In this section, we will describe the second step of Phase 1, which consists of selecting which of the potential vertices in $G_\rho^-(v)$ identified by each node in Step 1 will be pruned and which will be kept in their own view of the final consolidated graph $G_\rho(v)$. We say user v in G is represented by vertex w in $G_\rho(v)$ if $v \in nodes(w)$.

The simplest way to guarantee improvement in normalized sum-rates is with a selection algorithm that satisfies two properties:

1) Each node v from the conflict graph G is represented by *only one* node in the consolidated graph $G_\rho(v)$.

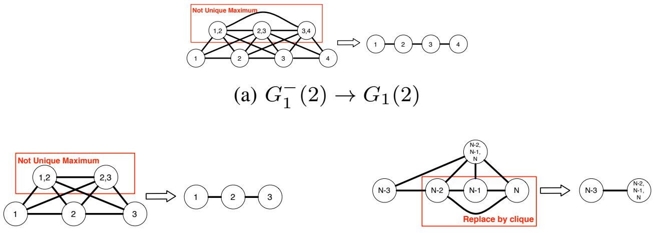
2) The degree of the vertex that represents v in the consolidated graph $G_\rho(v)$ is less than or equal to the degree of v in the conflict graph G.

These two simple properties ensure that the procedure will achieve a normalized sum-rate of $\alpha(\rho+1) = 1/(\Delta_{G_\rho}+1)$, where Δ_{G_ρ} is the maximum over all maximum degrees of the $G_\rho(v)$ graphs.

In Step 1, we assumed $\rho+1$ hops of connectivity information. Step 2 assumes $3\rho+3$ hops of connectivity knowledge. The reason behind this significant amount of knowledge required is that each node needs to know not only its own consolidated graph, but also the consolidated graphs of its neighbors to avoid coloring conflicts in Phase 2 since the distributed multicoloring algorithm is a process with a 1-hop footprint. We label this selection algorithm in Step 2 as Algorithm $\mathcal{A}_1(3\rho+3)$.

A. Example

Let $\rho = 1$ and consider the temporary graphs of Figure 2. For illustrative purposes, let us begin with graph $G_1^-(2)$ and node 2's selection process. According to our required properties, node 2 can only be represented by one vertex in the graph $G_\rho(2)$. Node 2 must choose one of the three options $\{1,2\}$, $\{2,3\}$, or $\{2\}$, and the others must be pruned. When possible, we want to keep vertices that represent more nodes since this could result, intuitively, in consolidated graphs with fewer vertices and with smaller maximum degree. The first step is to check if there exists a vertex that is a unique ρ-clique with the maximum number of members. In this case there are two cliques with 2 members each. Since we do not have a heuristic to prefer clique $\{1,2\}$ over clique $\{2,3\}$, in order to avoid conflicts with neighboring nodes about the clique chosen, node 2 concludes that neither of the two ρ-cliques will be selected and it chooses the vertex $\{2\}$ as its vertex to keep. Similarly, it chooses all the single node vertices for all other nodes in the graph as shown in Figure 3(a).



(a) $G_1^-(2) \to G_1(2)$

(b) $G_1^-(1) \to G_1(1)$

(c) $G_1^-(N) \to G_1(N)$

Fig. 3: Selection Algorithm Example

Now, consider the graph $G_1^-(1)$ and which vertices node 1 will select to be kept in the consolidated graph, $G_\rho(1)$. Node 1 sees that node 2 does not have a unique ρ-clique, therefore vertices $\{1,2\}$ and $\{2,3\}$ cannot be kept and all the single nodes vertices are chosen as shown in Figure 3(b).

Finally, let us look at the graph $G_1^-(N)$ and node N's selection process. Node N has a unique maximum clique $\{N-2, N-1, N\}$, and in this case, it is also the maximum clique for nodes $N-1$ and N. Also, the degree of the vertex representing that clique in $G_\rho(v)$ would be 1, which is smaller than the degree of nodes $N-2$, $N-1$ and N in G, which is 2. Therefore, node N decides that it will be represented by the vertex $\{N-2, N-1, N\}$ in the consolidated graph $G_\rho(N)$ shown in Figure 3(c). After this selection process with $\rho = 1$, the graphs $G_\rho(v)$, for all $v \in G$, have at most a maximum degree of 2; hence, $\Delta_{G_\rho} = 2$. This means the proposed algorithm achieves a normalized sum-rate of $1/3$ which is an improvement over the simple distributed independent set scheduling which achieves a normalized sum-rate of 1/4.

Throughout the example we have only been concerned with ρ-cliques and single-node vertices (0-cliques). In cases where $\rho > 1$, all r-cliques in $G_\rho^-(v)$, for $0 < r < \rho$, are automatically pruned. Also note that if the diameter of the graph G is $3\rho+3$ or less, we no longer have incomplete connectivity information and we can use the techniques in [8] or [12].

