Multihop Local Pooling for Distributed Throughput Maximization in Wireless Networks

Gil Zussman
Department of Electrical Engineering
Columbia University
New York, NY 10027
gil@ee.columbia.edu

Andrew Brzezinski
Fidelity Investments
Boston, MA 02210
Andrew.Brzezinski@fmr.com

Eytan Modiano
Massachusetts Institute of Technology
Cambridge, MA 02139
modiano@mit.edu

***Abstract*—Efficient operation of wireless networks requires *distributed* routing and scheduling algorithms that take into account interference constraints. Recently, a few algorithms for networks with primary- or secondary-interference constraints have been developed. Due to their distributed operation, these algorithms can achieve *only a guaranteed fraction* of the maximum possible throughput. It was also recently shown that if a set of conditions (known as Local Pooling) is satisfied, simple distributed scheduling algorithms achieve 100% throughput. However, previous work regarding Local Pooling focused mostly on obtaining abstract conditions and on networks with single-hop interference or single-hop traffic. In this paper, we identify several graph classes that satisfy the Local Pooling conditions, thereby enabling the use of such graphs in network design algorithms. Then, we study the multihop implications of Local Pooling. We show that in many cases, as the interference degree increases, the Local Pooling conditions are more likely to hold. Consequently, although increased interference reduces the maximum achievable throughput of the network, it tends to enable distributed algorithms to achieve 100% of this throughput. Regarding multihop traffic, we show that if the network satisfies only the single-hop Local Pooling conditions, distributed joint routing and scheduling algorithms are not guaranteed to achieve maximum throughput. Therefore, we present new conditions for Multihop Local Pooling, under which distributed algorithms achieve 100% throughout. Finally, we identify network topologies in which the conditions hold and discuss the algorithmic implications of the results.**

***Index Terms*—Stability, Distributed algorithms, Wireless networks, Local Pooling, Interference, Scheduling, Routing.**

I. Introduction

A major challenge in the design and operation of wireless networks is to jointly route packets and schedule transmissions to efficiently share the common spectrum among links in the same area. A *centralized* joint routing and scheduling policy that achieves the maximum attainable throughput region was presented by Tassiulas and Ephremides [25]. However, the lack of central control in wireless networks calls for the design of *distributed* algorithms. Such algorithms can usually achieve only a fraction of the maximum throughput. Recently, it has been shown by Dimakis and Walrand [13] that there exist network topologies in which distributed *scheduling* algorithms *achieve* 100% *throughput*. In this paper, we focus on identifying topologies in which distributed algorithms achieve 100% throughput and studying the effect of *multihop* interference on these topologies. We also provide conditions under which *joint routing and scheduling* algorithms achieve 100% throughput.

The policy of [25] applies to a multihop wireless network with a stochastic packet arrival process and is guaranteed to stabilize the network (i.e. provide 100% throughput) whenever the arrival rates are within the stability region. The results of [25] have been extended to various settings of wireless networks and input-queued switches. However, throughput optimal algorithms based on [25] require the repeated solution of a *global optimization problem*, taking into account the queue backlog information for every link in the network. For example, even under simple primary interference constraints[1] a maximum weight matching problem has to be solved in every slot. Obtaining a centralized solution to such a problem in a wireless network does not seem to be feasible, due to the overhead associated with continuously collecting the queue backlog information. Therefore, the design of *distributed algorithms* has attracted a lot of attention recently.

For single-hop traffic the joint problem reduces to a *scheduling* problem. Lin and Shroff [21] studied the impact of distributed imperfect scheduling on cross-layer rate control. Regarding primary interference constraints, they showed that using a *distributed maximal matching* algorithm along with a rate control algorithm may achieve 50% throughput. Similar results for different settings were also obtained in [10], [11], [20]. It was also proved in [10], [20], [24], [27] that under secondary interference constraints[2] the throughput obtained by a distributed maximal scheduling algorithm may be significantly smaller than the throughput under a centralized (optimal) scheduler. In particular, it was proved in [10] that a distributed algorithm may achieve as low as $1/8$ of the possible throughput.

Dimakis and Walrand [13] recently showed that although in *arbitrary topologies* the worst case performance of distributed maximal scheduling algorithms can be very low, there are some topologies in which 100% *throughput is achieved*. In particular, they consider a graph of interfering queues[3] and

[1]Primary interference constraints imply that each pair of simultaneously active links must be separated by at least one hop (i.e. the set of active links at any point of time constitutes a matching) [10], [21], [22].

[2]Secondary interference constraints imply that each pair of simultaneously active links must be separated by at least two hops (links). These constraints are usually used to model IEEE 802.11 networks [10], [24], [27].

[3]Such graph is constructed from the network graph according to the interference constraints and is referred to as interference or conflict graph [11].

study the performance of a *greedy maximal weight scheduling* algorithm (termed Longest Queue First - LQF) that selects the set of served queues greedily according to the queue lengths. They present sufficient conditions for such an algorithm to provide 100% throughput (notice that unlike a *maximum* weight solution, a *maximal* weight solution can be easily obtained in a distributed manner [16], [19]). These conditions are referred to as *Local Pooling* (LoP) and are related to the properties of all maximal independent sets in the conflict graph. The LoP conditions were recently generalized in [18] to provide conditions under which a greedy maximal weight matching algorithm obtains some guaranteed fractional throughput.

Identifying specific network topologies that satisfy LoP is important, since it enables the design of algorithms that either partition a wireless network into subnetworks with such topologies (e.g. via channel allocation) or add artificial interference constraints that create such topologies. Hence, in [6] a few interference graphs satisfying LoP were identified and it was proved that under primary interference constraints, tree network graphs yield interference graphs that satisfy LoP. Although some knowledge about LoP has been acquired, [13] provided mostly abstract conditions, while [6], [18] focused mostly on primary interference constraints. Despite the fact that these constraints may hold for specific technologies, they are not realistic in most practical settings. Therefore, in order to allow the development of algorithms that take advantage of LoP, we focus on identifying topologies of interference and network graphs that satisfy the LoP conditions, and on studying the effect of *multihop* interference on these topologies.

We first use the LoP conditions to identify several new classes of LoP-Satisfying interference graphs. It is shown that within the class of perfect graphs, chordal graphs, chordal bipartite graphs, cographs, and a subgroup of co-comparability graphs all satisfy LoP. These observations increase the number of graphs that are known to satisfy LoP by a few orders of magnitude. We also show that all odd rings with at least 9 nodes and all even rings with at least 6 nodes do not satisfy LoP. Using the latter observation, we show that all bipartite graphs that are not chordal bipartite do not satisfy LoP.

