Analyzing the Performance of Greedy Maximal Scheduling via Local Pooling and Graph Theory

Berk Birand*, Maria Chudnovsky†, Bernard Ries†, Paul Seymour‡, Gil Zussman* and Yori Zwols†
*Department of Electrical Engineering
†Department of Industrial Engineering and Operations Research
Columbia University, New York, NY
‡Department of Mathematics
Princeton University, Princeton, NJ

***Abstract*—Efficient operation of wireless networks and switches requires using simple (and in some cases distributed) scheduling algorithms. In general, simple greedy algorithms (known as Greedy Maximal Scheduling - GMS) are guaranteed to achieve only a fraction of the maximum possible throughput (e.g., 50% throughput in switches). However, it was recently shown that in networks in which the *Local Pooling* conditions are satisfied, GMS achieves 100% throughput. Moreover, in networks in which the σ-Local Pooling conditions hold, GMS achieves $\sigma\%$ throughput. In this paper, we focus on identifying the specific network topologies that satisfy these conditions. In particular, we provide the first characterization of *all the network graphs in which Local Pooling holds under primary interference constraints* (in these networks GMS achieves 100% throughput). This leads to a linear time algorithm for identifying Local Pooling-satisfying graphs. Moreover, by using similar graph theoretical methods, we show that *in all bipartite graphs (i.e., input-queued switches) of size up to* $7\times n$, *GMS is guaranteed to achieve* 66% *throughput,* thereby improving upon the previously known 50% lower bound. Finally, we study the performance of GMS in interference graphs and show that in certain specific topologies its performance could be very bad. Overall, the paper demonstrates that using graph theoretical techniques can significantly contribute to our understanding of greedy scheduling algorithms.**

***Index Terms*—Local pooling, scheduling, throughput maximization, graph theory, wireless networks, switches.**

I. Introduction

The effective operation of wireless and wireline networks relies on the proper solution of the packet scheduling problem. In wireless networks, the main challenge stems from the need for a decentralized solution to a centralized problem. Even when centralized processing is possible, as is the case in input-queued switches, designing low complexity algorithms that will enable efficient operation is a major challenge.

A centralized joint routing and scheduling policy that achieves the maximum attainable throughput region was presented by Tassiulas and Ephremides [27]. That policy applies to a multihop network with a stochastic packet arrival process and is guaranteed to stabilize the network whenever the arrival rates are within the stability region (i.e., it provides 100% throughput). The results of [27] have been extended to various settings of wireless networks and input-queued switches (e.g., [1], [9], [22]). However, algorithms based on [27] require the repeated solution of a *global optimization problem*, taking into account the queue backlog of every link. For example, even under simple primary interference constraints[1], a maximum weight matching problem has to be solved in every slot, requiring an $O(n^3)$ algorithm.

Hence, there has been an increasing interest in simple (potentially distributed) algorithms. One such algorithm is the *Greedy Maximal Scheduling* (GMS) algorithm (also termed Maximal Weight Scheduling or Longest Queue First - LQF). This algorithm selects the set of served links greedily according to the queue lengths [12], [20]. Namely, at each step, the algorithm selects the heaviest link (i.e., with longest queue size), and removes it and the links with which it interferes from the list of candidate links. The algorithm terminates when there are no more candidate links. Such an algorithm can be implemented in a distributed manner [12], [17], [18].

It was shown that the GMS algorithm is guaranteed to achieve 50% throughput in switches [7] and in general graphs under primary interference constraints [20]. It also was proved in [4], [25], [29] that under secondary interference constraints[2] the throughput obtained by GMS may be significantly lower than the throughput under a centralized scheduler.

Although in *arbitrary topologies* the worst case performance of GMS can be very low, there are some topologies in which 100% *throughput is achieved*. Particularly, Dimakis and Walrand [8] presented sufficient conditions for GMS to provide 100% throughput. These conditions are referred to as *Local Pooling* (LoP) and are related to the structure of the network. Based on these conditions, it was shown that GMS achieves maximum throughput in tree network graphs under k-hop interference (for any k) [16], [30], in $2 \times n$ switches [3], and in a number of interference graph classes [30].

The LoP conditions were recently generalized to provide

This work was partially supported by NSF grants DMS-0758364 and CNS-0916263, CIAN NSF ERC under grant EEC-0812072, FNR grant TR-PDR BFR08-17, and ONR grant N00014-01-1-0608.

[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).

[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 [4].

the σ-Local Pooling (σ-LoP) conditions under which GMS achieves σ% throughput [15], [16] (the conditions were reformulated in [19]). Using these conditions, lower bounds on the guaranteed throughput in geometric graphs [16] and in graphs under secondary interference constraints [18] were obtained.

From a practical point of view, identifying graphs that satisfy LoP and σ-LoP can provide important building blocks for partitioning a network (e.g., via channel allocation) into subnetworks in which GMS performs well [3]. Another possible application is to add artificial interference constraints to a graph that does not satisfy the LoP conditions in order to turn it into a LoP-satisfying graph. Adding such constraints may decrease the stability region but would enable GMS to achieve a large portion of the new stability region.

While it is known that under primary interference some graph families (mainly trees and $2 \times n$ bipartite graphs) satisfy LoP, the exact structure of networks that satisfy LoP was not characterized. In this paper, we use graph theoretic methods to *obtain the structure of* all *the network graphs that satisfy LoP under primary interference constraints* (in these networks GMS achieves 100% throughput). This allows us to develop an algorithm that checks if a network graph satisfies LoP in time linear in the number of vertices, significantly improving over any other known method. We note that although primary interference constraints may not hold in many wireless networking technologies, the characterization provides an important theoretical understanding regarding the performance of simple greedy algorithms. It also shows that the $2 \times n$ switch is the largest switch for which 100% throughput is guaranteed.

We then focus on graphs in which GMS does not achieve 100% throughout. We consider bipartite network graphs (i.e., input-queued switches) and show that *for bipartite graphs of size $k \times n$, where $k \leq 7$ and n is arbitrary, GMS achieves at least 66% throughput*. Namely, for switches with up to 7 inputs or 7 outputs, the throughput under GMS is bounded from below by 66%. This significantly improves upon the well known 50% lower bound [7] and confirms many simulation studies (e.g., [10]) in which it was shown that greedy algorithms perform relatively well in switches. To show that this result does not extend to all bipartite graphs, we show that there exists a 10×10 bipartite graph for which $\sigma = 0.6$.

Finally, we consider interference graphs[3] and categorize different graph families according to their σ values. In particular, we show that all co-strongly perfect graphs satisfy LoP. This class encapsulates all the classes of perfect LoP-satisfying interference graphs that were identified before (i.e., chordal graphs, interference graphs of trees, etc.). The observation increases the number of graphs known to satisfy LoP by an order of magnitude. Regarding σ-LoP we show that there are graphs with arbitrarily low σ. Since the worst case specific graph identified up to now had $\sigma = 0.6$ [15] and the lowest lower bound known for a graph family was 1/6 [16], [18], this provides an important insight regarding graphs in which GMS may have bad performance. We conclude with briefly describing a simulation study that compares the performance of GMS to the optimal algorithm in graphs with low σ.

[3]Although it has been recently shown that in some cases the interference 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.

To conclude, the main contributions of this paper are two-fold: (i) a characterization of *all* network graphs in which Local Pooling holds under primary interference constraints (in these network graphs Greedy Maximal Scheduling is guaranteed to achieve 100% throughput) and (ii) improved lower bounds on the throughput performance of Greedy Maximal Scheduling in small switches. Overall, the paper demonstrates that using graph theoretical techniques can significantly contribute to our understanding of greedy scheduling algorithms.

This paper is organized as follows. In Section II we present the model. We characterize all graphs that satisfy LoP under primary interference constraints in Section III. In Section IV we show that GMS achieves 66% throughput in switches with up to 7 inputs. We study the performance of GMS in interference graphs in Section V and we conclude and discuss open problems in Section VI.

II. Model and Preliminaries

In this section, we first present the network model under primary interference and then extend it for general interference. We also provide some graph theoretic definitions and derive results for graphs that exhibit certain symmetry.

A. Network Graphs

Consider a *network graph* $G = (V, E)$, where $V = \{1, \ldots, n\}$ is the set of nodes, and $E \subseteq \{ij : i, j \in V, i \neq j\}$ is a set of links indicating pairs of nodes between which data flow can occur. Following the model of [3], [8], [15], [27], assume that time is slotted and that packets are of equal size, each packet requiring one time slot of service across a link. The model considers only single-hop traffic. A queue is associated with each edge in the network. We assume that the stochastic arrivals to edge ij have long term rates λ_{ij} and are independent of each other. We denote by $\vec{\boldsymbol{\lambda}}$ the vector of the arrival rates λ_{ij} for every edge ij. For more details regarding the queue evolution process under this model, see [3], [8], [15].

For a graph G, let $\mathbf{M}(G)$ be a 0-1 matrix with $|E|$ rows, whose columns represent the maximal matchings of G. A *scheduling algorithm* selects a set of edges to activate at each time slot and transmits packets on those edges. Since they must not interfere under primary interference constraints, the selected edges form a matching. In other words, the scheduling algorithm picks a column $\boldsymbol{\pi}(t)$ from the maximal matching matrix $\mathbf{M}(G)$ at every time slot t. If $\boldsymbol{\pi}_k(t) = 1$, one of the two nodes along edge e_k can transmit, and the associated queue is decreased by one. We define the *stability region* (or *capacity region*) of a network as follows.

Definition 2.1 (Stability region [27]): The stability region of a network G is defined by

$$\Lambda^* = \left\{ \vec{\boldsymbol{\lambda}} \;|\vec{\boldsymbol{\lambda}} < \vec{u} \;\text{ for some } \vec{u} \in Co(\mathbf{M}(G)), \right\},$$

where $Co(\mathbf{M}(G))$ is the convex hull of the columns of $\mathbf{M}(G)$ (inequality operators are taken element-wise when their operands are vectors).

A *stable* scheduling algorithm (which we also refer to as a *throughput-optimal* algorithm or an algorithm that *achieves* 100% *throughput*) is defined as an algorithm for which the Markov chain that represents the evolution of the queues is positive recurrent for all arrivals $\vec{\boldsymbol{\lambda}} \in \Lambda^*$. It was shown in [27] that the Maximum Weight Matching algorithm that selects the matching with the largest total queue sizes at each slot is stable. When an algorithm is not throughput-optimal, the *efficiency ratio* γ^* indicates the fraction of the stability region for which the algorithm is stable (in simple words, the queues are bounded for all arrival rates $\vec{\boldsymbol{\lambda}} \in \gamma^*\Lambda^*$).

We briefly reproduce the definitions of Local Pooling (LoP) presented in [3], [8].[4] In the following, $\mathbf{e}$ denotes the vector having each entry equal to one.

Definition 2.2 (Subgraph Local Pooling - SLoP): A network graph G satisfies SLoP, if there exists $\boldsymbol{\alpha} \in [0,1]^{|E|}$ such that $\boldsymbol{\alpha}^T\mathbf{M}(G) = \mathbf{e}^T$.

This definition also corresponds to associating a weight, denoted $\boldsymbol{\alpha}(e)$, to all edges $e \in E$, such that

$$\sum_{e\in Z} \boldsymbol{\alpha}(e) = 1 \text{ for every maximal matching } Z \text{ in } G.$$

If a vector $\boldsymbol{\alpha}$ satisfies the above condition, we will say that it is a *good edge weighting*.

Definition 2.3 (Overall Local Pooling - OLoP): A network graph G satisfies OLoP, if every subgraph S of G satisfies SLoP.

In [8], Dimakis and Walrand proved that if a graph satisfies OLoP, GMS achieves 100% throughput. In networks in which OLoP is not satisfied, σ-Local Pooling [15], [16] provides a way of estimating the efficiency ratio γ^* of GMS. Below, we provide a different definition called σ-SLoP that is equivalent to the original one from [15], [16].

Definition 2.4 (σ-SLoP - Xi et. al. [19]): A network graph G satisfies σ-SLoP, if and only if there exists a vector $\boldsymbol{\alpha} \in [0,1]^{|E|}$ such that

$$\sigma\mathbf{e}^T \leq \boldsymbol{\alpha}^T\mathbf{M}(G) \leq \mathbf{e}^T.$$

Clearly, if a graph satisfies σ-SLoP, it also satisfies σ'-SLoP for every $\sigma' < \sigma$. Therefore, it is sufficient to focus on the largest value of σ such that G satisfies σ-SLoP. This value is denoted by $\sigma(G)$:

$$\sigma(G) := \max\{\sigma \mid G \text{ satisfies } \sigma\text{-SLoP}\}. \qquad (1)$$

This definition can also be written in terms of a Linear Program whose solution yields the $\sigma(G)$ for a given graph G [19]:

$$\sigma(G) = \max \sigma \quad \text{subject to } \sigma\mathbf{e}^T \leq \boldsymbol{\alpha}^T\mathbf{M}(G) \leq \mathbf{e}^T. \qquad (2)$$

[4]This definition slightly differs from that in [3] by setting the sum equal to $\mathbf{e}^T$ instead of $c\mathbf{e}^T$, where c is a positive constant.