B. Selection Algorithm $\mathcal{A}_1(3\rho+3)$ Description

Now that we have given a heuristic about the clique selection process, we go ahead and provide a formal description. We begin with a given ρ and the assumption that each node knows $3\rho+3$ hops of connectivity in the conflict graph $G = (V, E)$. After each node $v \in V$ performs the selection algorithm, they will have generated a graph $G_\rho(v) = (W(v), F(v))$, where each $w \in W(v)$ represents a subset of nodes from the graph G.

To initialize the algorithm, each node v in G finds the maximum ρ-clique it belongs to and checks to see if it is unique. This is easily performed by inspection of the graph $G_\rho^-(v)$. If the unique maximum ρ-clique exists, we call that vertex representing that clique $w^*(v)$, otherwise, node v will be represented by vertex $\{v\}$ in the consolidated graph. Given the existence of a unique maximum ρ-clique, node v needs to know the vertices representing every $u \in nodes(w^*(v))$.

With this knowledge, node v must find the degree of $w^*(v)$ if it were to be kept in the final consolidated graph. Node v also knows the clique representing every neighbor of every node in $nodes(w^*(v))$, since it has $3\rho+3$ hops of knowledge. That is, node v can create a set of these potential neighboring vertices $\mathcal{U}^-(v) = \{\bigcup_{z \in Z} w^*(z)\}$, where $Z = \bigcup_{u \in nodes(w^*(v))} \Gamma_G(u)$, where $\Gamma_G(u)$ is the set of neighbors of u in G, u inclusive.

Furthermore, v estimates which cliques from $\mathcal{U}^-(v)$ will be kept in the consolidated graph based on the $3\rho+3$ hops of knowledge since with this amount of knowledge v has access to all potential cliques neighboring each one of the members of $\mathcal{U}^-(v)$. Therefore, v is able to generate the set $\mathcal{U}(v)$, which consists of the vertices that v considers will be present in the consolidated graphs. The degree of $w^*(v)$ in $G_\rho(v)$ if it were selected is the number of members of $\mathcal{U}(v)$, $\delta_{w^*(v)} = |\mathcal{U}(v)|$.

If $\delta_{w^*(v)} \leq \delta_u$ for all $u \in nodes(w^*(v))$, then $w^*(v)$ is formed and it will appear as a clique in the consolidated graph $G_\rho(v)$. Otherwise, each $u \in nodes(w^*(v))$ will be represented by the 0-clique $\{u\}$ in $G_\rho(v)$. The summary of this algorithm can be found in Algorithm 1.

VII. Performance Analysis and Results

The main characteristic of this algorithm is that, by leveraging local information when $\rho \geq 1$, the normalized sum-rate is ensured to be greater than or equal than the normalized

Algorithm 1 Selection Algorithm $\mathcal{A}_1(3\rho+3)$

```
Input: Graphs $G_\rho^-(v)$ for each node $v \in V$ and $3\rho+3$ hops of connectivity information
Every node $v \in V$ find its unique maximum $\rho$-clique
if a unique maximum $\rho$-clique does not exists then
    The vertex labeled $\{v\}$ is in the final graph $G_\rho(v)$
else
    if $w^*(u) \neq w^*(v)$ for some $u \in nodes(w^*(v))$ then
        Every $u \in nodes(w^*(v))$ is represented by a vertex $\{u\}$ in the consolidated graph $G_\rho(v)$
    else
        if $\delta_{w^*(v)} > \delta_u$ for some $u \in w^*(v)$ then
            Every $u \in nodes(w^*(v))$ is represented by a vertex $\{u\}$ in the consolidated graph $G_\rho(v)$
        else
            $w^*(v)$ will be a vertex in the final graph $G_\rho(v)$
        end if
    end if
end if
The graph $G_\rho(v)$ consists of all the vertices representing cliques selected to be kept by node $v$.
```

sum-rate achieved by distributed multicoloring of the original network, G.

Theorem 1: Let $\alpha_1(\rho)$ be the normalized sum-rate of a network after applying Algorithm $\mathcal{A}_1(3\rho+3)$ to the original graph G. If $\alpha(0)$ is the normalized sum-rate achieved by distributed multicoloring of the original network, G, then $\alpha(0) \leq \alpha_1(\rho)$, for $\rho \geq 1$.

Proof: Since our overall distributed scheduling algorithm will conclude with the use of Kuhn's algorithm, the normalized sum-rate of the network is governed by the maximum degree of the final graph being scheduled. Using Kuhn's distributed multicoloring, $\alpha(0) = 1/(\Delta_G+1)$, where Δ_G is the maximum degree of the original graph, G. Now, Algorithm $\mathcal{A}_1(3\rho+3)$ ensures that, for every $v \in V$, the maximum degree of graph $G_\rho(v)$, $\Delta_{G_\rho(v)}$, is less than or equal to Δ_G. Since the proposed algorithm is an instance of Independent Graph Scheduling, the achievable normalized sum-rate is the fraction of active time slots of the worst-case user. Since Kuhn's multicoloring assigns a fraction of at least $1/(\Delta_{G_\rho(v)}+1)$ to each clique in $G_\rho(v)$ and each v only appears once in $G_\rho(v)$, the worst-case user is active a fraction $1/(\Delta_{G_\rho}+1)$. Therefore, for every $\rho \geq 1$, $\alpha(0) = \frac{1}{\Delta_G+1} \leq \frac{1}{\Delta_{G_\rho(v)+1}} = \alpha_1(\rho)$. ■