We use the acquired knowledge about graph classes that satisfy and fail LoP to study the effect of increased interference on LoP. We focus on a generalization of the primary (1-hop) and secondary (2-hop) interference models to a k-hop interference model [24], where k is termed the interference degree. We show that under any interference degree, tree network graphs yield interference graphs that satisfy LoP (i.e. under any interference degree distributed algorithms achieve 100% throughput in trees). We also show that in many cases, as k increases, it is more likely that the LoP conditions hold, and thereby, it is more likely that simple distributed algorithms achieve 100% throughput. Moreover, for every network topology, there is an interference threshold k^*, above which the corresponding interference graphs satisfy LoP. At first glance, it seems that since it is known that the worst case performance deteriorates as the interference degree increases [10], [27], the results are counter-intuitive. Yet, the actual meaning of the results is that in many topologies, as k increases, the resulting interference graph is such that distributed maximal weight scheduling achieves the maximum throughput instead of the worst case throughput.

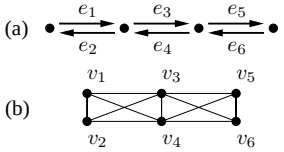
In general, networking environments in which the traffic is inherently *single-hop* and where packets must depart the system upon transmission across a link are rare. This results from the fact that many connections are necessarily multihop connections, due to geographical and physical constraints on user connectivity. Networks with *multihop traffic* have been studied in [20], [26], [27], where it was shown that, in general, only a fraction of the throughput is attainable when using distributed algorithms. Since the LoP results of [6], [13], [18] have been constrained to single-hop traffic, it is desirable to identify topologies in which distributed algorithms can obtain 100% throughput in the multihop traffic setting.

We show that the single-hop LoP conditions introduced in [13] are *insufficient* to guarantee stability in the multihop routing environment. Therefore, we study the LoP properties of a distributed routing and scheduling algorithm which is based on the backpressure mechanism of [25]. In this algorithm the edge weights are obtained by the backpressure mechanism but unlike in [25], a *distributed* maximal weight scheduling algorithm is used to determine which edges should be activated. We present new LoP conditions that are sufficient for guaranteeing that the algorithm achieves 100% throughput in the multihop routing environment. Then, we present a specific network topology that satisfies the multihop LoP conditions and show that the class of topologies satisfying these conditions is strictly included within the class of single-hop LoP-Satisfying graphs.

We note that the area of throughput maximization is somewhat related to the areas of distributed multicommodity routing [2] and adversarial queueing [3]. The algorithms of [2], [25] deal with a similar multihop setting by using similar backpressure methods. The adversarial queueing model (e.g. [3]) differs from the model of [25], mostly since usually in the former all edges can be simultaneously active, while the latter imposes constraints on the edge activations. Similarly to this work, previous works on adversarial queueing (e.g. [14]) also focused on identifying graph classes for which distributed algorithms are stable (in the sense of adversarial queueing stability).

The main contributions of this paper are two-fold. First, we identify several graph classes that satisfy Local Pooling, and show that as the interference degree increases, it is more likely that simple distributed algorithms achieve 100% throughput. The second contribution is the derivation of novel Local Pooling conditions for networks with multihop traffic. To the best of our knowledge this is the first attempt to study the multihop implications of Local Pooling. The obtained results can serve as a basis for the development of Local Pooling based algorithms.

This paper is organized as follows. In Section II we present the network model and the single-hop LoP conditions. In Section III we present several new classes of conflict graphs satisfying LoP. Then, in Section IV we discuss the effect of multihop interference on the satisfaction of the LoP conditions. New LoP conditions for networks with multihop traffic are presented in Section V. In Section VI we show that the multihop



(c) $\mathbf{M}(V_I) = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$

Fig. 1. (a) Network graph G_N, (b) the corresponding interference graph G_I under primary interference, and (c) the matrix of maximal link activations.

LoP conditions are distinct from the single-hop conditions and identify network topologies that satisfy them. We summarize the results and discuss future research directions in Section VII. Due to space constraints, the proofs are omitted and can be found in [5, Chapters 8 and 9]. We also recently dedicated [7] to a very detailed proof of Theorem 7.

II. Network Model and Stability

Consider a wireless network $G_N = (V_N, E_N)$, where $V_N = \{1, \ldots, n\}$ is the set of nodes, and $E_N \subseteq \{(i,j) : i,j \in V_N, i \neq j\}$ is a set of directed links indicating pairs of nodes between which data flows can occur, with $m \triangleq |E_N|$. The directionality of data flows across links necessitates the treatment of the network graph G_N as a directed graph. Depending on the circumstances, we denote links as either (i,j) or as e_k. In G_N, if two nodes $v_1, v_2 \in V_N$ are within communication range, then the directed edges $e_{12} = (v_1, v_2)$ and $e_{21} = (v_2, v_1)$ both belong to E_N. For a directed edge e, let $\sigma(e)$ denote the source (initial) vertex, and $\tau(e)$ denote the terminal (destination) vertex. Throughout the paper, bold symbols are associated with vectors and matrices.

The interference between network links can be summarized in an *interference graph* (or *conflict graph*) $G_I = (V_I, E_I)$ based on the network graph G_N [17]. We assign $V_I \triangleq E_N$. Thus, each edge e_k in the network graph is represented by a vertex v_k of the interference graph, and an edge (v_i, v_j) in the interference graph indicates a conflict between network graph links e_i and e_j (i.e. transmissions on e_i and e_j cannot take place simultaneously).[4] Fig. 1 contains a network graph G_N and the corresponding interference graph G_I under primary interference constraints.

Let $\Pi(G_N)$ denote the set of available link activations in the network graph G_N: the vector $\boldsymbol{\pi} = (\pi_e, e \in E_N) \in \Pi(G_N)$ is a 0-1 column vector representing a possible link activation. The set $\Pi(G_N)$ corresponds to all possible independent sets in the interference graph $G_I = (V_I, E_I)$. Under primary interference, $\Pi(G_N)$ corresponds to the set of matchings in G_N. We denote by $\mathbf{M}(V_I)$ the matrix of *maximal* independent sets in G_I; that is, the set of maximal column vectors in $\Pi(G_N)$. Continuing the example of Fig. 1, the matrix $\mathbf{M}(V_I)$ for interference graph G_I is contained in Fig. 1(c).

For simplicity, we assume that time is slotted and that packets are of equal size, each packet requiring one time slot of service across any link. There is no self-traffic. We will refer to packets destined to node $j \in V_N$ as *commodity j packets*. Let $A_{ij}(t)$ denote the number of exogenous commodity j packets that arrived at node i by the end of slot t. We assume that the arrivals have long term rates $\lambda_{ij} = \lim_{t\to\infty} A_{ij}(t)/t$, with overall system arrival rate vector $\boldsymbol{\lambda} = (\lambda_{ij},\, i,j \in V_N)$. Every node is assumed to have a queue for each possible destination. For $i,j \in V_N$, let $Q_{ij}(t)$ be the number of packets enqueued at node i at time t, whose destination in the network is node j. Assume that $Q_{ij}(0) = 0$ for all i,j.

