Online Scheduling and Routing with End-to-End Deadline Constraints in Multihop Wireless Networks

Christos Tsanikidis, Javad Ghaderi
Department of Electrical Engineering, Columbia University
New York, NY, USA
{c.tsanikidis,jghaderi}@columbia.edu

ABSTRACT

We consider scheduling deadline-constrained packets in multihop wireless networks. Packets with arbitrary deadlines and weights arrive at and are destined to different nodes. The goal is to design online admission, routing, and scheduling algorithms in order to maximize the cumulative weight of packets that reach their destinations within their deadlines. Under a general interference graph model of the wireless network, we provide online algorithms that are (γ, R)-competitive, i.e., they achieve at least $1/\gamma$ fraction of the value of the optimal offline algorithm, and do not exceed the capacity by more than a factor $\mathrm{R} \geq 1$. In particular, our algorithm can achieve $\gamma = O(\psi^\star \log(\Delta\rho L)/\mathrm{R})$ when $\mathrm{RC} = \Omega(\psi^\star \log(\Delta\rho L))$, where ρ is the ratio of maximum weight to minimum weight of packets, L is the length of the longest route of packets, and C is the minimum link capacity or the number of channels. Here, Δ is the *maximum degree* and $\psi^\star$ is the *local clique cover number* of the interference graph. Our results translate directly to many networks of interest, for example, in one-hop interference networks, $\psi^\star = 2$, and in the case of wired networks (no interference), $\psi^\star = 1$. We further provide lower bounds that show that our results are asymptotically optimal in many settings. Finally, we present extensive simulations that show our algorithms provide significant improvement over the prior approaches.

CCS CONCEPTS

• **Networks** → **Network algorithms**.

1 INTRODUCTION

Scheduling real-time traffic in communication networks has gained significant importance due to emerging real-time applications, e.g. in Internet of Things (IoT), vehicular networks, and other cyber-physical systems, where time-sensitive packets need to be carried across wired or wireless networks. Meeting the deadline requirements of these packets requires a departure from traditional schedulers that only focus on throughput. Despite the recent advances in scheduling real-time traffic in "single-hop" wireless networks [1–5] and multihop "wired" networks [6–8], the multihop wireless setting has remained notoriously difficult. In this setting, the space of decisions is considerably larger as it involves the path a packet takes, the set of non-interfering links scheduled in the network at any time, as well as the specific time slots the packet occupies for transmission over the scheduled links. Scheduling a packet at a link will impact the decisions at other links in future time. Moreover, one often expects that not all packets are equally important. Considering different rewards (weights) among the packets makes the problem even more complicated.

This paper makes important progress on scheduling deadline-constrained packets in multihop wireless networks. The objective is to maximize the total reward of the packets that reach their destinations within their deadlines. We provide a framework for designing online algorithms for this problem for *general interference graphs*. We state the performance in terms of bi-criteria competitive ratio: an online algorithm is (γ, R)-competitive if it achieves at least $1/\gamma$ fraction of the reward of the optimal offline algorithm, and does not exceed the network's capacity by more than a factor $\mathrm{R} \geq 1$.

The obtained competitive ratios are stated as a function of the parameters of the problem, namely, ρ: the ratio of the maximum weight to the minimum weight of packets, L: the length of the longest route of packets in the network, and C: the minimum link capacity or the number of channels. Our results further depend on the properties of the interference graph of the wireless network, such as Δ: the maximum number of interferers of any link, and β: the maximum number of links that can be activated out of a link and its set of interfering links (a.k.a. interference degree).

1.1 Related Work

There is rich literature on online packet scheduling with deadlines, however, it mainly pertains to single-hop wireless networks or multihop wired networks.

Wireless Networks. The work on deadline-constrained scheduling in wireless networks has mostly focused on single-hop traffic, e.g., [1–5]. In [3], for single-hop wireless networks, it is shown that by using an α-approximation of the well-known Max Weight Scheduling (MWS) policy, it is possible to obtain $\frac{\alpha}{\alpha+1}$ of the "real-time" capacity region. For example, a greedy scheduler is shown to achieve $\frac{1}{\beta+1}$ fraction of the capacity region, where β is the interference degree of the graph. Moreover, it is shown in [2, 3] that randomized algorithms can improve this ratio. The work on the multihop wireless setting has been very limited. The problem has been considered by [9], where the bandwidth is divided in C orthogonal channels, and the results are obtained under various limits for the traffic and C. The multi-channel model assumed in our paper is similar to the one in [9], however we consider the case of finite C as opposed to C $\to \infty$ in [9]. Further, our results concern the worst-case packet sequences as opposed to the *i.i.d.* setting in [9].