We say that a graph satisfies σ-OLoP, if all of its subgraphs satisfy σ-SLoP. We can then define the local pooling factor of a graph as follows:

Definition 2.5 (Joo et. al. [15]): The local pooling factor $\sigma^*(G)$ of a network graph G is the largest value of σ for which σ-SLoP is satisfied for all subgraphs S.

This definition can also be written in terms of $\sigma(S)$:

$$\sigma^*(G) := \min\{\sigma(S) \mid \text{ for all subgraphs } S \text{ of } G\}. \qquad (3)$$

It was proved in [15] that the local pooling factor σ^* of a graph is equal to the efficiency ratio γ^* of GMS in that graph. For instance, if a graph has a local-pooling factor of $2/3$, GMS is stable for all arrival rates $\vec{\boldsymbol{\lambda}} \in \frac{2}{3}\Lambda^*$ and therefore achieves 66% throughput. Note that $\sigma^*(G) = 1$, if and only if G satisfies the OLoP condition.

B. Interference Graphs

We now generalize the model by introducing interference graphs. Based on the network graph and the interference constraints, the interference between network links can be modeled by an *interference graph* (or a *conflict graph*) $G_I = (V_I, E_I)$ [14]. We assign $V_I = E$. Thus, each edge e_k in the network graph is represented by a node v_k in the interference graph, and an edge v_iv_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). Under primary interference, the interference graph G_I corresponds to the *line graph* of G.

The model and the LoP theory described so far extend to interference graphs. The nodes of G_I correspond to queues to which packets arrive according to a stochastic process at every time slot. A scheduling algorithm must pick an independent set at each slot so that neighboring nodes will not be activated simultaneously. Each column of the matrix $\mathbf{M}(G_I)$ corresponds to a maximal independent set of G_I. An algorithm which selects the independent set with the largest weights (i.e., solves the Maximum Weight Independent Set Problem) is stable. SLoP corresponds to finding a vector $\boldsymbol{\alpha} \in [0,1]^{|V_I|}$ that assigns a weight $\boldsymbol{\alpha}(u)$ to each node u such that $\sum_{u\in I}\boldsymbol{\alpha}(u) = 1$ for every maximal independent set I in G_I. If such a vector exists, we will call it a *good node weighting*. For OLoP to be satisfied, SLoP must be satisfied by all *induced* subgraphs (i.e., with respect to node removals). σ-SLoP and σ-OLoP extend to this case in a very similar way.

C. Graph Theoretic Definitions

We briefly review some definitions from graph theory that are required in the following sections (for details, see [28]). For a graph G, we denote by $N(v)$ the set of neighbors of v and by $\deg(v) = |N(v)|$ the *degree of* v. For $x, y \in V(G)$, we say that x is *a clone of* y if $N(x) = N(y)$. We say that $X \subseteq V(G)$ is a *clique* (*independent set*) if the vertices in X are pairwise adjacent (non-adjacent). A matching M is said to *cover* a vertex v, if there exists an edge in M that is incident with v. For $x \in V(G)$, we denote by $G-x$ the graph obtained from G by deleting x and all edges incident with it. An *induced* *subgraph of* G is a subgraph of G that can be obtained from G by repeatedly deleting a vertex and all edges incident with it. For two graphs G_1, G_2, an *isomorphism from* G_1 *to* G_2 is a bijection $\phi : V(G_1) \to V(G_2)$ such that $uv \in E(G_1)$, if and only if $\phi(u)\phi(v) \in E(G_2)$. Two graphs G_1, G_2 are *isomorphic*, if there exists an isomorphism from G_1 to G_2. An *automorphism of* G is an isomorphism from G to itself. A graph G is *edge-transitive*, if for all $uv, wx \in E(G)$, there exists an automorphism ϕ of G such that $\phi(u)\phi(v) = wx$. A graph G is *vertex-transitive*, if for all $u, v \in V(G)$, there exists an automorphism ϕ of G such that $\phi(u) = v$. For $k \geq 1$, we say that G is k*-connected* if for every two distinct vertices in G, there exist k vertex-disjoint paths between them. A graph is connected if it is 1-connected. A *connected component of* G is a maximal connected induced subgraph of G. Finally, for $n \geq 1$, we let K_n denote the complete graph on n nodes and, for $t \geq 1$, we let $K_{t,n}$ denote the $t \times n$ complete bipartite graph.

D. $\sigma(G)$–values and Bounds on $\sigma(G)$

We now describe a simple method to compute a lower bounds on $\sigma(G)$ and provide a method for calculating $\sigma(G)$ of edge- and vertex-transitive graphs. These are graphs that exhibit a high degree of symmetry (e.g., cycles). We will use the following notation:

$$\nu(G) = \max\{|Z| : Z \text{ is a maximal matching in } G\},$$
$$\mu(G) = \min\{|Z| : Z \text{ is a maximal matching in } G\}.$$

The following lemma presents a lower-bound on $\sigma(G)$ [18].

Lemma 2.1 (Leconte et al. [18]): For any graph G, $\sigma(G) \geq \mu(G)/\nu(G)$.

Using Definition 2.4, we provide an alternative proof to this lemma in Appendix A. To demonstrate the benefits of the σ-OLoP definition, we provide a very simple proof to the fact that GMS achieves 50% throughput in any network graph G (shown in different methods in [7], [20]). First, note that the size of any maximal matching is at least half the size of a maximum matching [24], which means that $\mu(G) \geq \nu(G)/2$, for all G. By Lemma 2.1 and (3), it follows that $\sigma^*(G) \geq 1/2$ for every graph G, and therefore that $\gamma^* \geq 1/2$.

For edge-transitive graphs, the following lemma is a stronger counterpart of Lemma 2.1 (the proof is in Appendix A):

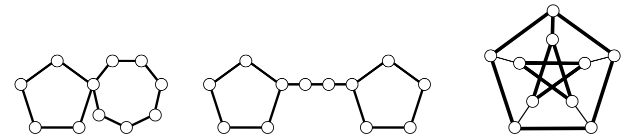
Lemma 2.2: If G is edge-transitive, then $\sigma(G) = \mu(G)/\nu(G)$.

The lemmas introduced previously also extend to general interference graphs. We define the independent set counterparts of $\mu(G)$ and $\nu(G)$ as follows:

$$\overline{\nu}(G) = \max\{|S| : Z \text{ is a maximal independent set in } G\},$$
$$\overline{\mu}(G) = \min\{|S| : Z \text{ is a maximal independent set in } G\}.$$

It is easy to generalize Lemma 2.1 to interference graphs G and show that $\sigma(G) \geq \overline{\mu}(G)/\overline{\nu}(G)$. For vertex-transitive graphs, the following is proved in Appendix A:

Lemma 2.3: If G is vertex-transitive, then $\sigma(G) = \overline{\mu}(G)/\overline{\nu}(G)$.



(a) $D_0^{5,7}$ (b) $D_3^{5,5}$ (c) Petersen graph

Fig. 1. Graphs (a) and (b): examples of graphs from the family $D_k^{p,q}$, all of which fail OLoP under primary interference. Graph (c): the Petersen graph. This graph does not satisfy OLoP because it contains, among other graphs, C_6 and $D_1^{5,5}$ (bold edges) as subgraphs.

III. Network graphs that satisfy OLoP under primary interference

Only a small collection of network graphs have been shown to satisfy OLoP under primary interference. Among the known cases are trees [3], [16], and $2 \times n$ bipartite graphs [3]. The main result of this section is a *description of the structure of all network graphs* that satisfy OLoP under primary interference. This structure shows that such graphs are relatively easy to construct and, moreover, they can be *recognized in linear time*. The proofs of the results can be found in Appendix B.

Define the following families of graphs. For $k \geq 3$, let C_k be a cycle with k edges (or, equivalently, k nodes). For $k \geq 0$ and $p, q \in \{5, 7\}$, let $D_k^{p,q}$ be the graph formed by the union of two cycles of size p and q joined by a k-edge path (where $k \geq 0$). If $k = 0$, the cycles share a common node (see Fig. 1-(a) and 1-(b)). Let $\mathcal{F} = \{C_k \mid k \geq 6, k \neq 7\} \cup \{D_k^{p,q} \mid k \geq 0; p, q \in \{5, 7\}\}$. For two graphs G and H, we say that G contains H as a subgraph if G has a subgraph that is isomorphic to H. We will say that a graph G is $\mathcal{F}$*-free*, if it does not contain any graph $F \in \mathcal{F}$ as a subgraph.

We will focus on connected graphs, because it is easy to see that a graph satisfies OLoP, if and only if all its connected components satisfy OLoP. So we may assume without loss of generality that all graphs in this section are connected graphs.

The results in this section are three-fold. First, in Subsection III-A, we give a structural description of all $\mathcal{F}$-free graphs. Second, in Subsection III-B, we will use this description to prove the following theorem:

Theorem 3.1: A network graph G satisfies OLoP under primary interference, if and only if G is $\mathcal{F}$-free.

Theorem 3.1 shows that if a network graph G does not satisfy OLoP under primary interference, then G contains some $F \in \mathcal{F}$ as a subgraph. For example, it was previously shown that the Petersen graph (Fig. 1-(c)) fails OLoP [15]. Using Theorem 3.1 we can immediately see this from the fact that it contains, for example, C_6 and $D_1^{5,5}$ as a subgraph.

Testing whether a network graph satisfies SLoP previously required enumerating all maximal matchings (of which there are an exponential number) and solving a Linear Program [8]. To test the OLoP condition, this procedure had to be repeated for every subgraph. The weakness of this approach is its large computational effort. In Subsection III-C, we present the third result, which uses the structure of $\mathcal{F}$-free graphs to construct an algorithm that decides in linear time whether a graph satisfies OLoP, as described in the following theorem:

Theorem 3.2: It can be decided in $O(|V(G)|)$ time whether a network graph G satisfies OLoP under primary interference.

A. The Structure of $\mathcal{F}$-free Graphs

We will start with a structural description of $\mathcal{F}$-free graphs. The reason for our interest in $\mathcal{F}$-free graphs is the fact (which will be proved in Subsection III-B) that the class of $\mathcal{F}$-free graphs is precisely the class of network graphs that satisfy OLoP under primary interference.

We will describe the structure of $\mathcal{F}$-free graphs in terms of the so-called 'block decomposition'. Let G be a connected graph. We call $x \in V(G)$ a *cut-node of* G, if $G - x$ is not connected. We call a maximal connected induced subgraph B of G such that B has no cut-node a *block of* G. Let $B_1, B_2, \ldots, B_q$ be the blocks of G. We call the collection $\{B_1, B_2, \ldots, B_q\}$ the *block decomposition of* G. It is known that the block decomposition is unique and that $E(B_1), E(B_2), \ldots, E(B_q)$ forms a partition of $E(G)$ (e.g., [28]). Furthermore, the node sets of every two blocks intersect in at most one node and this node is a cut-node of G.

Block decompositions give a tree-like decomposition of a graph in the following sense. Construct the *block-cutpoint graph of* G by keeping the cut-nodes of G and replacing each block B_i of G by a node b_i. Make each cut-node v adjacent to b_i if and only if $v \in V(B_i)$. It is known that the block-cutpoint graph of G forms a tree (e.g., [28]). With this tree-like structure in mind, we say that a block B_i is a *leaf block* if it contains at most one cut-node of G. Clearly, if $q \geq 2$, then $\{B_i\}_{i=1}^{q}$ contains at least two leaf blocks.

It turns out that the block decomposition of an $\mathcal{F}$-free graph is relatively simple in the sense that there are only two types of blocks. The types are defined by the following two families of graphs. Examples of these families appear in Fig. 2.

$\mathcal{B}_1$: Construct $\mathcal{B}_1$ as follows. Let H be a graph with $V(H) = \{c_1, c_2, \ldots, c_k\}$, with $k \in \{5, 7\}$, such that

1) c_1-c_2-$\cdots$-c_k-c_1 is a cycle;
2) if $k = 5$, then the other adjacencies are arbitrary; if $k = 7$, then all other pairs are non-adjacent, except possibly $\{c_1, c_4\}$, $\{c_1, c_5\}$ and $\{c_4, c_7\}$.

Then, $H \in \mathcal{B}_1$.