Service is applied to the system at each time slot by activating a set of edges, and routing a packet of a single commodity across each active edge. We denote the corresponding $m \times n$ *service activation matrix* by $\mathbf{S} = (S_{ej}, e \in E_N, j \in V_N)$. Here, for edge $e \in E_N$ and commodity $j \in V_N$, S_{ej} can have value 0 or 1, depending on whether e is inactive or active for serving commodity j, respectively. Note that an admissible service activation matrix must have a valid underlying link activation belonging to $\Pi(G_N)$. This property characterizes the set of admissible service activation matrices, $\mathcal{S}$:

$$\mathcal{S} = \left\{ \mathbf{S} \in \{0,1\}^{m\times n} : \pi_e = \textstyle\sum_{j\in V_N} S_{ej}, \boldsymbol{\pi} \in \Pi(G_N) \right\}.$$

The matrix $\mathbf{S} \in \mathcal{S}$ leads to packet transitions in the network. Denote by $d_{ij}(\mathbf{S})$ the *net amount of service*, in packets per time slot, to queue Q_{ij} under activation matrix $\mathbf{S}$.

A. Stability Considerations

We can now define the stability region of the network.

Definition 1 (Admissible Rate Vector): A non-negative arrival rate vector $\boldsymbol{\lambda}$ is admissible, if there exists $L \geq 1$ and a collection of service activation matrices $\mathbf{S}^l \in \mathcal{S}$, $1 \leq l \leq L$ such that

$$\lambda_{ij} \leq \textstyle\sum_{l=1}^{L} \alpha_l d_{ij}(\mathbf{S}^l), \text{ where } \alpha_l \geq 0 \,\forall l, \textstyle\sum_{l=1}^{L} \alpha_l \leq 1.$$

The set of all admissible rate vectors is called the stability region and is denoted by $\boldsymbol{\Lambda}^*$.

At each time slot, a joint scheduling and routing algorithm makes a link activation and routing decision that must satisfy the interference constraints. A stable algorithm, which we also refer to as a throughput optimal algorithm or an algorithm that achieves 100% throughput, is defined as follows.

Definition 2 (Stable Algorithm): An algorithm is stable, if for any arrival process with rate vector $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}^*$, $\lim_{t\to\infty} Q_{ij}(t)/t = 0$ with probability 1 $\forall i,j \in V_N$.

This stability criterion is termed *rate stability* [1], [10].

Tassiulas and Ephremides [25] developed a stable joint routing and scheduling algorithm. At time $t \geq 0$, the algorithm computes for each edge $e \in E_N$ the maximum backpressure:

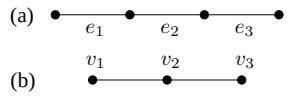
$$Z_e^*(t) = \max_{j\in V_N} (Q_{\sigma(e)j}(t) - Q_{\tau(e)j}(t)), \tag{1}$$

which we express in vector form as $\mathbf{Z}^*(t) = (Z_e^*(t), e \in E_N)$. Their algorithm then selects a link activation vector

$$\boldsymbol{\pi}^*(t) = \operatorname*{arg\,max}_{\boldsymbol{\pi}\in\Pi(G_N)} \boldsymbol{\pi}^T \mathbf{Z}^*(t). \tag{2}$$

Routing is carried out over each edge e having $\pi_e^*(t) = 1$, by serving a commodity achieving the maximum in (1) across that edge (for more details, see Section V-A).

[4] Although it has been recently shown that in some cases the conflict graph does not fully capture the wireless interference characteristics [23], it still provides a reasonable abstraction. Extending the results to general SINR-based constraints is a subject for further research.



(c) $\mathbf{M}(V_I) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$

Fig. 2. (a) Undirected network graph G_N, (b) the corresponding interference graph G_I under primary interference, and (c) the matrix of maximal link activations.

For general interference graph G_I, the algorithm of [25] must find the *maximum weight independent set* in G_I at each time slot[5] to obtain an optimal solution to (2). Namely, it must solve an NP-hard problem in every time slot or time frame. Under primary interference, the graph is simpler and the algorithm has to schedule the edges of a *maximum weight matching* in the network graph at each slot. This requires $O(n^3)$ computation time, using a centralized algorithm. In wireless networks, implementing a centralized algorithm is often not feasible and simple distributed algorithms usually obtain an approximate solution, resulting in a fractional throughput. Hence, in general graphs, even under simple interference constraints it is difficult to obtain 100% throughput in a distributed manner. This motivates us to study in which graph topologies simple distributed algorithms *can* obtain 100% throughput.

B. Simplifications for Single-Hop Traffic

When the network is subjected exclusively to single-hop traffic, a few simplifications occur in the model (see e.g. [1], [10], [22]). In this case, by definition, each network link e can only carry traffic destined to the terminal node of e. In other words, link e can only carry traffic of commodity $\tau(e)$. Thus, the differential backlog (backpressure) of link e equals the queue backlog of commodity $\tau(e)$. The algorithm of [25] then specializes to require that single-hop service be applied at each time t to a link activation vector

$$\boldsymbol{\pi}^*(t) = \underset{\boldsymbol{\pi} \in \Pi(G_N)}{\arg\max}\, \boldsymbol{\pi}^T \mathbf{Q}(t). \tag{3}$$

Above we understand $\mathbf{Q}(t)$ as the vector $\mathbf{Q}(t) = (Q_e(t), e \in E_N)$, where $Q_e(t)$ is the queue backlog of packets awaiting single-hop service across link e.

Since routing plays no role in the single-hop scenario, it is convenient to treat the network graph G_N as undirected. This simplifies the interference graph (an example of an undirected graph and its interference graph under primary interference appears in Figs. 2(a)-(b)). In this case, the weight at time t of each undirected edge $e = (u, v)$ equals the maximum weight of the queues that can be served across that link: $\max\{Q_{uv}(t), Q_{vu}(t)\}$. We will adopt this convention in our study of Local Pooling in sections III and IV.

C. Local Pooling for Single-Hop Traffic

We briefly reproduce important definitions and implications of Local Pooling (LoP) in networks with single-hop traffic, presented in [6], [13]. In Section V we will introduce the LoP conditions for the *multihop* traffic case. Recall that $\mathbf{M}(V_I)$ is the collection of maximal independent vertex sets on G_I, organized as a matrix (an example appears in Fig. 2(c)). We designate by $\mathbf{e}$ the vector having each entry equal to unity. We deliberately avoid specifying its size, because it will be obvious by the context of its use.

Definition 3 (Subgraph Local Pooling - SLoP): An interference graph G_I satisfies SLoP, if there exists nonzero $\boldsymbol{\alpha} \in \mathbb{R}_+^{|V_I|}$ and $c > 0$ such that $\boldsymbol{\alpha}^T \mathbf{M}(V_I) = c\mathbf{e}^T$.

Definition 4 (Overall Local Pooling - OLoP): An interference graph G_I satisfies OLoP, if each induced subgraph over the nodes $V \subseteq V_I$ *satisfies SLoP.*

Continuing with the example of Fig. 2, we can see that SLoP is satisfied for the interference graph G_I using the vector $\boldsymbol{\alpha}^T = (1\ 2\ 1)$: $\boldsymbol{\alpha}^T \mathbf{M}(V_I) = 2\mathbf{e}^T$. In a similar manner, it can be easily shown that all subgraphs of G_I satisfy SLoP, and therefore, G_I satisfies OLoP.