We compare our algorithm's performance to the distributed multicoloring algorithm of the original graph to highlight the advantages of leveraging local information. The distributed multicoloring algorithm serves as a reasonable baseline of performance for one-shot algorithms. In contrast, other algorithms such as distributed greedy scheduling [4] or randomized maximal schedulers [13] consist of rounds of exchanges to make decisions. By making our algorithm a one-shot algorithm, we ensure that the amount of knowledge required is constrained to $3\rho+3$ hops. This quantifiable guarantee cannot be made under algorithms that involve several rounds as knowledge about the network propagates with each round.

We have used normalized sum-rate as a metric for performance of our algorithm since it provides a guaranteed fraction of sum-capacity even though computing a net sum-rate for an arbitrary network is unavailable due to the open problem of the capacity of the general interference channel. Nevertheless, our net sum-rate can be upper-bounded if we fix the direct channel gains and consider different realizations of interference channels with interference management techniques.

Several sample topologies (excluded due to space) display cases where the upper bound on net sum-rate of our one-shot algorithm is higher than that of other round-based algorithms, such as distributed greedy scheduling [4]. Therefore, when interference is manageable, our algorithm has opportunities to perform better even in terms of *net* sum-rate. While this is encouraging, the actual net sum-rate remains dependent on network channel states and direct comparison becomes unclear. On the other hand, normalized sum-rate gives us a metric that is independent of channel state and guarantees that in the worst-case over all possible channel realizations, we can guarantee improvement in performance by leveraging local information.

As expected, our algorithm's ability to present gains is not only based on interference properties, but on topology and the amount of information available. There exist topologies where, even with manageable interference, our algorithm cannot achieve a higher net sum-rate than the greedy distributed scheduling, unless complete knowledge is available. One important note is that the greedy scheduling algorithm is *not* a one-shot algorithm and it results in maximal schedules in each time slot. If we used a greedy-scheduling-type algorithm to schedule our final graph $G_\rho(v)$, we can also guarantee improvement over greedy scheduling with no ρ-clique formation, thus furthering the case for scheduling beyond interference avoidance.

References

[1] L. Tassiulas and A. Ephremides, "Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks," *IEEE Trans. on Autom. Control*, Dec. 1992.

[2] X. Lin and N. B. Shroff, "The Impact of Imperfect Scheduling on Cross-Layer Congestion Control in Wireless Networks," *IEEE/ACM Trans. on Networking*, vol.14, no.2, pp. 302-315, Apr., 2006.

[3] L. Jiang and J. Walrand, "A Distributed CSMA Algorithm for Throughput and Utility Maximization in Wireless Networks," *IEEE/ACM Trans. on Networking*, vol.18, no.3, pp. 960-972, Jun., 2010.

[4] X. Wu, R. Srikant, and J.R. Perkins, "Scheduling Efficiency of Distributed Greedy Scheduling Algorithms in Wireless Networks," *IEEE Trans. on Mobile Computing*, vol. 6, no. 6, pp 595-605, Jun., 2007.

[5] A. Gupta, X. Lin, and R. Srikant, "Low-Complexity Distributed Scheduling Algorithms for Wireless Networks," *IEEE/ACM Trans. on Networking*, vol.17, no.6, pp. 1846-1859, Dec., 2009.

[6] A. El Gamal and Y. Kim, *Network Information Theory*. Cambridge, MA: Cambridge University Press, 2011.

[7] R. Irmer et al., "Coordinated multipoint: Concepts, performance and field trial results," *IEEE Comm. Magazine*, vol. 49, no. 2, Feb. 2011.

[8] V. Aggarwal, A. S. Avestimehr and A. Sabharwal, "On Achieving Local View Capacity Via Maximal Independent Graph Scheduling," *IEEE Trans. on Info. Theory*, vol. 57, no. 5, pp. 2711-2729, May 2011.

[9] V. Aggarwal, Y. Liu and A. Sabharwal, "Sum-capacity of Interference Channels with a Local View: Impact of Distributed Decisions," *IEEE Trans. on Info. Theory*, Oct. 2009.

[10] F. Kuhn, "Local Multicoloring Algorithms," in Proc. *Intl. Symposium on Theoretical Aspects of Computer Science*, pp. 613-624, Feb. 2009.

[11] R. M. Karp, "Reducibility Among Combinatorial Problems," *Complexity of Computer Computations*, 1972.

[12] K. Sutuntivorakoon, V. Aggarwal, A. S. Avestimehr, and A. Sabharwal, "Maximal k-Clique Scheduling: A Simple Algorithm to Bound Maximal Independent Graph Scheduling," *Proc. Allerton Conf. on Communication, Control, and Computing*, Sept., 2011.

[13] E. Modiano, D. Shah, and G. Zussman, "Maximizing Throughput in Wireless Networks via Gossiping," *SIGMetrics/Performance*, June, 2006.