Now iteratively perform the following operation. Let $H' \in \mathcal{B}_1$ and let $x \in V(H')$ with $\deg(x) = 2$. Construct H'' from H' by adding a node x' such that $N(x') = N(x)$. Then, $H'' \in \mathcal{B}_1$. We say that a graph is *of the* $\mathcal{B}_1$ *type* if it is isomorphic to a graph in $\mathcal{B}_1$.

$\mathcal{B}_2$: Let $\mathcal{B}_2 = \{K_2, K_3, K_4\} \cup \{K_{2,t}, K_{2,t}^{+} \mid t \geq 2\}$, where $K_{2,t}^{+}$ is constructed from $K_{2,t}$ by adding an edge between the two nodes on the side that has cardinality 2. We say that a graph is *of the* $\mathcal{B}_2$ *type*, if it is isomorphic to a graph in $\mathcal{B}_2$.

In simple words, graphs of the $\mathcal{B}_1$ type are constructed by starting with a cycle of length five or seven. Then we may add some additional edges between nodes of the cycle, subject to some constraints. Finally, we may iteratively take a node x of degree 2 and add a clone x' of x. It will turn out that $\mathcal{F}$-free graphs have at most one block of the $\mathcal{B}_1$ type and that all other blocks are of the $\mathcal{B}_2$ type. This means that $\mathcal{F}$-free graphs can be constructed by starting with a block that is either of the $\mathcal{B}_1$ or of the $\mathcal{B}_2$ type, and then iteratively adding a block of the $\mathcal{B}_2$ type by 'glueing' it on an arbitrary node.

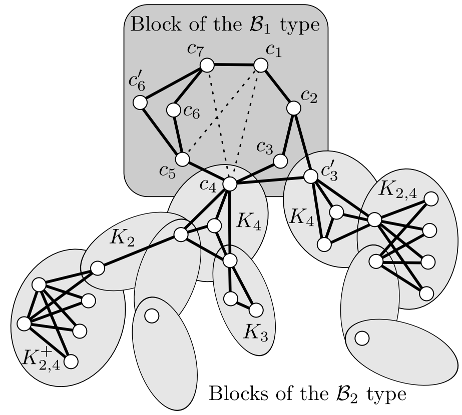


Fig. 2. An example of an $\mathcal{F}$-free graph (the dashed edges may or may not be present). The ellipses show the blocks of the graph.

Fig. 2 shows an example of an $\mathcal{F}$-free graph. The tree-like structure is clearly visible. The graph has one block of the $\mathcal{B}_1$ type with $k = 7$. This block consists of a cycle of length 7 together with two clones. The other blocks are of the $\mathcal{B}_2$ type. Some of them are attached to the block of the $\mathcal{B}_1$ type through a cut-node. Others are attached to other blocks of the $\mathcal{B}_1$ type. Notice that trees and $2 \times n$ complete bipartite graphs, which were previously known to satisfy OLoP [3], [16], are, as should be expected, subsumed by this structure.

The goal of this subsection is to prove the following formal version of the characterization given above:

Theorem 3.3: Let G be a connected graph and let $\{B_1, B_2, \ldots, B_q\}$ be the block decomposition of G. Then G is $\mathcal{F}$-free, if and only if there is at most one block that is of the $\mathcal{B}_1$ type and all other blocks are of the $\mathcal{B}_2$ type.

The proof of the 'if' direction is straightforward. Here, we will give a proof sketch of the 'only-if' direction in a number of steps. For a block B in an $\mathcal{F}$-free graph, its type depends on the size of the longest cycle in B. It will turn out that if B contains a cycle of length 5 or 7, then B is of the $\mathcal{B}_1$ type. Otherwise, B is of the $\mathcal{B}_2$ type. We have the following result on blocks that have a cycle of length five or seven.

Lemma 3.1: Let G be an $\mathcal{F}$-free graph and let B be a block of G. Let F be a cycle in B that has maximum length. If $|V(F)| \geq 5$, then B is of the $\mathcal{B}_1$ type.

Next, we deal with blocks that do not contain a cycle of length 5 or 7. It follows from the definition of $\mathcal{F}$-free graphs that such blocks do not have cycles of length at least 5. Maffray [21] proved the following theorem:

Theorem 3.4: [Maffray [21]] Let G be a graph. Then, the following statements are equivalent:

(1) G does not contain any odd cycle of length at least 5.
(2) For every connected subgraph G' of G, either G' is isomorphic to K_4, or G' is a bipartite graph, or G' is

isomorphic to $K_{2,t}^{+}$ for some $t \geq 1$, or G' has a cut-node.

Theorem 3.4 implies the following lemma.

Lemma 3.2: Let G be an $\mathcal{F}$-free graph and let B be a block of G. Suppose that B contains no cycle of length at least 5. Then, B is of the $\mathcal{B}_2$ type.

We are now ready to prove Theorem 3.3:

Proof of Theorem 3.3: Let G be an $\mathcal{F}$-free graph and let $\{B_1, B_2, \dots, B_m\}$ be the block decomposition of G. For every $i \in \{1, 2, \dots, m\}$, if B_i contains a cycle of length 5 or 7, it follows from Lemma 3.1 that B_i is of the $\mathcal{B}_1$ type. Otherwise, it follows from from Lemma 3.2 that B_i is of the $\mathcal{B}_2$ type. Now suppose that there are $i \neq j$ and $p, q \in \{5, 7\}$ such that B_i contains a cycle T_1 of length p and B_j contains a cycle T_2 of length q. Since G is connected, there exists a path P of length $k \geq 0$ from a node in T_1 to a node in T_2. Since T_1 and T_2 are subgraphs of different blocks, T_1 and T_2 share at most one node. If they share a node, then $k = 0$. Now the edges of T_1, T_2, P form a graph isomorphic to $D_k^{p,q}$, a contradiction. This proves Theorem 3.3. ∎

B. Network graphs satisfy OLoP under primary interference, if and only if they are $\mathcal{F}$-free

Now that we have described the structure of all $\mathcal{F}$-free graphs, we use this structure to prove Theorem 3.1 which states that a network graph satisfies OLoP under primary interference, if and only if it is $\mathcal{F}$-free. It was shown in [3] (Theorems 2 and 3) that all cycles of length $k \geq 6, k \neq 7$ fail SLoP.[5] Therefore, such cycles do not appear as subgraphs in graphs that satisfy OLoP. The following lemma shows that the same is true for the graphs $D_k^{p,q}$.

Lemma 3.3: $D_k^{p,q}$ fails SLoP for all $p, q \in \{5, 7\}$, $k \geq 0$.

The results from [3] together with Lemma 3.3 imply the following result:

Corollary 3.1: Graphs that satisfy OLoP are $\mathcal{F}$-free.

Proof: Let G be a graph that satisfies OLoP. By the definition of OLoP, every subgraph H of G satisfies SLoP. Since every graph in $\mathcal{F}$ fails SLoP, it follows that G does not contain any graph in $\mathcal{F}$ as a subgraph. ∎

Corollary 3.1 settles the ‘only-if’ direction of Theorem 3.1. To prove the ‘if’ direction, we will start with a useful lemma. We will give the idea of the proof, the full proof being in Appendix B.

Lemma 3.4: Let G be a graph and $x, x' \in V(G)$ such that $\deg(x) = 2$ and x' is a clone of x. Then, G satisfies SLoP.

Proof idea: Let $\{z_1, z_2\} = N(x)$. Define $\boldsymbol{\alpha} \in [0,1]^{|E|}$ by letting $\boldsymbol{\alpha}(e) = 1/2$, if e is incident with z_1 or z_2, and $e \neq z_1z_2$, $\boldsymbol{\alpha}(z_1z_2) = 1$, if $z_1z_2 \in E(G)$, and $\boldsymbol{\alpha}(e) = 0$ for all other edges e. Now every maximal matching uses either two edges e_1, e_2 with $\boldsymbol{\alpha}(e_1) = \boldsymbol{\alpha}(e_2) = 1/2$, or edge z_1z_2. ∎

The following lemma is the crucial step in settling the ‘only-if’ direction of Theorem 3.1. Again, we give the proof idea.

Lemma 3.5: Every connected $\mathcal{F}$-free satisfies SLoP.

[5]Although the case considered in [3] pertains to interference graphs, the network case is identical since the interference graph (under primary interference) of a cycle is a cycle of the same length.

Proof idea: Let G be a connected $\mathcal{F}$-free graph and let $\{B_1, B_2, \dots, B_q\}$ be the block decomposition of G. It follows from Theorem 3.3 that there is at most one block B_i that is of the $\mathcal{B}_1$ type and all other blocks are of the $\mathcal{B}_2$ type. We will construct a good edge weighting $\boldsymbol{\alpha}$ for G.

Suppose first that G has a leaf block B_i of the $\mathcal{B}_2$ type. If $q = 2$, then let x be the cut-node of G in $V(B_i)$. If $q = 1$, let $x \in V(B_i)$ be arbitrary. There are four cases:

(1) B_i is isomorphic to K_2: let x, v denote the nodes of B_i. Let $\boldsymbol{\alpha}(e) = 1$ for all edges incident with x and $\boldsymbol{\alpha}(e) = 0$ for every other edge e.

(2) B_i is isomorphic to K_3: let $\boldsymbol{\alpha}(e) = 1$ for all $e \in E(B_i)$ and $\boldsymbol{\alpha}(e) = 0$ for every other edge e.

(3) B_i is isomorphic to K_4: let x, v_1, v_2, v_3 denote the nodes of B_i and let $\boldsymbol{\alpha}(v_1v_2) = \boldsymbol{\alpha}(v_1v_3) = \boldsymbol{\alpha}(v_2v_3) = 1$ and $\boldsymbol{\alpha}(e) = 0$ for all $e \in (E(G) \setminus \{v_1v_2, v_1v_3, v_2v_3\})$.

(4) B_i is isomorphic to $K_{2,t}$ or $K_{2,t}^{+}$ for some $t \geq 2$: let $V(B_i) = V_1 \cup V_2$ such that $|V_1| = 2$ and V_2 is an independent set. Let $V_1 = \{y_1, y_2\}$ and let $V_2 = \{z_1, z_2, ..., z_t\}$. If B_i is isomorphic to $K_{2,2}^{+}$ and $x \in V_2$, then assume that $x = z_1$ and set $\boldsymbol{\alpha}(y_1z_2) = \boldsymbol{\alpha}(y_2z_2) = \boldsymbol{\alpha}(y_1y_2) = 1$. Otherwise, B_i contains nodes p, p' such that $\deg(p) = \deg(p') = 2$ and p' is a clone of p, and hence, the result follows from Lemma 3.4.

Thus, we may assume that G does not have a leaf block of the $\mathcal{B}_2$ type. Since if $q \geq 2$, G has at least two leaf blocks, and hence, at least one leaf block of the $\mathcal{B}_2$ type, we may assume that $q = 1$ and $G = B_1$ is of the $\mathcal{B}_1$ type. Let H, k be as in the definition of $\mathcal{B}_1$. It follows from the definition of H that $|V(H)| = k$. First, suppose that $V(G) \setminus V(H) \neq \emptyset$. Then, it follows from the definition of $\mathcal{B}_1$ that there exist two nodes x, x' such that $\deg(x) = \deg(x') = 2$ and x' is a clone of x. It follows from Lemma 3.4 that G satisfies SLoP. So we may assume that $V(G) = V(H)$. If $k = 5$, then every maximal matching has size two, and hence, we may set $\boldsymbol{\alpha}(e) = 1/2$ for all $e \in E(G)$. If $k = 7$, then every maximal matching has size three, and hence, we may set $\boldsymbol{\alpha}(e) = 1/3$ for all $e \in E(G)$. ∎

We are now in a position to prove Theorem 3.1:

Proof of Theorem 3.1: Corollary 3.1 is the ‘only-if’ part of the theorem. For the ‘if’ part, since every subgraph of G is $\mathcal{F}$-free, it follows from Lemma 3.5 that every subgraph of G satisfies SLoP. Therefore, G satisfies OLoP. ∎

C. Recognizing network graphs that satisfy OLoP under primary interference

Having described the structure of graphs that satisfy OLoP, we provide an efficient algorithm for testing whether a given network graph satisfies OLoP under primary interference. A useful observation is the following (see Appendix B for the proof).

Lemma 3.6: $|E(G)| \leq 2|V(G)|$ for every $\mathcal{F}$-free graph G.

This puts us in a position to prove Theorem 3.2.

Proof idea of Theorem 3.2: We may assume that G is connected. By Theorems 3.1 and 3.3, it suffices to check whether G has the structure described in Theorem 3.3. We propose the following algorithm. Let $n = |V(G)|$ and $m =$

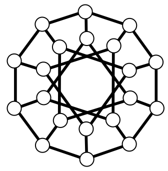


Fig. 3. The Desargues graph $\mathcal{D}$ for which $\sigma(\mathcal{D}) = 0.6$ and which is a subgraph of $K_{10,10}$, showing that $\sigma^*(K_{10,10}) \leq 0.6$.