We can now describe the stability of the system when the service in each time slot is scheduled according to the Maximal Weight Independent Set (MWIS) algorithm. This algorithm is an iterative greedy algorithm that selects the node of G_I with the longest corresponding queue, and removes it and its neighbors from the interference graph. This process is repeated successively until no nodes remain. When multiple queues have the same length, a tie-breaking rule is applied. The set of selected nodes is a maximal independent set in the interference graph. Such a greedy algorithm can be implemented in a distributed manner and has the following property.

Theorem 1 (Dimakis and Walrand, 2006 [13]): If interference graph G_I satisfies OLoP, a Maximal Weight Independent Set (MWIS) scheduling algorithm achieves 100% throughput.

The fact that a graph satisfies OLoP *does not* guarantee that a MWIS algorithm obtains an optimal solution to the *maximum* weight independent set problem in that graph. When OLoP is satisfied, despite the fact that in some time slots the MWIS algorithm obtains an approximate solution to the maximum weight problem, 100% throughput *is achieved*.

III. Interference Graphs Satisfying Local Pooling

Since Theorem 1 and Definitions 3 and 4 do not provide a clear intuition regarding the graphs that satisfy OLoP, the properties of such graphs are only beginning to be understood. Small graphs were studied by exhaustive search [6]. Additionally, structural properties were used in [6], [13] to show that the following interference graphs satisfy OLoP: trees, forests, *clique trees*, where each pair of cliques shares at most a single vertex, and a *pair-of-cliques* connected by disjoint edges.

In this Section, we use structural properties to identify various graph classes that satisfy OLoP. We define a new class of graphs as the *OLoP-Satisfying* class. We identify known graph classes that are included within this class or intersect with it. It turns out that all the graph classes that we identify using structural properties are subclasses of the class of perfect graphs.[6] On the other hand, some of the graphs identified by the exhaustive search [6] are not perfect graphs. Hence, in the following discussion we differentiate between perfect and non-perfect graphs. Our investigation leads to the taxonomy of

[5] It can be shown that throughput optimality is maintained when solutions are obtained at bounded time frames that are longer than a time slot (e.g. [22]).

[6] A graph is *perfect*, if for each induced subgraph the size of the largest clique equals the chromatic number.

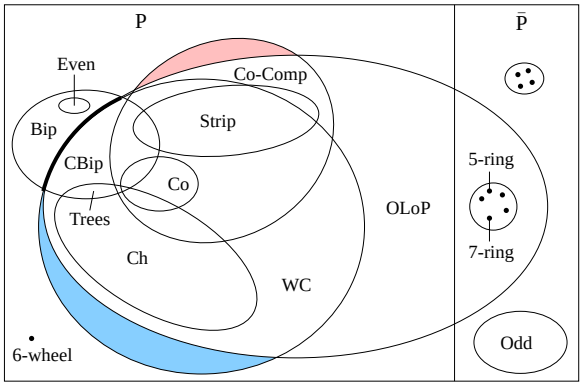


Fig. 3. The relations between the OLoP-Satisfying class and other graph classes: P - perfect, $\bar{P}$ - non-perfect, WC - weakly chordal, Ch - chordal, CBip - chordal bipartite, Bip - bipartite, Co - cograph, Co-Comp - co-comparability, Strip - strip-of-cliques, Even - cycles C_n with n even and $n \geq 6$, Odd - graphs with induced C_n with n odd and $n \geq 9$.

graph classes depicted in Fig. 3, showing the relationship of the OLoP-Satisfying class to the graph classes considered here.

We will make use of the following graph properties and definitions. For graph $G = (V, E)$, the *induced subgraph* over vertex set $V' \subseteq V$ is the graph $G' = (V', E')$, where E' is the set of edges in E whose endpoints are in V'. The complement $\overline{G} = (V, \overline{E})$ of graph $G = (V, E)$ is defined by $\overline{E} = (u, v) : u, v \in V, u \neq v$ and $(u, v) \notin E$. A *chord* of a cycle (path) is an edge between two vertices of the cycle (path) that is not an edge of the cycle (path). A cycle (path) is *chordless*, if it contains no chords. We denote by C_n and P_n a chordless cycle and a chordless path, respectively, of length n. We denote by K_n a clique (complete graph) of n nodes.

A. Perfect Graphs

Several classical graph classes such as bipartite graphs, chordal graphs, comparability graphs, and their complements are perfect [4]. Here, we will identify a number of important classes of perfect graphs that are also subclasses of the OLoP-Satisfying class. We will show that all of the graphs identified in [6], [13] are *simple* special cases in these classes. Before describing the results we introduce some classes of perfect graphs [4].

Definition 5: A graph G is chordal *if each cycle in G of at least 4 nodes has at least one chord. A graph G is* weakly chordal *if G and its complement contain no induced chordless cycle C_n, $n \geq 5$. A bipartite graph B is* chordal bipartite *if each cycle in B of length at least 6 has a chord. A graph is a* cograph *if it does not contain the path graph P_4 (depicted in Fig. 2(a)) as an induced subgraph.*

Notice that the chordal bipartite class is the intersection of the weakly chordal and bipartite classes.

The following theorem summarizes five results concerning the OLoP properties of several large graph classes. The proof can be found in [5, Chapter 8].

Theorem 2:

1) *The following graph classes belong to the OLoP-Satisfying class: Chordal Graphs, Chordal Bipartite Graphs, and Cographs.*
2) *All even cycles C_n with $n \geq 6$ fail SLoP.*

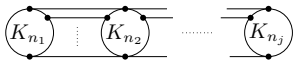


Fig. 4. The structure of a strip-of-cliques.

3) *Bipartite Graphs that are not Chordal Bipartite Graphs do not belong to the OLoP-Satisfying class.*

Fig. 3 illustrates the inclusion of the chordal, chordal bipartite, and cograph classes within the OLoP-Satisfying class. The class of chordal graphs has a few notable subclasses (i.e. classes of special graphs that are known to be chordal), including the strongly chordal, split, interval, threshold, and tree classes (for more information see [4]). Theorem 2 implies that all these subclasses satisfy OLoP. Therefore, the observation of [13] that trees satisfy OLoP immediately follows from Theorem 2. Similarly, since a clique tree is chordal, the observation of [6] that a clique tree satisfies OLoP is also a result of the theorem. Theorem 2 also implies that all subclasses of chordal bipartite graphs satisfy OLoP, including the convex and bipartite $\cap$ distance-heriditary classes.