$|E(G)|$. First, check that $m \leq 2n$, because otherwise G is not $\mathcal{F}$-free by Lemma 3.6 and we can stop. Now, construct the block decomposition $\{B_1, B_2, ..., B_q\}$ of G. Since $m \leq 2n$, this can be done in $O(n+m) = O(n)$ time (see e.g., [11]). For each block B_i, test in $O(|V(B_i)|)$ time whether B_i is of the $\mathcal{B}_2$ type. If G has more than one block that is not of the $\mathcal{B}_2$ type, then G is not $\mathcal{F}$-free and we stop. If we encounter no such block, then G is $\mathcal{F}$-free and we stop. Next, check whether B^* is of the $\mathcal{B}_1$ type using multiple applications of Bodlaender's algorithm [2] which, for fixed k, finds a cycle of length at least k in a given graph H, if it exists, in $O(k!2^k|V(H)|)$ time. Checking this can be done in $O(|V(B)|)$ time. Therefore, the overall complexity of the algorithm is $O(n)$. ∎

IV. $t \times n$ SWITCHES WITH $t \leq 7$ SATISFY $\sigma^* \geq 2/3$

In the previous section, we characterized the full set of graphs that satisfy OLoP. It is only natural to ask the question: what happens to graphs that do not satisfy OLoP? In this section, we will show that every bipartite graph G that has one side with at most 7 nodes satisfies $\sigma^*(G) \geq 2/3$, which implies that $\sigma^*(K_{t,n}) = 2/3$ for $3 \leq t \leq 7, n \geq 3$. We also note that this bound is close to being tight by presenting a bipartite graph with 10 nodes on one side for which $\sigma^*(G) < 2/3$. Consider the so-called Desargues graph $\mathcal{D}$ in Fig. 3. $\mathcal{D}$ is edge-transitive and hence it follows from Lemma 2.2 and the fact that $\nu(\mathcal{D}) = 10$ and $\mu(\mathcal{D}) = 6$ that $\sigma(\mathcal{D}) = 3/5$. Since $\mathcal{D}$ is a subgraph of $K_{10,10}$, this implies that $\sigma^*(K_{t,n}) \leq 3/5$ for all $t \geq 10, n \geq 10$. The proofs of the results can be found in Appendix C.

We now concentrate on subgraphs of $K_{t,n}$ with $t \leq 7, n \geq 1$. We will start with some easy observations that help give a lower bound on $\sigma(G)$.

Lemma 4.1: Let G be a graph.

(a) If there exists $v \in V(G)$ such that every maximal matching in G covers v, then $\sigma(G) = 1$.
(b) If $\deg(v) = 1$ for some $v \in V(G)$, then $\sigma(G) = 1$.
(c) If $\deg(v) = 2$ for some $v \in V(G)$, then $\sigma(G) \geq 2/3$.

Proof: Part (a): let $\boldsymbol{\alpha}(e) = 1$ for all edges incident with v and $\boldsymbol{\alpha}(e) = 0$ for all other edges. Clearly, every maximal matching Z satisfies $\sum_{e \in Z} \boldsymbol{\alpha}(e) = 1$. This proves (a). Part (b) follows immediately because if $\deg(v) = 1$, then every maximal matching covers the unique neighbor u of v. Part (c): let a, b be the neighbors of v. Let $\boldsymbol{\alpha}(av) = \boldsymbol{\alpha}(bv) = 2/3$, $\boldsymbol{\alpha}(ab) = 1$ if $ab \in E(G)$, $\boldsymbol{\alpha}(e) = 1/3$ for all edges $e \notin \{ab, bv, av\}$ that are incident with a or b, and $\boldsymbol{\alpha}(e) = 0$ for all other edges. It is not hard to see that $\sum_{e \in Z} \boldsymbol{\alpha}(e) \geq 2/3$ for every maximal matching Z in G. This proves (c), thus proving Lemma 4.1. ∎

By using the conditions given in Lemmas 4.1 and 2.1, we prove the following lemma in Appendix C:

Lemma 4.2: Let G be a bipartite graph with $\mu(G) \leq 4$. Then $\sigma(G) \geq 2/3$.

Lemma 4.2 has the following corollaries:

Corollary 4.1: Every bipartite graph G with $\nu(G) \leq 7$ satisfies $\sigma^*(G) \geq 2/3$.

Proof: Let H be a subgraph of G (perhaps $H = G$). Clearly, $\nu(H) \leq \nu(G) \leq 7$. If $\mu(H) \leq 4$, then it follows from Lemma 4.2 that $\sigma(H) \geq 2/3$. Otherwise, $\mu(H) \geq 5$ and hence it follows from Lemma 2.1 that $\sigma(H) \geq 5/7 > 2/3$. Therefore, $\sigma(H) \geq 2/3$ for all subgraphs H of G. It follows that $\sigma^*(G) \geq 2/3$. ∎

It is already known that $\sigma^*(K_{t,n}) = 1$ for $t \in \{1, 2\}$. For $3 \leq t \leq 7$, we obtain:

Corollary 4.2: $\sigma^*(K_{t,n}) = 2/3$ for all $3 \leq t \leq 7$, $n \geq 3$.

Proof: Let $3 \leq t \leq 7$, $n \geq 3$. It follows from Corollary 4.1 that $\sigma^*(K_{t,n}) \geq 2/3$. Since $K_{t,n}$ contains C_6 as a subgraph and $\sigma(C_6) = 2/3$, it follows that $\sigma^*(K_{t,n}) = 2/3$. ∎

V. INTERFERENCE GRAPHS AND THEIR σ^*–VALUES

Our focus so far has been on network graphs and primary interference constraints. We now consider general interference graphs that represent various transmission constraints. Recall that under general interference constraints, a scheduling algorithm has to select an independent set from the interference graph at each slot. We are interested in the performance of a low-complexity GMS algorithm which greedily picks the nodes with the largest weight (this algorithm is also referred to as the Maximal Weighted Independent Set algorithm). The results are summarized in Fig. 4 which illustrates throughput guarantees of several graph families.

A. OLoP-satisfying Interference Graphs

We first show that the OLoP condition holds in a large subclass of perfect graphs which we will call co-strongly perfect graphs:

Definition 5.1 (Co-strongly perfect graph): A graph G is co-strongly perfect if for every induced subgraph H of G there exists $\boldsymbol{\alpha} = \boldsymbol{\alpha}(H) \in \{0,1\}^{|V(H)|}$ such that $\boldsymbol{\alpha}^T \mathbf{M}(H) = \mathbf{e}^T$. Equivalently, a graph G is co-strongly perfect, if and only if G contains a clique that intersects every maximal independent set in G. It follows from the definition, and from the interference graph counterparts of Definitions 2.2 and 2.3 that *every graph that is co-strongly perfect satisfies OLoP*.

Note from the above weighting that co-strongly perfect graphs satisfy OLoP with an integer vector $\boldsymbol{\alpha}$. An open question is whether all perfect graphs that satisfy OLoP do so with integer weights $\boldsymbol{\alpha}$. This is not true for imperfect graphs, because C_5 is an imperfect graph that satisfies OLoP with the unique optimal node weighting $\boldsymbol{\alpha}(v) = 1/2$ for all $v \in V(C_5)$. The vertical division of Fig. 4 into perfect and non-perfect

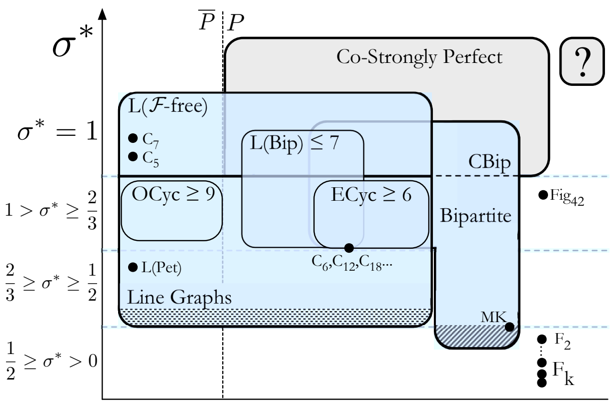


Fig. 4. Throughput guarantees (bounds on σ^*) for various interference graph families: P - Perfect graphs, $\overline{P}$ - Non-perfect graphs, ECyc ≥ 6 - cycles C_n with n even and $n \geq 6$, OCyc ≥ 9 - cycles C_n with n odd and $n \geq 9$, L(Bip) ≤ 7 - line graphs of $k \times n$ bipartite graphs with $k \leq 7$, CBip - Chordal bipartite graphs, L(Pet) - Line graph of the Petersen graph, L($\mathcal{F}$-free) - Line graphs of $\mathcal{F}$-free graphs, Fig$_{42}$ - Graph from Fig. 42 in [13], MK - Möbius-Kantor graph, F_2 - graph obtained by a single cycle substitution, F_k - sequence of graphs obtained by recursive cycle substitutions.

graphs, denoted P and $\overline{P}$, respectively, allows us to represent this open problem by the question mark in the perfect division.

The `Co-Strongly Perfect` class includes a large number of perfect graph families (some of them identified individually in [30]). To provide some context about the magnitude of the result, consider the set of simple graphs with 10 nodes. There are 3,063,185 such co-strongly perfect graphs. This can be compared to the 126,768 chordal graphs with 10 nodes (the chordal graphs family is one of the largest previously known families satisfying OLoP) and to the 106 trees [26].

We proved in Section III-A that $\mathcal{F}$-free network graphs are OLoP-satisfying under primary interference. This is shown in Fig. 4 by the class `L(F-free)` (line graphs of $\mathcal{F}$-free graphs), which is a subclass of the `Line Graphs` family. Since L($\mathcal{F}$-free) graphs represent all OLoP-satisfying line graphs, this family covers the entire section of `Line Graphs` that is in the $\sigma^* = 1$ division. The chordal bipartite family, denoted `CBip` on Fig. 4, is another family that is entirely OLoP-satisfying and forms the subclass of `Bipartite` graphs that are co-strongly perfect and OLoP-satisfying [30].

B. σ^*–values for Line Graphs

We examine the σ^* values of interference graphs that are `Line Graphs` and that fail OLoP. As mentioned in Section II and in [20], $\sigma^* \geq 1/2$ for all `Line Graphs`. In Fig. 4, the bottom part of this family is shaded to indicate that we still do not have any specific example of a line graph for which $\sigma^* = 1/2$. The line graph with the lowest known σ^* value ($\sigma^* = 0.6$) is the line graph of the Petersen graph (Fig. 1-(c)) [15], denoted `L(Pet)`.

We consider families that are subclasses of line graphs. The results on bipartite network graphs from Section IV (line graphs of subgraphs of $K_{t,n}$ with $t \leq 7$ have $\sigma^* \geq 2/3$) are shown on the figure as the `L(Bip) ≤ 7` class which is located in the top and the second divisions.

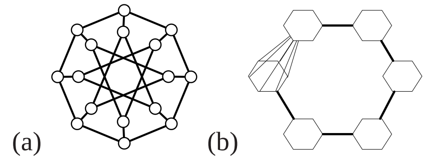


Fig. 5. Graphs that have low σ^* values: (a) Möbius-Kantor graph (b) F_2, a graph where each node of a C_6 is substituted by a C_6.

We now obtain the σ^* values of the entire family of cycles, some of which have been considered individually in the literature. For the 6-cycle it has been shown that $\sigma^* = 2/3$ [8], [15] (represented by the point `C6` on Fig. 4). It has also been shown that `C5` and `C7` satisfy OLoP, while larger cycles ($n \geq 8$) do not [30]. Using Lemma 2.3, the following lemma provides the σ^* of all cycles.

Lemma 5.1: For $n \geq 3$, $\sigma^*(C_n) = \lceil n/3 \rceil / \lfloor n/2 \rfloor$.

Proof: Let $n \geq 3$. Since every proper induced subgraph H of C_n (i.e. $H \neq C_n$) is a forest, we have $\sigma(H) = 1$ for every such H. Now consider C_n itself. A maximum independent set in C_n can be constructed by choosing nodes alternatingly on the cycle. This implies that $\overline{\nu}(G) = \lfloor n/2 \rfloor$. A smallest maximal independent set can be constructed by choosing nodes skipping two nodes at a time. This implies that $\overline{\mu}(G) = \lceil n/3 \rceil$. Since C_n is vertex-transitive, it follows from Lemma 2.3 that $\sigma(G) = \lceil n/3 \rceil / \lfloor n/2 \rfloor$. From this and the above, the result follows from the definition of $\sigma^*(G)$. ∎

To the best of our knowledge, this is the first time an entire family's σ–value has been characterized this precisely. This result is shown in Fig. 4 as the classes `ECyc ≥ 6` and `OCyc ≥ 9`, for large even and odd cycles, respectively. No odd cycle can have $\sigma^* = 2/3$, which is why the `OCyc` family is strictly within the second division. The family of even cycles is exactly the intersection of the `Bipartite` and the `Line Graphs` families that do not satisfy OLoP. In other words, there are no bipartite line graphs that have $\sigma^* < 1$ and that are not large even cycles. Since $\lceil n/3 \rceil / \lfloor n/2 \rfloor \geq 2/3$ for all $n \geq 3$, Lemma 5.1 provides a lower-bound of $2/3$ for arbitrary cycles, resulting in the following corollary:

Corollary 5.1: For all cycles, $\sigma^*(C_n) \geq 2/3$.

C. Low σ^*–values

We now focus on graphs with very low σ^*. The current knowledge of σ^*–values is limited to a handful graphs in which GMS achieves a large portion of the stability region. The lowest σ^*–value of a specific graph is $\sigma^* = 0.6$ for the line graph of the Petersen graph [15]. In [16], it was shown that for geometric graphs $1/6 \leq \sigma^* \leq 1/3$. Below, we present a graph that has $\sigma^* = 0.5$ and provide a method through which it is possible to create networks with arbitrarily low σ^*.

Consider the graph shown in Fig. 5-(a). It is a generalized Petersen graph with factors $GP(8, 3)$, also known as the Möbius-Kantor graph $\mathcal{MK}$. Because of its vertex-transitivity, it follows from Lemma 2.3 and from the fact that $\bar{\nu}(\mathcal{MK}) = 8$ and $\bar{\mu}(\mathcal{MK}) = 4$ that $\sigma^*(\mathcal{MK}) = 1/2$. Hence, GMS can only guarantee 50% throughput.[6] Being a bipartite graph, the Möbius-Kantor implies that `Bipartite` graphs can have σ^* values as low as 0.5, as illustrated in Fig. 4. Whether bipartite graphs can have $\sigma^* < 0.5$ is still an open question, shown by the shaded region in Fig. 4.

Now consider the following family. Let F_1 be a 6-cycle and, for $k \geq 2$, construct F_k from F_{k-1} by substituting a 6-cycle for each node $v \in V(F_k)$. By *substituting* C_6 for a node x of the original graph, we mean that we replace x by a 6-cycle H and we make every $v \in V(H)$ adjacent to every neighbor of x. For example, F_2 is shown in Fig. 5-(b), (where the hexagons represent 6-cycles). Using Lemma 2.3 and the fact that the F_k are vertex-transitive, we prove the following in Appendix D:

Observation 5.1: $\sigma^*(F_k) \leq (2/3)^k$ for all $k \geq 1$.

Since we may choose k arbitrarily large, it follows that there exist graphs with arbitrarily small σ^*. A graph generated by this method appears in Fig. 4 as `F2` and the sequence of graphs obtained through recursive substitution with decreasing σ^*–values is shown as `Fk`.

Finally, it can be shown that the family of weakly chordal graphs that was left unresolved in [30] is not entirely OLoP-satisfying. An example of a weakly chordal graph that is not co-strongly perfect and that has $\sigma^* < 1$ appears in Fig. 42 in [13] and is denoted in Fig. 4 as `Fig42`.

D. Simulation Results

When GMS guarantees only low throughput efficiency γ^*, there may exist a specific arrival rate outside of $\gamma^*\Lambda^*$ for which GMS is not stable. In real-life arrival processes, it is sometimes unlikely that such an arrival process would occur. Hence, GMS may behave better than predicted. We used Matlab simulations in order to evaluate the performance of GMS in graphs with low σ^* identified in Section V-B.

We consider i.i.d. uniform arrivals to every node at each time slot for a range of normalized loads within the stability region. We tested the GMS and the stable algorithm that solves the Maximum Weight Independent Set problem[7]. For each arrival rate, the simulation was run for 10,000 iterations. For each graph and arrival rate value, the average queue lengths appear in Fig. 6. The cycle C_{12} has $\sigma^* = 2/3$. In the figure, we see that in a cycle, the queues under GMS become unstable at around load level of 0.85. Although the Möbius-Kantor graph has a $\sigma^* = 1/2$, GMS performs similarly.

VI. Conclusion

The Local Pooling (LoP) conditions provide a new tool for better understanding the performance of Greedy Maximal Scheduling (GMS) algorithms. In this paper, we identified *all* the network graphs in which these conditions hold under primary interference constraints (in these graphs Greedy Maximal Scheduling achieves 100% throughput). In addition, we

[6]Note that since this graph contains a claw (i.e., a complete bipartite graph $K_{1,3}$), it cannot be the interference graph of any network under primary interference constraints.

[7]Although the problem is NP-complete, we obtained numerical solutions in small graphs.

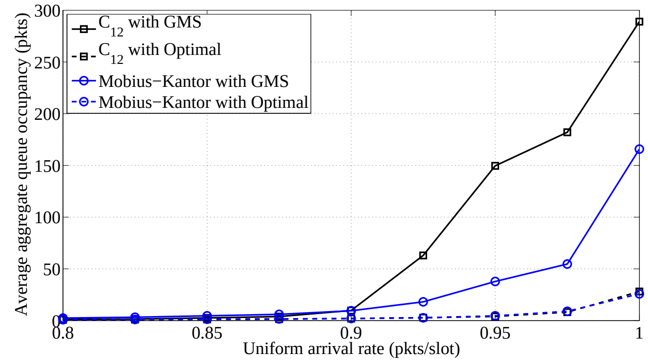


Fig. 6. Average queue sizes as a function of the arrival rate under GMS and the optimal algorithm. The results obtained via simulation in a 12-cycle and a Möbius-Kantor graph.

showed that in all bipartite graphs of size up to $7 \times n$, GMS is guaranteed to achieve 66% throughput. Finally, we studied the performance of GMS in interference graphs and showed that σ^* can be arbitrarily low.

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. As such, the paper demonstrates that using graph theoretical methods can significantly contribute to our understanding of greedy scheduling algorithms. From a graph theoretical point of view, LoP raises many interesting open problems. For example, three of the authors [5], [6] are currently working on extending some of the results to claw-free graphs, which are a generalization of the interference graphs of networks under primary interference. From the networking point of view, there remain many open problems. For example, generalizing the interference model to a model based on SINR and deriving the corresponding LoP conditions remain major subjects for future research.

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] H. Bodlaender, "On linear time minor tests with depth-first search," *J. Algorithms*, vol. 14, no. 1, pp. 1–23, 1993.

[3] 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.

[4] 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, pp. 572–594, Feb. 2008.

[5] M. Chudnovsky, B. Ries, and Y. Zwols, "Claw-free graphs with strongly perfect complements. Fractional and integral version. Part I: Graphs with $\alpha \leq 3$ and fuzzy long circular interval graphs." 2009, in preparation.

[6] ——, "Claw-free graphs with strongly perfect complements. Fractional and integral version. Part II: Graphs with $\alpha \geq 4$ that are not fuzzy long circular interval graphs." 2009, in preparation.

[7] J. G. Dai and B. Prabhakar, "The throughput of data switches with and without speedup," in *Proc. IEEE INFOCOM'00*, Mar. 2000.

[8] 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.

[9] L. Georgiadis, M. J. Neely, and L. Tassiulas, *Resource Allocation and Cross-Layer Control in Wireless Networks*. NOW Publishers, 2006.

[10] P. Giaccone, B. Prabhakar, and D. Shah, "Randomized scheduling algorithms for high-aggregate bandwidth switches," *IEEE J. Select. Areas Commun.*, vol. 21, no. 4, pp. 546–559, May 2003.

[11] J. Gross and J. Yellen, *Graph theory and its applications*. CRC press, 2006.

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

[13] S. Hougardy, "Classes of perfect graphs," *Discrete Math.*, vol. 306, no. 19-20, pp. 2529–2571, 2006.

[14] 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.

[15] 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.

[16] C. Joo, X. Lin, and N. Shroff, "Understanding the capacity region of the greedy maximal scheduling algorithm in multi-hop wireless networks," in *Proc. IEEE INFOCOM'08*, Apr. 2008.

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

[18] M. Leconte, J. Ni, and R. Srikant, "Improved bounds on the throughput efficiency of greedy maximal scheduling in wireless networks," in *Proc. ACM MOBIHOC'09*, May 2009.

[19] B. Li, C. Boyaci, and Y. Xia, "A refined performance characterization of longest-queue-first policy in wireless networks," in *Proc. ACM MOBIHOC'09*, May 2009.

[20] 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.

[21] F. Maffray, "Kernels in perfect line-graphs," *J. Combin. Theory Ser. B*, vol. 55, no. 1, pp. 1–8, 1992.

[22] N. McKeown, A. Mekkittikul, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," *IEEE Trans. Commun.*, vol. 47, no. 8, pp. 1260–1267, Aug. 1999.

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

[24] R. Preis, "Linear time 1/2-approximation algorithm for maximum weighted matching in general graphs," in *LNCS*, vol. 1563, pp. 259–269, 1999.

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

[26] N. J. A. Sloane. (2009) The on-line encyclopedia of integer sequences. [Online]. Available: http://www.research.att.com/∼njas/sequences/

[27] 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.

[28] D. West, *Introduction to Graph Theory*. Prentice Hall, Upper Saddle River, NJ, 2001.

[29] X. Wu and R. Srikant, "Bounds on the capacity region of multi-hop wireless networks under distributed greedy scheduling," in *Proc. IEEE INFOCOM'06*, Apr. 2006.

[30] G. Zussman, A. Brzezinski, and E. Modiano, "Multihop local pooling for distributed throughput maximization in wireless networks," in *Proc. IEEE INFOCOM'08*, Apr. 2008.

APPENDIX A
FULL PROOFS OF THE RESULTS IN SECTION II

Lemma 2.1: Let G be a graph. Then, $\sigma(G) \geq \mu(G)/\nu(G)$ (in the network graph sense).

Proof: Let $\boldsymbol{\alpha} : E(G) \to [0,1]$ be defined by $\boldsymbol{\alpha}(e) = 1/\nu(G)$ for all $e \in E(G)$. This is clearly a good edge weighting for G. Since every maximal matching in G has size at least $\nu(G)$, it follows that $\mu(G)/\nu(G) \leq \sum_{e\in Z} \boldsymbol{\alpha}(e) \leq 1$ for every maximal matching Z in G. Therefore, $\sigma(G) \geq \mu(G)/\nu(G)$. This proves Lemma 2.1. ∎

Let G be a graph. The following lemma provides a useful method for constructing optimal solutions to the following Linear Program, which is the interference graph analogue of (2):

$$\sigma(G) = \max \sigma \quad \text{subject to } \sigma \mathbf{e}^T \leq \boldsymbol{\alpha}^T \mathbf{I}(G) \leq \mathbf{e}^T, \tag{4}$$

where $\boldsymbol{\alpha} \in [0,1]^{|V(G)|}$ and $\mathbf{I}(G)$ is the maximal independent set/vertex incidence matrix corresponding to G. For an integer $t \geq 0$ and an automorphism ϕ of G, we denote by ϕ^t the tth composition of ϕ with itself (where ϕ^0 denotes the identity function).

Lemma A.1: Let G be a vertex-transitive graph. If (4) has a solution $(\sigma, \boldsymbol{\alpha})$, then (4) has a solution $(\sigma, \boldsymbol{\alpha}')$ such that $\boldsymbol{\alpha}'(v) = c$ for all $v \in V(G)$.