The final contribution of Theorem 2 is its characterization of a *sharp* boundary separating the chordal bipartite graphs (OLoP-satisfying) from the bipartite graphs that are not chordal bipartite (not OLoP-satisfying). This boundary is depicted as a thick line in Fig. 3. This result follows directly from the failure of the OLoP conditions in even cycles C_n with $n \geq 6$. Hence, any graph class that includes the bipartite graphs as a subclass cannot be fully included within the OLoP-Satisfying class. This allows us to exclude many of the major subclasses of perfect graphs (e.g. preperfect, strongly perfect, quasi-parity, and bip* [4]) from the list of classes that can be fully included in the OLoP-Satisfying class.

Two major classes that have not been excluded as subclasses of the OLoP-Satisfying class are weakly chordal and co-comparability. In Fig. 3 we have shaded portions of the weakly chordal and co-comparability classes to indicate the uncertainty of their inclusion relations with OLoP-Satisfying. Determining the nature of these shaded regions (whether or not they exist) is left as an open problem.

We now present a subclass of the co-comparability class to which we refer as a *strip-of-cliques*. A graph is in this class, if it is composed from an ordered set of cliques $1, \ldots, j$, where two adjacent cliques i, $i + 1$ are connected by any number of disjoint edges, and cliques that are not adjacent are not connected directly. Fig. 4 illustrates such a graph. Notice that the pair-of-cliques presented in [6] is a specific case of a strip-of-cliques. The following lemmas show that a strip-of-cliques graph satisfies OLoP and that any such graph is a co-comparability graph.

Lemma 1: Every strip-of-cliques graph satisfies OLoP.

Lemma 2: Every strip-of-cliques is a co-comparability graph.

We finish this section by providing some context regarding the magnitude of the results. Consider the set of simple graphs having 7 nodes, of which there are 1,044 distinct graphs. Of these graphs, 393 are chordal, and 180 are cographs, with some overlap between these two classes. These numbers can be compared to the 37 forests and 11 trees that were known to satisfy

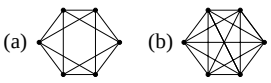


Fig. 5. (a) 2-hop and (b) 3-hop interference graphs of a 6-ring network graph

OLoP. Similarly, when considering the set of simple 11 node graphs, the number of chordal graphs is 1,392,387, compared to 710 forests and 235 trees. To summarize, our understanding of the OLoP-Satisfying class has expanded significantly beyond the trees and forest graphs.

B. Non-Perfect Graphs

The *OLoP-Satisfying* class includes also graphs that are not perfect. We first use the numerical observations of [6] to identify non-perfect graphs that satisfy OLoP. The 5-ring, C_5, which is the only 5-node non-perfect graph, satisfies OLoP. Moreover, since all 6-node graphs except C_6 satisfy OLoP, all non-perfect 6-node graphs satisfy OLoP. Finally, all 7-node graphs satisfy OLoP besides a specific one illustrated in [6] and those that have an induced 6-ring, which leads us to the observation that *134 out of the 138 non-perfect 7-node graphs satisfy OLoP*. In Fig. 3 all these graphs appear in a single class (containing C_5 and C_7) within the OLoP-Satisfying class. The 4 graphs that fail OLoP are represented in the top-right corner.

We now show that all non-perfect graphs that have an induced odd cycle with at least 9 nodes fail OLoP (see the Odd class in Fig. 3). This follows from the following theorem.

Theorem 3: All odd cycles C_n with $n \geq 9$ fail SLoP.

IV. Local Pooling under Multihop Interference

In this section, we show that counter-intuitively, *more interference often assists the operation of distributed algorithms*. Denote the stability region under k-hop interference by $\mathbf{\Lambda}_k^*$. It is clear that $\mathbf{\Lambda}_k^*$ cannot increase with k (and often decreases with k), as interference between the links of the network can only increase. Thus, although an increase in k can lead to a smaller stability region, such an increase makes it more likely that the OLoP conditions hold, and thereby more likely that simple distributed algorithms will achieve $\mathbf{\Lambda}_k^*$.

A. Interference Graphs

We first demonstrate the intuition on which the above observation is based. Consider the network graph C_6 (a 6 node ring), whose interference graph under primary interference is also C_6. According to [13], C_6 *does not satisfy OLoP* and, in general, a MWIS algorithm does not achieve 100% throughput. It has been recently shown in [18] that a MWIS algorithm guarantees 66% throughput in C_6. Under 2-hop interference, the interference graph has 6 more edges (see Fig. 5(a)). According to [6], this specific graph satisfies OLoP, and therefore, a MWIS algorithm achieves 100% throughput. Under 3-hop (or higher) interference, the interference graph becomes a clique (see Fig. 5(b)) which satisfies OLoP [6]. Hence, although under 1-hop interference, a maximal weight algorithm guarantees 66% throughput, under k-hop interference ($k \geq 2$) 100% throughput is guaranteed.

Under k-hop interference, the interference graph becomes an OLoP-Satisfying clique when k equals the network diameter. It seems reasonable to expect that for many network graphs, as the interference degree increases, there exists an *interference threshold* above which OLoP is satisfied. We tested this property by considering small graphs. In [6] it was shown that out of 1,252 simple interference graphs of up to 7 nodes, 14 fail OLoP. The following observation is obtained by exhaustively considering the corresponding k-hop ($k \geq 2$) interference graphs.

Observation 1: All k-hop ($k \geq 2$) interference graphs corresponding to network graphs with up to 7 edges satisfy OLoP.

Applying our acquired knowledge from Section III regarding the OLoP-Satisfying class, we will now proceed to study multihop interference properties of graphs. We focus on graph classes that appear in Fig. 3.

First, we indicate that due to Observation 1, a number of 1-hop interference graphs outside the OLoP-Satisfying class yield k-hop interference graphs that are OLoP-Satisfying. These graphs are the 6-ring, the 6-wheel, and the four non-perfect 7-node graphs outside the OLoP-Satisfying class.

We next introduce the Strongly Chordal class, a subclass of the chordal graphs, which exhibits an interference threshold property.

Definition 6 (Strongly Chordal [4]): A graph G is strongly chordal *if G is chordal and each cycle in G of even length at least 6 has an odd chord (a chord (i, j) is odd if the distance in the cycle between i and j is odd).*

Denote by G^k the k-th power of G: G^k has the same vertex set V as G, and $u, v \in V$ are adjacent in G^k, if the minimum path length between u and v in G is at most k. Given a 1-hop interference graph G_I^1, the corresponding k-hop interference graph is G_I^k.

Since the strongly chordal graphs belong to the chordal class, Theorem 2 implies that strongly chordal graphs are OLoP-Satisfying. A property of the strongly chordal class is that it is *strongly closed under power*. Namely, if an interference graph G_I^k is strongly chordal, then G_I^{k+j} is strongly chordal for all $j \geq 1$ [4]. Therefore, even if the 1-hop interference graph is not strongly chordal, once an interference graph becomes strongly chordal (and thereby OLoP-Satisfying), increased interference degree will generate OLoP-Satisfying graphs. Based on this property, the following theorem establishes that every graph has an interference threshold k^* *above which* all interference graphs satisfy OLoP.