Proof: Let $(\sigma, \boldsymbol{\alpha})$ be a solution of (4) such that

$$f(\boldsymbol{\alpha}) := \sum_{u,v} \big|\boldsymbol{\alpha}(u) - \boldsymbol{\alpha}(v)\big|$$

is minimum. If $f(\boldsymbol{\alpha}) = 0$, then the lemma holds. So suppose for a contradiction that $f(\boldsymbol{\alpha}) > 0$. Then let $x, y \in V(G)$ such that $|\boldsymbol{\alpha}(x) - \boldsymbol{\alpha}(y)|$ is maximum. Since G is vertex-transitive, there exists an automorphism ϕ of G such that $\phi(x) = y$. Let $T \geq 1$ be smallest such that $\phi^T = \phi$ and define

$$\boldsymbol{\beta} = \frac{1}{T} \sum_{t=0}^{T-1} \boldsymbol{\alpha} \circ \phi^t.$$

In this expression, because ϕ is an automorphism, every term in the summation corresponds to a solution $(\sigma, \boldsymbol{\alpha} \circ \phi^t)$ of (4). Since $\boldsymbol{\beta}$ is the convex combination of solutions of (4), $\boldsymbol{\beta}$ is also a solution of (4). It follows from the triangle inequality that, for $u, v \in V(G)$,

$$\big|\boldsymbol{\beta}(u) - \boldsymbol{\beta}(v)\big| \leq \frac{1}{T} \sum_{t=0}^{T-1} \big|\boldsymbol{\alpha}(\phi^t(u)) - \boldsymbol{\alpha}(\phi^t(v))\big|.$$

By the construction of $\boldsymbol{\beta}$, we have $\boldsymbol{\beta}(x) = \boldsymbol{\beta}(y)$. Notice also that $f(\boldsymbol{\alpha}) = f(\boldsymbol{\alpha} \circ \phi^t)$ for all t. Now, since $|\boldsymbol{\alpha}(x) - \boldsymbol{\alpha}(y)| > 0$, we obtain

$$\begin{aligned}
f(\boldsymbol{\beta}) &= \sum_{u,v} \big|\boldsymbol{\beta}(u) - \boldsymbol{\beta}(v)\big| \\
&= \sum_{\substack{u,v \\ \{u,v\} \neq \{x,y\}}} \big|\boldsymbol{\beta}(u) - \boldsymbol{\beta}(v)\big| \\
&\leq \sum_{\substack{u,v \\ \{u,v\} \neq \{x,y\}}} \left[\frac{1}{T} \sum_{t=0}^{T-1} \big|\boldsymbol{\alpha}(\phi^t(u)) - \boldsymbol{\alpha}(\phi^t(v))\big| \right] \\
&\leq \frac{1}{T} \left[\left(\sum_{u,v} \big|\boldsymbol{\alpha}(u) - \boldsymbol{\alpha}(v)\big| \right) - \big|\boldsymbol{\alpha}(x) - \boldsymbol{\alpha}(y)\big| \right] \\
&\quad + \frac{1}{T} \sum_{t=1}^{T-1} f(\boldsymbol{\alpha} \circ \phi^t) \\
&< \frac{1}{T} \sum_{u,v} \big|\boldsymbol{\alpha}(u) - \boldsymbol{\alpha}(v)\big| + \frac{1}{T} \sum_{t=1}^{T-1} f(\boldsymbol{\alpha}) \\
&= \sum_{u,v} \big|\boldsymbol{\alpha}(u) - \boldsymbol{\alpha}(v)\big| = f(\boldsymbol{\alpha}),
\end{aligned}$$

which contradicts the assumption that $\boldsymbol{\alpha}$ was chosen with $f(\boldsymbol{\alpha})$ minimum. This proves Lemma A.1. ■

Lemma 2.3: If G is vertex-transitive, then $\sigma(G) = \bar{\mu}(G)/\bar{\nu}(G)$ (in the interference graph sense).

Proof: It follows from Lemma A.1 that there exists an optimal solution $(\sigma, \boldsymbol{\alpha})$ for the Linear Program (4) such that $\boldsymbol{\alpha}(v) = c$ for all $v \in V(G)$ for some c. Therefore, (4) may be reduced to the following Linear Program in two variables:

$$\begin{aligned} \sigma(G) = \quad & \max \ \sigma \\ & \text{subject to } \sigma \mathbf{e}^T \leq c\,\mathbf{M}(G) \leq \mathbf{e}^T. \end{aligned} \tag{5}$$

In this Linear Program, it is clearly optimal to choose c as large as possible and choose σ as large as possible subject to the choice of c. Clearly, the largest possible value of c is $\frac{1}{\bar{\nu}(G)}$. The corresponding largest possible value of σ is $\frac{\bar{\mu}(G)}{\bar{\nu}(G)}$. This proves Lemma 2.3. ■

The following lemma is an easy corollary of Lemma 2.3:

Lemma 2.2: If G is edge-transitive, then $\sigma(G) = \mu(G)/\nu(G)$ (in the network graph sense).

Proof: Notice that it follows, from the fact that G is edge-transitive, that the line graph $L(G)$ of G is vertex-transitive. Moreover, since matchings in G correspond to independent sets in $L(G)$, it follows that $\mu(G) = \overline{\mu}(L(G))$ and $\nu(G) = \overline{\nu}(L(G))$. Hence, it follows from Lemma 2.3 that $\sigma(G) = \overline{\mu}(L(G))/\overline{\nu}(L(G)) = \mu(G)/\nu(G)$. ■

Appendix B
Full proofs of the results in Section III

Lemma 3.1: Let G be an $\mathcal{F}$-free graph and let B be a block of G. Let F be a cycle in B that has maximum length. If $|V(F)| \geq 5$, then B is of the $\mathcal{B}_1$ type.

Proof: We prove this by induction on $|V(B)|$. It follows from the definition of $\mathcal{F}$ that $k \in \{5, 7\}$. Let $f_1, f_2, \ldots, f_k$ be the nodes of F. We will start with a number of subclaims:

(i) *Every node in $V(B) \setminus V(F)$ is a clone for F.*

Let $x \in (V(B) \setminus V(F))$. Since $|V(B)| \geq |V(F)| \geq 5$ and $|V(B)|$ has no cut-node, it follows that B is 2-connected and hence there exist two edge-disjoint paths P_1 and P_2 from x to two distinct nodes of F, say f_i and f_j, respectively. From the symmetry, we may assume that $i = 1$ and $j > k/2$. First assume that $|E(P_1)| + |E(P_2)| \geq 3$. Now f_1-P_1-x-P_2-f_j-f_{j-1}-$\cdots$-f_2-f_1 is a cycle of length $|E(P_1)| + |E(P_2)| + j > 3 + k/2$, contradicting the maximality of F.

It follows that $|E(P_1)| + |E(P_2)| = 2$ and, therefore, $|E(P_1)| = |E(P_2)| = 1$. Thus, x has two neighbors in $V(F)$. If x has two consecutive neighbors in $V(F)$, say f_1, f_2, then f_1-x-f_2-f_3-$\cdots$-f_{k-1}-f_k-f_1 is a cycle of length $k+1$, contary to the maximality of F. If $k = 5$, then, since x has at least two neighbors in $V(F)$, it follows that x is a clone for F. So we may assume that $k = 7$. Suppose that x is adjacent to f_p and f_{p+3} for some $p \in \{1, 2, \ldots, 7\}$. From the symmetry, we may assume that $p = 1$. But now f_1-x-f_4-f_5-f_6-f_7-f_1 is a cycle of length six, a contradiction. From the symmetry, it follows that x has exactly two neighbors in F and they are f_q and f_{q+2} for some $q \in \{1, 2, \ldots, 7\}$. Hence, x is a clone for F. This proves **(i)**.

(ii) *$V(B) \setminus V(F)$ is an independent set.*

Suppose that $x, y \in V(B) \setminus V(F)$ are adjacent nodes. We may assume that x is a clone of f_2. First suppose that y is also a clone of f_2. Then x-f_3-f_4-$\cdots$ f_{k-1}-f_k-f_1-y-x is a cycle of length $k+1$, contary to maximality of F. Next, suppose that y is a clone of a node at distance 2 of f_2, say f_k. Then x-f_1-f_2-$\cdots$-f_{k-1}-y-x is a cycle of length $k+1$, contary to maximality of F. Finally, suppose that $k = 7$ and y is a clone of a node at distance 3 of f_2, say f_5. It follows that x-f_1-f_2-f_3-f_4-f_5-f_6-y-x is a cycle of length eight, a contradiction. This proves **(ii)**.

Now suppose there exists $x \in (V(B) \setminus V(F)) \neq \emptyset$. It follows from the above that x is a clone for F. From the symmetry, we may assume that x is a clone of f_2. We claim that $\deg(f_2) = 2$. For suppose not. Then f_2 has a neighbor $y \notin \{f_1, f_2, f_3\}$. First suppose that $y \in (V(B) \setminus V(F))$. It follows from **(i)** that y is a clone of f_1 or of f_3. From the symmetry, we may assume that y is a clone of f_3. But now y-f_2-f_3-x-f_1-f_2-y is a cycle of length six, a contradiction. Therefore, it follows that $y = f_j$ for some $j \in \{4, 5, \ldots, k\}$. First assume that $j = 5$. Then x-f_1-f_2-f_5-f_4-f_3-x is a cycle of length six, a contradiction. From the symmetry, this leaves only the case where $k = 7$ and $j \in \{4, 6\}$. We may assume that $j = 4$. But now f_2-f_4-f_5-f_6-f_7-f_1-f_2 is a cycle of length six, a contradiction. This proves that $\deg(f_2) = 2$. It follows from the induction hypothesis that $B - f_2$ is of the $\mathcal{B}_1$ type. Therefore, by the definition of $\mathcal{B}_1$, it follows that B is of the $\mathcal{B}_1$ type.

So we may assume that $V(B) \setminus V(F) = \emptyset$. If $k = 5$, then we are done. So we may assume that $k = 7$. If F is an induced cycle in B, then we are also done. So we may assume that there is at least one edge between some f_i and f_j with $|i - j| \geq 2$.

(iii) *There is no $i \in \{1, 2, \dots, 7\}$ such that either (a) f_i is adjacent to f_{i+2}, or (b) f_i is adjacent to f_{i+3} and f_{i+1} is adjacent to f_{i+5}.*

From the symmetry, we may assume that $i = 1$. If f_i is adjacent to f_{i+2}, it follows that f_1-f_3-f_4-f_5-f_6-f_7-f_1 is a cycle of length six, a contradiction. For part (b), if f_i is adjacent to f_{i+3} and f_{i+1} is adjacent to f_{i+5}, then it follows that f_1-f_4-f_3-f_2-f_6-f_7-f_1 is a cycle of length six, a contradiction. This proves **(iii)**.

It follows from the above and from **(iii)** that there exists $i \in \{1, 2, \dots, 7\}$ such that f_i is adjacent to f_{i+3}. From the symmetry, we may assume that $i = 1$. It follows from **(iii)** that f_2 is non-adjacent to f_5 and f_6, and f_3 is non-adjacent to f_6 and f_7. Hence, the only possible other edges are $f_1 f_5$ and $f_4 f_7$. Therefore, B is of the $\mathcal{B}_1$ type. This proves Lemma 3.1. ∎

Lemma 3.2: Let G be an $\mathcal{F}$-free graph and let B be a block of G. Suppose that B contains no cycle of length at least 5. Then B is of the $\mathcal{B}_2$ type.

Proof: Since B has no cycle of length at least 5 and B has no cut-node, it follows from Theorem 3.4 that either B is a bipartite graph, or B is isomorphic to K_4, or B isomorphic to $K_{2,t}^+$. In the latter two cases, we are done. So suppose that B is a bipartite graph. Let $V(G) = X \cup Y$ such that X and Y are independent sets. If $|X| \leq 1$, then $x \in X$ is a cut-node, a contradiction. From the symmetry, it follows that $|X| \geq 2$ and $|Y| \geq 2$. Now suppose $x \in X$ is non-adjacent to $y \in Y$. Since B is 2-connected, it follows that there are two edge-disjoint paths P_1 and P_2 from x to y. Since x and y are non-adjacent and B is bipartite, it follows that $|E(P_1)| \geq 3$ and $|E(P_2)| \geq 3$. But now x-P_1-y-P_2-x is a cycle of length at least six, a contradiction. It follows that X is complete to Y. If $|X| \geq 3$ and $|Y| \geq 3$, then clearly, B contains a cycle of length six, a contradiction. Therefore, at least one of X, Y has size exactly 2. Hence, B is isomorphic to $K_{2,t}$ with $t = \max\{|X|, |Y|\}$ and therefore B is of the $\mathcal{B}_2$ type. This proves Lemma 3.2. ∎

Lemma B.1: Let $m \in \{5, 7\}$ and let $q \geq 0$. Let G' be a graph and let F be a m-cycle disjoint from G'. Let $v \in V(G')$ such that there exists a matching in G' that covers all neighbors of v in G', but not v itself. Let G be the graph constructed from the disjoint union of G' and F by adding a path P of length q between $f \in V(F)$ and v. Then every good edge weighting $\boldsymbol{\alpha}$ for G satisfies $\boldsymbol{\alpha}(e) = 0$ for every $e \in E(F) \cup E(P)$.

Proof: Let $f_1, f_2, \dots, f_m$ be the nodes of F in order and let $p_1, p_2, \dots, p_q$ be the nodes of P. We may assume that $f = f_m$, $p_1 = f$ and $p_q = v$. We use induction on q. First suppose that $q = 0$, i.e. $v = f_m$. We will prove this for the case when $m = 5$. The case when $m = 7$ is analogous. Let M be a maximal matching in G' that covers v. Let $M_1 = M \cup \{f_1 f_2, f_3 f_4\}$ and let $M_2 = M \cup \{f_2 f_3\}$. Since $\boldsymbol{\alpha}$ is a good edge weighting and M_1 and M_2 are maximal matchings, it follows that $\boldsymbol{\alpha}(f_2 f_3) = \boldsymbol{\alpha}(f_1 f_2) + \boldsymbol{\alpha}(f_3 f_4)$. Now let M' be a maximal matching in G' that does not cover v. Let $M_1' = M' \cup \{f_1 v, f_2 f_3\}$ and $M_2' = M' \cup \{f_1 v, f_3 f_4\}$. Since $\boldsymbol{\alpha}$ is a good edge weighting and M_1' and M_2' are maximal matchings, it follows that $\boldsymbol{\alpha}(f_2 f_3) + \boldsymbol{\alpha}(f_1 v) = \boldsymbol{\alpha}(f_3 f_4) + \boldsymbol{\alpha}(f_1 v)$. Hence, $\boldsymbol{\alpha}(f_2 f_3) = \boldsymbol{\alpha}(f_3 f_4)$. Using the symmetry, it follows that $\boldsymbol{\alpha}(f_2 f_3) = \boldsymbol{\alpha}(f_1 f_2)$. Combining this with the equality found above, it follows that $\boldsymbol{\alpha}(f_2 f_3) = 2\boldsymbol{\alpha}(f_2 f_3)$ and hence that $\boldsymbol{\alpha}(f_1 f_2) = \boldsymbol{\alpha}(f_2 f_3) = \boldsymbol{\alpha}(f_3 f_4) = 0$. Finally, let M'' be a maximal matching in G' that covers all neighbors of v but not v itself. Let $M_1'' = M'' \cup \{f_1 v, f_2 f_3\}$ and $M_2'' = M'' \cup \{f_1 f_2, f_3 f_4\}$. Since $\boldsymbol{\alpha}$ is a good edge weighting and M_1'' and M_2'' are maximal matchings, it follows that $\boldsymbol{\alpha}(f_1 v) + \boldsymbol{\alpha}(f_2 f_3) = \boldsymbol{\alpha}(f_1 f_2) + \boldsymbol{\alpha}(f_3 f_4) = 0$. Hence, $\boldsymbol{\alpha}(f_1 v) = 0$ and, from the symmetry, $\boldsymbol{\alpha}(f_4 v) = 0$. This proves the claim for $q = 0$.

Next, suppose that $q \geq 1$. It follows from the induction hypothesis that $\boldsymbol{\alpha}(e) = 0$ for all $e \in (E(F) \cup E(P)) \setminus \{p_{q-1} p_q\}$. Let M be a matching in G' that covers all neighbors of v but not v itself. Let M_1 be a maximal matching in $G|(V(F) \cup V(P))$ that covers v and let M_2 be a maximal matching in $G \setminus (V(F) \cup V(P))$ that does not cover v. Since $M \cup M_1$ and $M \cup M_2$ are maximal matchings, it follows that $\boldsymbol{\alpha}(M_1) = \boldsymbol{\alpha}(M_2)$. Since $\boldsymbol{\alpha}(M_2) = 0$, it follows that $\boldsymbol{\alpha}(p_{q-1} p_q) = 0$. This proves Lemma B.1. ∎

Lemma 3.3: $D_k^{p,q}$ fails SLoP for all $p, q \in \{5, 7\}$, $k \geq 0$.

Proof: Let $k \geq 0$, $p, q \in \{5, 7\}$ and suppose that $D_k^{p,q}$ satisfies SLoP. Then there exists a good edge weighting $\boldsymbol{\alpha}$ for $D_k^{p,q}$. It follows from Lemma B.1 applied to $D_k^{p,q}$ that $\boldsymbol{\alpha}(e) = 0$ for all $e \in E(D_k^{p,q})$. This is clearly not a good edge weighting for $D_k^{p,q}$, a contradiction. This proves Lemma 3.3. ∎

Lemma 3.4: Let G be a graph and let $x, x' \in V(G)$ be such that $\deg(x) = \deg(x') = 2$ and x' is a clone of x. Then G satisfies SLoP.

Proof: Let x and x' be as in the claim and let $\{z_1, z_2\} = N(x)$. Define $\boldsymbol{\alpha} \in [0, 1]^{|E|}$ by

$$\boldsymbol{\alpha}(e) = \begin{cases} 1/2 & \text{if } e \text{ is incident with } z_1 \text{ or } z_2\text{, and } e \neq z_1 z_2 \\ 1 & \text{if } e = z_1 z_2 \\ 0 & \text{otherwise.} \end{cases}$$

To see that $\boldsymbol{\alpha}$ is a good edge weighting for G, let M be a maximal matching in G'. If $z_1 z_2 \in M$, then no other edge in M is incident with z_1 or z_2 and hence $\sum_{e \in M} \boldsymbol{\alpha}(e) = 1$. Therefore we may assume that $z_1 z_2 \notin M$. It suffices to show that M covers both z_1 and z_2. So let us assume to the contrary that M does not cover z_1. Since M is a matching, at most one of $x z_2, x' z_2$ is in M. From the symmetry, we may assume that $x z_2 \notin M$. But now we may add $x z_1$ to the matching and obtain a larger matching, contary to the maximality of M. This proves Lemma 3.4. ■

Lemma 3.5: Every connected $\mathcal{F}$-free graph satisfies SLoP.

Proof: The proof is by induction on $|E(G)|$. Let $\{B_1, B_2, \ldots, B_q\}$ be the block decomposition of G. It follows from Theorem 3.3 that B_i is either of the $\mathcal{B}_1$ type or of the $\mathcal{B}_2$ type, and for at most one value of i, B_i is of the $\mathcal{B}_1$ type. Since, inductively, every proper subgraph of G satisfies SLoP, it suffices to find a good edge weighting $\boldsymbol{\alpha}$ for G.

Suppose first that G has a leaf block B_i of the $\mathcal{B}_2$ type. If $q = 2$, then let x be the cut-node of G in $V(B_i)$. If $q = 1$, let $x \in V(B_i)$ be arbitrary. There are four cases:

(1) B_i is isomorphic to K_2: let x, v denote the nodes of B_i. Let $\boldsymbol{\alpha}(e) = 1$ for all edges incident with x and $\boldsymbol{\alpha}(e) = 0$ for every other edge e. Let M be a maximal matching in G. If $xv \in M$, then, since M is a matching, M does not contain any other edge e with $\boldsymbol{\alpha}(e) = 1$ and, hence, $\sum_{e\in M} \boldsymbol{\alpha}(e) = 1$. If $xv \notin M$, then, since M is maximal, M contains an edge incident with x and, hence, $\sum_{e\in M} \boldsymbol{\alpha}(e) = 1$. Since this is true for every maximal matching M of G, it follows that $\boldsymbol{\alpha}$ is a good edge weighting for G.

(2) B_i is isomorphic to K_3: let x, v_1, v_2 denote the nodes of B_i and let $\boldsymbol{\alpha}(e) = 1$ for all $e \in E(B_i)$ and $\boldsymbol{\alpha}(e) = 0$ for every other edge e. Let M be a maximal matching in G. If $v_1v_2 \in M$, then, since M is a matching, M does not contain either of xv_1, xv_2 and, hence, $\sum_{e\in M} \boldsymbol{\alpha}(e) = 1$. If $v_1v_2 \notin M$, then, since M is maximal and M is a matching, exactly one of xv_1, xv_2 is in M and, hence, $\sum_{e\in M} \boldsymbol{\alpha}(e) = 1$. Since this is true for every maximal matching M of G, it follows that $\boldsymbol{\alpha}$ is a good edge weighting for G.

(3) B_i is isomorphic to K_4: let x, v_1, v_2, v_3 denote the nodes of B_i and let $\boldsymbol{\alpha}(v_1v_2) = \boldsymbol{\alpha}(v_1v_3) = \boldsymbol{\alpha}(v_2v_3) = 1$ and $\boldsymbol{\alpha}(e) = 0$ for all $e \in (E(G) \setminus \{v_1v_2, v_1v_3, v_2v_3\})$. To see that this is a good edge weighting, let M be a maximal matching in G. Suppose that M does not contain any of the edges v_1v_2, v_1v_3, v_2v_3. Since M does not contain v_1v_3 and M is maximal, it follows that M contains either xv_1 or xv_3. Assume without loss of generality that $xv_1 \in M$. Now we may add v_2v_3 to M to obtain a larger matching, a contradiction. It follows that $\sum_{e\in M} \boldsymbol{\alpha}(e) = 1$. Since this is true for every maximal matching M of G, it follows that $\boldsymbol{\alpha}$ is a good edge weighting for G.

(4) B_i is isomorphic to $K_{2,t}$ or $K^+_{2,t}$ for some $t \geq 2$: let $V(B_i) = V_1 \cup V_2$ such that $|V_1| = 2$ and V_2 is an independent set. Let $V_1 = \{y_1, y_2\}$ and let $V_2 = \{z_1, z_2, ..., z_t\}$. First suppose that B_i is isomorphic to $K^+_{2,2}$ and $x \in V_2$. We may assume that $x = z_1$. Set $\boldsymbol{\alpha}(y_1z_2) = \boldsymbol{\alpha}(y_2z_2) = \boldsymbol{\alpha}(y_1y_2) = 1$ and $\boldsymbol{\alpha}(e) = 0$ for all other edges e. Let M be a maximal matching in G. Suppose that M does not use any of the edges y_1z_2, y_2z_2, y_1y_2. Since M is a matching, at least one of the edges xy_1, xy_2 is not in M, say xy_1. But now we may add y_1z_2 to M to obtain a larger matching, a contradiction. It follows that $\sum_{e\in M} \boldsymbol{\alpha}(e) = 1$. Since this is true for every maximal matching M of G, it follows that $\boldsymbol{\alpha}$ is a good edge weighting for G. This solves the case when B_i is isomorphic to $K^+_{2,2}$ and $x \in V_2$. So we may assume this is not the case.

We claim that B_i contains two nodes p, p' of degree 2 such that p' is a clone of p. Suppose that $x \in V_1$. Then let $p = z_1$, $p' = z_2$. It follows that $\deg(p) = \deg(p') = 2$ and p' is a clone of p. Therefore, we may assume that $x \in V_2$. We may assume that $x = z_1$. Suppose that $|V_2| \geq 3$. Then let $p = z_2$, $p' = z_3$. It follows that $\deg(p) = \deg(p') = 2$ and p' is a clone of p. So we may assume that $|V_2| = 2$. From the above, it follows that B_i is isomorphic to $K_{2,2}$. Let $p = y_1$, $p' = y_2$. It follows that $\deg(p) = \deg(p') = 2$ and p' is a clone of p.

Now the result follows from Lemma 3.4.

Thus, we may assume that G does not have a leaf block of the $\mathcal{B}_2$ type. Since if $q \geq 2$, G has at least two leaf blocks, and hence at least one leaf block of the $\mathcal{B}_2$ type, we may assume that $q = 1$ and $G = B_1$ is of the $\mathcal{B}_1$ type. First suppose that $V(G) \setminus V(C) \neq \emptyset$. Then it follows from the definition of $\mathcal{B}_1$ that there exist two nodes x, x' such that $\deg(x) = \deg(x') = 2$ and $N(x) = N(x')$. It follows from Lemma 3.4 that there exists a good edge weighting for G. So we may assume that $V(G) = V(C)$. Suppose first that $k = 5$. Recall that it follows from the definition of $\mathcal{B}_1$ that G is a 5-cycle plus some arbitrary additional edges. Clearly, no maximal matching has size 1. Hence, since $|V(G)| = 5$, it follows that every maximal matching in G has size exactly 2. Therefore, $\boldsymbol{\alpha}(e) = 1/2$ for all $e \in E(G)$ is a good edge weighting for G. So we may assume that $k = 7$. Clearly, G has no maximal matching of size 1. It is also easy to see that G has no maximal matching of size 2. Hence, since $|V(G)| = 7$, it follows that every maximal matching in G has size exactly 3 and therefore $\boldsymbol{\alpha}(e) = 1/3$ for all $e \in E(G)$ is a good edge weighting for G. This proves Theorem 3.1. ■

Lemma 3.6: $|E(G)| \leq 2|V(G)|$ for every $\mathcal{F}$-free graph G.