Theorem 4: There exists a k^* such that for $k \geq k^*$, G_I^k satisfies OLoP.

The following lemmas show that certain graphs, identified in Section III-A, exhibit interference threshold $k^* = 1$ (Lemma 3 immediately follows from the above mentioned property of the strongly chordal class).

Lemma 3: If the 1-hop interference graph G_I^1 is a strongly chordal graph, such as a tree or a clique tree, then G_I^k satisfies OLoP for every $k \geq 1$.

Lemma 4: If the 1-hop interference graph G_I^1 is a cograph or a strip-of-cliques, then G_I^k satisfies OLoP for every $k \geq 1$.

When we study the transition from G_I^k to G_I^{k+1}, we find that

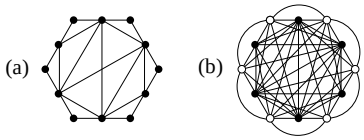


Fig. 6. (a) A chordal 1-hop interference graph and (b) the corresponding 2-hop interference graph that fails OLoP.

there *are* cases where increasing the interference degree can result in a graph that fails OLoP: although any interference graph has an interference threshold, the transition to this threshold may not be smooth. Namely, below the interference threshold, the interference graphs may alternate between being OLoP-Satisfying and OLoP-Failing for different values of k. The following lemma summarizes this result, and is based on the 1-hop interference graph in Fig. 6(a). The corresponding 2-hop interference graph appears in Fig. 6(b). It can be seen that the subgraph induced by the white nodes is C_6, which fails SLoP. Thus, OLoP fails in the overall graph.

Lemma 5: There are OLoP-Satisfying k-hop interference graphs for which OLoP is not satisfied in a corresponding j-hop ($j > k$) interference graph.

B. Network Graphs

Thus far, we have studied the LoP properties under multihop interference for most graphs represented in Fig. 3. We next turn our attention to particular *network graph* structures. An example of an interference graph G_I^1 resulting from 1-hop interference is given in Fig. 2. A second example is the ring network graph C_n, whose 1-hop interference graph is also C_n. Recall from Section III that C_n fails OLoP for $n = 6$ and $n \geq 8$. Our numerical tests show that the 2-hop interference graph of any C_n with $n \leq 8$ satisfies OLoP. Hence, we observe that rings are network graphs that benefit from additional interference degrees.

Clearly, any network graph whose corresponding interference graph is one of the structures indicated in Lemmas 3 and 4 satisfies OLoP for any $k \geq 1$. In particular, we can derive the following result.

Theorem 5: Distributed MWIS algorithms achieve 100% *throughput in a tree* network graph *under any interference degree k.*

The 2-hop interference model is important, since it represents the IEEE 802.11 transmission constraints [24], [27]. We obtain the following result that applies to this model by using results regarding squares of line graphs[7] [8], [9].

Theorem 6: Distributed MWIS algorithms achieve 100% throughput in a chordal network graph *under a k-hop interference model, with any even k.*

Several subclasses of chordal graphs have the potential to allow a MWIS algorithm to be throughput-optimal under a k-hop interference model, with even k. One of the subclasses is the class of interval graphs.[8] For that class the following stronger result holds.

Lemma 6: Distributed MWIS algorithms achieve 100% throughput in an interval network graph under a k-hop interference model, where $k \geq 2$.

V. LoP in Networks with Multihop Routing

In this section, we study the LoP properties in networks employing multihop routing, under *general* interference constraints. We present a simple adaptation to the framework of [25] that allows decentralized implementation by using a distributed MWIS scheduling algorithm. We obtain multihop local pooling conditions that are sufficient for guaranteeing 100% throughput under the presented algorithm.

A. Backpressure-based Routing and Scheduling

Recall from Section II-A that the optimal centralized scheduler (2) calculates *maximum* weight independent sets based on *backpressure* link weights. Instead, the algorithm presented below finds *Maximal* Weight Independent Sets (MWIS) based on the backpressure link weights. Similarly to the single-hop traffic setting [13], we use a MWIS algorithm, but unlike in [13], we use the backpressure link weights (instead of the queue backlogs). The MWIS algorithm operates on the interference graph and since it is a greedy algorithm, it can be easily implemented in a distributed manner (e.g. the algorithm of [16] that can be applied to a network with primary interference constraints). As in the single-hop case, the algorithm is *independent of the global network topology and traffic statistics*.

The Backpressure Routing and (Maximal) Scheduling (BRMS) algorithm is presented below. In step 4, the algorithm uses the MWIS algorithm as a subroutine in order to select a *maximal* weight link activation based upon maximum link backpressures, obtained in step 3 (notice that this is the difference from the algorithm of [25] that used *maximum* weight). In step 5, the algorithm makes routing decisions in order to serve commodities achieving maximum backpressure.

Algorithm 1 Backpressure Routing and (Maximal) Scheduling (BRMS)

1: **for** time index $t = 1, 2, \ldots$ **do**
2: For each directed edge $e \in E_N$ assign
$$Z_{e,j}(t) \leftarrow (Q_{\sigma(e)j}(t) - Q_{\tau(e)j}(t))$$
3: Assign $Z_e^*(t) = \max_j Z_{ej}(t)$
4: Obtain a maximal link activation $\boldsymbol{\pi}^*(t) \in \Pi(G_N)$ using a decentralized MWIS algorithm, based on the edge weight vector $\mathbf{Z}^*(t) = (Z_e^*(t), e \in E_N)$
5: For each $e \in E_N$ such that $\pi_e^*(t) = 1$, choose $j^* = \arg\max_j Z_{ej}(t)$. Route $\min\{1, Q_{\sigma(e)j^*}(t)\}$ packets of commodity j^* across e
6: **end for**

Recall that the OLoP conditions consider all possible vertex subsets of the interference graph, $V \subseteq V_I$. By the definition of the interference graph, the node set V corresponds to a subset

[7] In graph theoretic terminology, the interference graph resulting from 1-hop interference is called line graph [15].

[8] An interval graph is the intersection graph of a set of intervals on the real line. Intervals are represented by nodes and nodes are connected if they correspond to intervals that intersect [4], [9].

of the network graph edges, $E \subseteq E_N$. Thus, the OLoP conditions effectively consider every subset of network graph edges $E \subseteq E_N$. In the multihop routing scenario, we must again consider each set of network graph edges $E \subseteq E_N$. Therefore, we will refer to $\mathbf{M}(E)$ that similarly to $\mathbf{M}(V)$ includes all the maximal possible link activations. Since routing across network graph edges is not unique in the multihop scenario, we must *additionally* consider various combinations of commodities associated with network graph edges. We formalize the possible edge/commodity combinations by introducing the Maximum Commodity Family.