Proof: We may assume that G is connected, because if not, then the lemma follows from considering each connected component of G. We first claim that $|E(B)| \leq 2|V(B)|$ for all $B \in \mathcal{B}_1$. Let $B \in \mathcal{B}_1$ and let C be a longest cycle in B. It follows from the definition of $\mathcal{B}_1$ that $|V(C)| \in \{5, 7\}$. Clearly, we have $|E(B_i)| \leq |V(C)| + 5 + 2(|V(B_i) \setminus V(C)|) \leq 2|V(C)| + 2(|V(B_i)| - |V(C)|) = 2|V(B_i)|$. This proves the claim. Next we claim that $|E(B)| \leq 2|V(B)| - 2$ for all $B \in \mathcal{B}_2$. If B is isomorphic to K_4, then $|E(B)| = 6 = 2|V(B)| - 2$. If B is isomorphic to $K_{2,t}$ or $K^+_{2,t}$ for some $t \geq 1$, then $|E(B)| \leq 1 + 2(|V(B)| - 2) < 2|V(B)| - 2$. This proves the claim.

Now let G be an $\mathcal{F}$-free graph and let $\{B_1, B_2, \ldots, B_q\}$ be the block decomposition of G. We prove by induction on q that $|E(G)| \leq 2|V(G)|$. If $q = 1$, it follows immediately from the above that $|E(G)| = |E(B_1)| \leq 2|V(B_1)| = 2|V(G)|$. Next, let $q \geq 2$. Since G has at least two leaf blocks and at most one block is in $\mathcal{B}_1$, we may choose i such that B_i is a leaf block and B_i is of the $\mathcal{B}_2$ type. Let x be the unique cut-node of G that lies in B_i. By induction, the graph $G|(V(B_i)\backslash\{x\})$ has at most $2(|V(G)| - |V(B_i)| + 1)$ edges. From the above, since B_i is of the $\mathcal{B}_2$ type, it follows that $|E(B_i)| \leq 2|V(B_i)| - 2$. Hence, we have $|E(G)| \leq 2(|V(G)| - |V(B_i)| + 1) + 2|V(B_i)| - 2 = 2|V(G)|$. This proves Lemma 3.6. ■

Lemma B.2: It can be decided in $O(|V(B)|)$ time whether a given graph B is of the $\mathcal{B}_1$ type.

Proof: We may assume that $|E(B)| \leq 2|V(B)|$, because if not, then it follows from Lemma 3.6 that B is not of the $\mathcal{B}_1$ type. Bodlaender [2] proved that, for any fixed k, finding a cycle of length at least k in a given graph H, if it exists, can be done in $O(k!2^k|V(H)|)$ time. The following algorithm uses Bodlaender's algorithm multiple times to recognize graphs of the $\mathcal{B}_1$ type.

(1) For $p = 8, 7, 6, 5$, do:

  Check if B contains a cycle of length p or more. If so, let F be the cycle and go to step (3).

(2) B does not contain a cycle of length 5 or larger, and hence B is not of the $\mathcal{B}_1$ type and we return NO.

(3) Let $k = |V(F)|$. If $k \in \{6, 8\}$, then B is not of the $\mathcal{B}_1$ type and we return NO. Let $f_1, f_2, \ldots, f_k$ be the nodes of F in order. If $k = 7$, check that the 'inner edges' of F are as in the definition of $\mathcal{B}_1$. If not, B is not of the $\mathcal{B}_1$ type and we return NO.

  For $i \in \{1, 2, \ldots, k\}$, do:

  Let A_i be the nodes in $V(B)\backslash V(F)$ that are adjacent to exactly f_{i-1} and f_{i+1}. If $|A_i| \geq 1$ and $\deg(f_i) \neq 2$, then B is not of the $\mathcal{B}_1$ type and we return NO.

  If $\sum_{i=1}^{k} |A_i| + |V(F)| < |V(B)|$, then B is not of the $\mathcal{B}_1$ type and return NO.

(4) B is of the $\mathcal{B}_1$ type and we return YES.

It is not hard to verify that this algorithm takes $O(|V(B)|)$ time. This proves Lemma B.2. ■

Lemma B.3: It can be decided in $O(|V(B)|)$ time whether a given graph B is of the $\mathcal{B}_2$ type.

Proof: We may assume that $|E(B)| \leq 2|V(B)|$, because if not, then it follows from Lemma 3.6 that B is not of the $\mathcal{B}_2$ type. Clearly, it can be checked in constant time whether B is isomorphic to K_2, K_3, K_4, $K_{2,2}$ or $K_{2,2}^+$. So we may assume that B is either isomorphic to $K_{2,t}$ or $K_{2,t}^+$ for some $t \geq 3$, or B is not of the $\mathcal{B}_2$ type. Let $X \subseteq V(B)$ be the set of nodes of degree 2. If $|X| \neq |V(B)| - 2$, then B is not of the $\mathcal{B}_2$ type and we may stop. Otherwise, let $\{a_1, a_2\} = V(B) \setminus X$. We need to check that X is an independent set and X is complete to $\{a_1, a_2\}$. If so, then B is of the $\mathcal{B}_1$ type and we may stop. If not, then B is not of the $\mathcal{B}_2$ type and we may stop. Notice that, since $|E(B)| \leq 2|V(B)|$, the check above can be done in $O(|E(B)|)$ time. This proves Lemma B.3. ■

Theorem 3.2: It can be decided in $O(|V(G)|)$ time whether a network graph G satisfies OLoP under primary interference.

Proof: We may assume that G is connected. By Theorem 3.1 and Theorem 3.3, it suffices to check whether G admits the structure described in Theorem 3.3. We propose the following algorithm. Let $n = |V(G)|$ and $m = |E(G)|$. First we check that $m \leq 2n$, because otherwise G is not $\mathcal{F}$-free by Lemma 3.6 and we stop immediately. Now, construct the block decomposition $\{B_1, B_2, ..., B_q\}$ of G. This can, in general, be done in $O(n + m)$ time (see e.g. [11]). However, since we know that $m \leq 2n$, this step actually takes $O(n)$ time. For each block B_i, we test whether B_i is of the $\mathcal{B}_2$ type. This can be done $O(|V(B_i)|)$ time by Lemma B.3. If G has more than one block that is not of the $\mathcal{B}_2$ type, then G is not $\mathcal{F}$-free and we stop. If we encounter no such block, then G is $\mathcal{F}$-free and we stop. So let B^* be the unique block that is not of the $\mathcal{B}_2$ type. It follows from Lemma B.2 that it can be decided in $O(|V(B^*)|)$ time whether B^* is of the $\mathcal{B}_1$ type or not. If it is, then G is $\mathcal{F}$-free and we stop. If not, then G is not $\mathcal{F}$-free and we stop. This proves Theorem 3.2. ■

Appendix C
Full proofs of the results in Section IV

Lemma C.1: Let G be a bipartite graph with bipartition X, Y. If $|X| \geq k$ and $\deg(x) \geq k$ for all $x \in X$, then every maximal matching in G has size at least k.

Proof: The proof is by induction on k. The lemma is clearly true for $k = 0$. So let $k \geq 1$. Let M be a maximal matching in G. Since X is not anticomplete to Y, it follows that M contains an edge xy with $x \in X$, $y \in Y$. Let $M' = M \setminus \{xy\}$, $X' = X \setminus \{x\}$, $Y' = Y \setminus \{y\}$ and $G' = G|(X' \cup Y')$. Clearly, M' is a maximal matching in G', $|X'| \geq k - 1$ and $\deg_{G'}(x') \geq k - 1$ for all $x' \in X'$. Hence, it follows by induction that $|M'| \geq k - 1$ and therefore that $|M| \geq k$. ■

Lemma 4.2: Let G be a bipartite graph with $\mu(G) \leq 4$. Then $\sigma(G) \geq 2/3$.

Proof: Write $\nu = \nu(G)$ and $\mu = \mu(G)$. It follows from Lemma 4.1.(b)-(c) that we may assume that $\deg(v) \geq 3$ for all $v \in V(G)$. If $\mu \geq \frac{2}{3}\nu$, then $\sigma(G) \geq \frac{2}{3}$ by Lemma 2.1. We may therefore assume that $\mu < \frac{2}{3}\nu$.

Let $V(G) = X \cup Y$ such that X and Y are independent sets. Let M^* be a maximal matching of size μ. Let A, B be the set of nodes in X, Y, respectively, that are covered by M^*. Let $C = Y \setminus B$ and $D = X \setminus A$. Since M^* is maximal, it follows that C is anticomplete to D. Moreover, by this and the fact that $\deg(v) \geq 3$ for all $v \in V(G)$, every $c \in C$ has at least three neighbors in A, and every $d \in D$ has at least three neighbors in B.

Let E_{AB}, E_{AC}, E_{BD} be the edges between A and B, A and C, and B and D, respectively. Since C is anticomplete to D, it follows that $E(G) = E_{AB} \cup E_{AC} \cup E_{BD}$.

We claim that:

$$|C|, |D| > \frac{1}{2}\mu. \tag{6}$$

Proof of the claim: Suppose to the contrary that $|C| \leq \frac{1}{2}\mu$ and let M be a maximal matching in G. Let $M_1 = M \cap E_{AB}$,

$M_2 = M \cap E_{AC}$, $M_3 = M \cap E_{BD}$. First, we have $|M_2| \leq |C| \leq \frac{1}{2}\mu$. Second, since every edge in $M_1 \cup M_3$ covers a unique node in B, it follows that $|M_1 \cup M_3| \leq |B| = \mu$. Therefore, $|M| \leq \frac{3}{2}\mu$. Since this is true for every maximal matching M, it follows that $\nu \leq \frac{3}{2}\mu$. But this means that $\mu \geq \frac{2}{3}\nu$, contrary to our assumption. Hence, $|C| > \frac{1}{2}\mu$ and, by the symmetry, that $|D| > \frac{1}{2}\mu$. This proves the claim.

If $\mu \leq 2$, then, since every node in C has at least three neighbors in A, it follows that $|C| = 0$, contrary to (6). Hence, $\mu \in \{3, 4\}$. It follows from (6) and if $\mu = 3$, then $|C|, |D| \geq 2$, and if $\mu = 4$, then $|C|, |D| \geq 3$. Define

$$\boldsymbol{\alpha}(e) = \begin{cases} \frac{1}{\mu} & \text{if } e \in E_{AB} \\ \frac{1}{2\mu} & \text{if } e \in E_{AC} \cup E_{BD}. \end{cases}$$

We need to prove that $\frac{2}{3} \leq \sum_{e \in M} \boldsymbol{\alpha}(e) \leq 1$ for every maximal matching M in G. So let M be a maximal matching in G. Since every edge in M is incident with a node of $A \cup B$, it is easy to see that $\sum_{e \in M} \boldsymbol{\alpha}(e) \leq 1$. Let $k = |M \cap E_{AB}|$. It suffices to show that $|M \cap E_{AC}| \geq \mu - k - 1$ and that $|M \cap E_{BD}| \geq \mu - k - 1$, because if so, then

$$\begin{aligned} \sum_{e \in M} \boldsymbol{\alpha}(e) &\geq k \times \frac{1}{\mu} + 2 \times (\mu - k - 1) \times \frac{1}{2\mu} \\ &= \frac{\mu - 1}{\mu} \geq \frac{2}{3}, \qquad \text{for } \mu \in \{3, 4\}. \end{aligned}$$

From the symmetry, it even suffices to show that $|M \cap E_{AC}| \geq \mu - k - 1$. To do so, let A' be the nodes of A that are not covered by M. We may assume that $k < \mu - 1$, because otherwise there is nothing to prove. Consider the graph $G' = G|(A' \cup C)$. Clearly, we have $|C| \geq \mu - k - 1$, $\deg_{G'}(c) \geq 3 - k \geq \mu - k - 1$ for all $c \in C$. Moreover, $M \cap E_{AC}$ is a maximal matching in G'. Hence, it follows from Lemma C.1 that $|M \cap E_{AC}| \geq \mu - k - 1$. This proves Lemma 4.2. ∎

Appendix D
Full proofs of the results in Section V

Observation 5.1: $\sigma^*(F_k) \leq \left(\frac{2}{3}\right)^k$ for all $k \geq 1$.

Proof: Clearly, every F_k is vertex-transitive. Let us consider F_2. A maximum independent set in F_2 can be constructed by first choosing three non-consecutive 6-cycles and, next, choosing three non-consecutive nodes from each of these three 6-cycles. It is clear that this constitutes a maximum independent set and its size is $3 \times 3 = 9$. A minimum maximal independent set in F_2 can be constructed by choosing two opposite 6-cycles and, next, choosing two opposite nodes from each of these two 6-cycles. This gives a maximal independent set of size $2 \times 2 = 4$. Since F_2 is vertex-transitive, it follows from a direct extension of Lemma 2.1 to interference graphs that $\sigma(F_2) = 4/9$ and hence $\sigma^*(F_2) \leq 4/9$. This reasoning extends easily to the general case, where we have $\overline{\nu}(F_k) = 3^k$ and $\overline{\mu}(F_k) = 2^k$. Therefore, $\sigma^*(F_k) \leq \left(\frac{2}{3}\right)^k$. ∎