Definition 7 (Maximum Commodity Family): For $E \subseteq E_N$, $E \neq \emptyset$, the Maximum Commodity Family is given by $\mathcal{J}_E = \{(J_e^{\mathbf{Q}}, e \in E_N) : \mathbf{Q} \in \mathcal{Q}_E, \mathbf{Q} \neq 0\}$, *where*

$$\mathcal{Q}_E = \{(Q_{ij},\, i,j \in V_N, i \neq j) : Q_{ij} \in \mathbb{R}_+ \,\forall i,j, \\ E = \arg\max_e \max_j (Q_{\sigma(e)j} - Q_{\tau(e)j})\},$$

$$J_e^{\mathbf{Q}} = \{j \in V_N : j \neq \sigma(e), \\ Q_{\sigma(e)j} - Q_{\tau(e)j} \geq Q_{\sigma(e)j'} - Q_{\tau(e)j'} \;\forall j' \in V_N\}.$$

The Maximum Commodity Family $\mathcal{J}_E$ relates closely to a system of differential equations called a *fluid limit model* [12], derived from the queueing system (more details and an illustrative example can be found in [7]). The following definitions are necessary to introduce the multihop LoP conditions.

Definition 8 (Maximal Service Activation Set): For $E \subseteq E_N$ and $J = (J_e, e \in E_N) \in \mathcal{J}_E$, the Maximal Service Activation Set is given by:

$$\mathcal{S}_{E,J} = \big\{\mathbf{S} \in \mathcal{S} : \textstyle\sum_j \mathbf{S}_{Ej} \in \mathbf{M}(E), \\ S_{ej} = 1 \textit{ implies } j \in J_e \textit{ when } e \in E_N\big\}$$

Above, $\mathbf{S}_{Ej}$ is the vector $(S_{ej}, e \in E)$. The Maximal Service Activation Set $\mathcal{S}_{E,J}$ for a set of edges $E \in E_N$ consists of every service activation matrix whose underlying link activation is maximal over the edges in E. Recall that each edge $e \in E_N$ is a vertex in the interference graph G_I. In order to characterize the stability properties of the BRMS algorithm, we will track the dynamics of the link differential backlogs. Hence, we must understand how each service matrix $\mathbf{S} \in \mathcal{S}$ affects the distribution of commodity backpressures over the network links. We now introduce the Backpressure Service Vector (recall that the quantity $d_{ij}(\mathbf{S})$ is the amount of service at queue Q_{ij} resulting from applying service activation $\mathbf{S}$ for one time slot).

Definition 9 (Backpressure Service Vector): For $E \subseteq E_N$, $J = (J_e, e \in E_N) \in \mathcal{J}_E$, and service matrix $\mathbf{S} \in \mathcal{S}$, the Backpressure Service Vector $\mathbf{u}_{E,J}(\mathbf{S})$ contains the decrease *in differential backlog of commodity j across link e under service matrix $\mathbf{S}$ for every edge/commodity pair (e,j) where $e \in E, j \in J_e$:*

$$\mathbf{u}_{E,J}(\mathbf{S}) = ((d_{\sigma(e)j}(\mathbf{S}) - d_{\tau(e)j}(\mathbf{S})), e \in E, j \in J_e).$$

The Backpressure Service Vector $\mathbf{u}_{E,J}(\mathbf{S})$ tracks the change in backpressure incurred by a set of edge/commodity pairs, when a particular service activation matrix $\mathbf{S}$ is employed for a single time slot.

B. Stability of the BRMS Algorithm

Here, we study the stability of the BRMS algorithm, and introduce new LoP conditions for stability. We denote $\mathbf{d}(\mathbf{S}) = (d_{i,j}(\mathbf{S}),\, i,j \in V_N)$.

Definition 10 (Subgraph Multihop Local Pooling - SMLoP): The directed network graph $G = (V,E)$ satisfies SMLoP by commodity collection $J \in \mathcal{J}_E$, if there exist vectors $\boldsymbol{\alpha}, \boldsymbol{\beta} \geq 0$ with $\boldsymbol{\alpha} \neq 0$, and a constant scalar $c \geq 0$ such that

$$\boldsymbol{\alpha}^T \mathbf{u}_{E,J}(\mathbf{S}) + \boldsymbol{\beta}^T \mathbf{d}(\mathbf{S}) \leq c, \quad \forall \mathbf{S} \in \mathcal{S}, \tag{4}$$

$$\boldsymbol{\alpha}^T \mathbf{u}_{E,J}(\mathbf{S}) \geq c, \quad \forall \mathbf{S} \in \mathcal{S}_{E,J}. \tag{5}$$

The SMLoP conditions associate with each link/commodity pair (e,j) a non-negative weight $\alpha_{e,j}$, where $e \in E, j \in J_e$. Further, for each node/commodity pair (v,j), the conditions associate a non-negative weight $\beta_{v,j}$, where $v, j \in V_N$.

Definition 11 (Overall Multihop Local Pooling - OMLoP): The network graph $G_N = (V_N, E_N)$ satisfies OMLoP if SMLoP is satisfied by each subgraph $G'_N = (V_N, E)$ and commodity collection $J \in \mathcal{J}_E$, where $E \subseteq E_N$.

We next state the main theorem regarding the stability of the BRMS algorithm. A detailed proof appears in [7].

Theorem 7: If network graph G_N satisfies OMLoP, then the BRMS algorithm achieves 100% throughput.

Theorem 7 demonstrates the sufficiency of the OMLoP conditions for stability under the BRMS algorithm. In the next section, we consider natural questions that arise out of these conditions.

VI. Studying the OMLoP Conditions

We now show that the OMLoP conditions are distinct from the single-hop Local Pooling conditions studied in [6], [13], and demonstrate stability for a specific class of networks. We first show that any network graph G_N under which single-hop LoP fails should also fail the OMLoP conditions.

Lemma 7: If G_N fails OLoP, then it also fails OMLoP.

In terms of Fig. 3, this implies that the class of graphs that are not OLoP-Satisfying can not be OMLoP-Satisfying. Namely, all network graphs having interference graphs with induced subgraphs that are bipartite and not weakly chordal, or induced C_n when $n = 6$ or $n \geq 8$ must fail OMLoP. The next theorem demonstrates that the OMLoP conditions are in fact *more restrictive* than their single-hop counterparts. Thus, the family of OMLoP-satisfying graphs is *strictly* smaller than that depicted in Fig. 3. It was indicated in Section III-B that the 5-ring satisfies the single-hop OLoP conditions. Here we show that OMLoP fails for the 5-ring.

Theorem 8: The 5-ring (C_5) fails OMLoP.

We now verify that the OMLoP conditions hold for a class of graphs in which the BRMS algorithm is known to achieve 100% throughput. This class is the *forest of stars*, where every connected component of the network graph is a star graph, consisting of a central node v_0, and non-v_0 vertex degree of 1 for every edge incident on v_0. Under any k-interference model, the star's interference graph is a clique (appearing in Fig. 3 within the intersection region of the chordal and cograph classes). Therefore, only one edge can ever be active at once.

Accordingly, a maximal weight edge activation is identical to a *maximum* weight edge activation, thereby achieving 100% throughput. The following lemma shows that OMLoP is indeed satisfied in such graphs, and therefore, it is satisfied in a forest of stars.

Lemma 8: The star network graph satisfies OMLoP.

VII. Conclusions

The consideration of Local Pooling has the potential to enable efficient distributed operation of wireless networks. However, since previously LoP was studied mostly under the assumptions of single-hop traffic and primary interference, in this paper we focused on its multihop implications. We identified several graph subclasses of the OLoP-Satisfying class and increased the number of known graphs that satisfy LoP by a few orders of magnitude. Using these observations, we showed that as the interference degree increases, it is more likely that simple distributed algorithms achieve 100% throughput. For example, it was proved that under *secondary* interference constraints, a maximal weight scheduling algorithm achieves 100% throughput in chordal network graphs and that under any interference degree, such an algorithm achieves 100% throughput in trees. Moreover, we presented the LoP conditions for networks with multihop traffic (OMLoP) and showed that they are distinct from the single-hop conditions. Finally, we showed that the class of graphs satisfying the OMLoP conditions is a strict subclass of the OLoP-Satisfying class.

We emphasize that our objective in this paper is to obtain a better *theoretical* understanding of LoP that will assist the development of future algorithms. Hence, although a theoretical contribution has been made, there remain many algorithmic open problems. For example, LoP-based algorithms can partition the network into LoP-satisfying subnetworks or add artificial interference constraints to generate a LoP-satisfying network. Our identification of several LoP-satisfying graph classes that can serve as building blocks for these networks, and the understanding of multihop traffic and interference effects are advances toward such algorithms. For instance, one can now develop algorithms that add artificial edges to the interference graph to yield a chordal graph.

Moreover, there are a number of theoretical issues that remain unresolved. For example, Lemma 5 demonstrates that further study is necessary to determine the general evolution of the LoP property with varying interference degree. The complete characterization of the OLoP-Satisfying and the OMLoP-Satisfying graph classes is also a subject for further research. Finally, the effect of generalizing the interference model from a simplistic k-hop model to a model based on SINR remains a subject for future research.

Acknowledgement

This work was supported by NSF ITR grant CCR-0325401, ONR grant number N000140610064, a Marie Curie International Fellowship within the 6th European Community Framework Programme, and the MAGNET/ISRC Consortium.

References

[1] M. Ajmone Marsan, E. Leonardi, M. Mellia, and F. Neri, "On the stability of isolated and interconnected input-queueing switches under multiclass traffic," *IEEE Trans. Inform. Th.*, vol. 51, no. 3, pp. 1167–1174, Mar. 2005.

[2] B. Awerbuch and T. Leighton, "Improved approximation algorithms for the multi-commodity flow problem and local competitive routing in dynamic networks," in *Proc. ACM STOC'94*, 1994.

[3] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan, and D. P. Williamson, "Adversarial queuing theory," *J. ACM*, vol. 48, no. 1, pp. 13–38, Jan. 2001.

[4] A. Brandstädt, V. B. Le, and J. P. Spinrad, *Graph Classes: A Survey*. SIAM, 1999.

[5] A. Brzezinski, "Scheduling algorithms for throughput maximization in data networks," Ph.D. dissertation, MIT, May 2007. [Online]. Available: http://web.mit.edu/brzezin/www/thesis.pdf

[6] A. Brzezinski, G. Zussman, and E. Modiano, "Enabling distributed throughput maximization in wireless mesh networks - a partitioning approach," in *Proc. ACM MOBICOM'06*, Sept. 2006.

[7] ——, "Local pooling conditions for joint routing and scheduling," in *Proc. Inform. Th. and Appl. (ITA'08)*, Jan. 2008.

[8] K. Cameron, "Induced matchings," *Discrete Appl. Math.*, vol. 24, no. 1–3, pp. 97–102, 1989.

[9] ——, "Induced matchings in intersection graphs," *Discrete Math.*, vol. 278, no. 1–3, pp. 1–9, Mar. 2004.

[10] P. Chaporkar, K. Kar, X. Luo, and S. Sarkar, "Throughput and fairness guarantees through maximal scheduling in wireless networks," *IEEE Trans. Inform. Th.*, vol. 54, no. 2, Feb. 2008.

[11] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, "Optimal cross-layer congestion control, routing and scheduling design in ad hoc wireless networks," in *Proc. IEEE INFOCOM'06*, Apr. 2006.

[12] J. G. Dai, "On the positive Harris recurrence for open multiclass queueing networks: a unified approach via fluid limit models," *Ann. Appl. Probab.*, vol. 5, no. 1, pp. 49–77, Feb. 1995.

[13] A. Dimakis and J. Walrand, "Sufficient conditions for stability of longest queue first scheduling: second order properties using fluid limits," *Adv. Appl. Probab.*, vol. 38, no. 2, pp. 505–521, June 2006.

[14] A. Goel, "Stability of networks and protocols in the adversarial queueing model for packet routing," *Networks*, vol. 37, no. 4, pp. 219–224, 2001.

[15] F. Harary, *Graph Theory*. Addison-Wesley, 1969.

[16] J.-H. Hoepman, "Simple distributed weighted matchings," Oct. 2004, eprint cs.DC/0410047.

[17] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," *ACM/Springer Wireless Networks*, vol. 11, no. 4, pp. 471–487, July 2005.

[18] C. Joo, X. Lin, and N. B. Shroff, "Performance limits of greedy maximal matching in multi-hop wireless networks," in *Proc. IEEE CDC'07*, Dec. 2007.

[19] F. Kuhn, T. Moscibroda, and R. Wattenhofer, "What cannot be computed locally!" in *Proc. ACM PODC'04*, July 2004.

[20] X. Lin and S. Rasool, "Constant-time distributed scheduling policies for ad hoc wireless networks," in *Proc. IEEE CDC'06*, Dec. 2006.

[21] X. Lin and N. B. Shroff, "The impact of imperfect scheduling on cross-layer rate control in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 14, no. 2, pp. 302–315, Apr. 2006.

[22] E. Modiano, D. Shah, and G. Zussman, "Maximizing throughput in wireless networks via gossiping," in *Proc. ACM SIGMETRICS'06*, June 2006.

[23] T. Moscibroda and R. Wattenhofer, "The complexity of connectivity in wireless networks," in *Proc. IEEE INFOCOM'06*, Apr. 2006.

[24] G. Sharma, R. R. Mazumdar, and N. B. Shroff, "On the complexity of scheduling in wireless networks," in *Proc. ACM MOBICOM'06*, Sept. 2006.

[25] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Automat. Contr.*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.

[26] X. Wu and R. Srikant, "Regulated maximal matching: a distributed scheduling algorithm for multi-hop wireless networks with node-exclusive spectrum sharing," in *Proc. IEEE CDC'05*, Dec. 2005.

[27] ——, "Bounds on the capacity region of multi-hop wireless networks under distributed greedy scheduling," in *Proc. IEEE INFOCOM'06*, Apr. 2006.