MobiHoc '22, October 17–20, 2022, Seoul, Republic of Korea

ACM ISBN 978-1-4503-9165-8/22/10…$15.00
https://doi.org/10.1145/3492866.3549729

Wired Networks. In the single-hop setting, there is considerable work on single-link buffer management, e.g., [10, 11], where it is possible to design online algorithms with constant approximation ratios. For multihop traffic, the problem is significantly more challenging and has been studied in a sequence of papers [6–8, 12–15]. We highlight two papers [7, 8] that have the best theoretical guarantees. [7] considers the unweighted packets case, i.e., $\rho = 1$, and provides two online algorithms, one with resource augmentation $\mathrm{R} > 1$ (i.e., when the link capacity is increased by a factor of R), and one without augmentation, with $\mathrm{R} = 1$. Their algorithm for $\mathrm{R} = 1$ is asymptotically optimal as the minimum link capacity $C_{\min} \to \infty$, as it becomes $O(\log L)$-competitive, matching a previously known lower bound by [6]. For $\mathrm{R} > 1$, in the limit of $\mathrm{C}_{\min} \to \infty$, [7] provides a competitiveness of $1 + \frac{L}{e^{\mathrm{R}}-1}$, which is linear in L for constant R. Recently, [8] provided an algorithm called GLS-ADP that is $(\mathrm{C}_{\max} L^{\frac{1}{\mathrm{C}_{\min}}})$-competitive in the unweighted case ($\rho = 1$). This can yield logarithmic competitiveness if $\mathrm{C}_{\min} = \log L$, however clearly for larger $\mathrm{C}_{\max}$ the bound deteriorates. In addition, [8] provides an algorithm called GLS in the case of weighted packets with fixed routes, which is $(\rho(L+1))$-competitive.

Our algorithm draws inspiration from the primal-dual techniques, e.g., [7, 16]. However, the optimization problem in our setting is considerably harder than the one in wired networks due to the presence of interference constraints among the links. In this case, the set of scheduled links at any time should be an independent set of the network's interference graph. A direct LP relaxation of the optimization problem in this case yields a solution which turns out to be too loose. In this paper, we propose a tighter relaxation by adding constraints corresponding to the *local clique covers* of the interference graph. Although there is vast literature on LP relaxation of independent set constraints, we have not identified a similar relaxation in the past work.

1.2 Main Contributions

Our main contributions can be summarized as follows.

Scheduling in Wireless Multihop Networks. We provide an online algorithm for scheduling and routing deadline-constrained packets in wireless multihop networks with general interference graphs. Our algorithm relies on a new relaxation technique based on local clique covers of the interference graph, followed by a primal-dual technique to obtain an approximate solution to the relaxed problem, and finally, a greedy technique to map the primal-dual solution to the independent sets of the network's interference graph. Our results depend on the properties of the constructed local clique cover, captured through two parameters ξ and ψ. In particular, in an interference graph with maximum degree Δ, our algorithm yields competitiveness $\gamma = O(\psi \log(\xi\rho L)/\mathrm{R})$ if $\mathrm{RC} = \Omega(\psi \log(\xi\rho L))$, where $\xi \leq \Delta + 1$, $\beta \leq \psi \leq \Delta$, and β is the interference degree. Our results translate directly to many interference graphs of interest, for example, in the case of one-hop interference model, $\psi = \xi = 2$.

Scheduling in Wired Multihop Networks. Our framework applies to the special case of wired networks with $\psi = \xi = 1$. In this case, our results translate to an $O(\log(\rho L)/\mathrm{R})$-competitive algorithm when $\mathrm{R}C_{\min} = \Omega(\log(\rho L))$. This improves the prior results [7, 8] for general ρ (weighted packets) in terms of the assumptions required to achieve a logarithmic competitiveness. Further, our results are stated in terms of known constants.

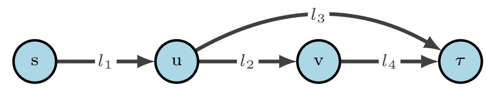


Figure 1: A multihop network where at time 1 a packet arrives at s and needs to be delivered to τ with deadline 5. $k = \{(l_1, 1), (l_3, 3)\}$ and $k' = \{(l_1, 2), (l_2, 3), (l_4, 5)\}$ are both valid route-schedules. $\{(l_1, 2), (l_2, 4), (l_4, 6)\}$ is not a valid route-schedule because the packet expires at time 6. Given a collection of packets, their selected route-schedules should further respect the interference constraints among the links.

Performance Lower Bounds. We provide lower bounds on the performance guarantee of online algorithms. We show that online admission-control and scheduling algorithms in networks with unit capacity cannot be better than $\rho\beta(L-1)$-competitive in general interference networks. Similarly, in the case of wired networks, no such algorithm can be better than $\rho(L-1)$-competitive when $\mathrm{C} = 1$. Further, for general online algorithms, $O(\log(\rho L))$ is a lower bound on competitiveness. These lower bounds, along with the guaranteed upper bounds, indicate that our algorithms are optimal up to constant multiplicative factors when $\mathrm{R} = 1$.

2 NETWORK MODEL AND PROBLEM SETUP

We consider a communication network represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{L})$, where each $v \in \mathcal{V}$ is a node, and each edge $l = (u, v) \in \mathcal{L}$ (with $u, v \in \mathcal{V}$) indicates a communication link over which packets can be transmitted between two nodes. Time is divided into time slots $t = 1, 2, 3, \ldots$, where a time slot is the time required to complete transmission of one packet over a link. A sequence of packets $\mathcal{M}$ arrive to the nodes in the network over time. Each packet $m \in \mathcal{M}$ has a source node s_m, a destination node τ_m, an arrival time a_m, a deadline d_m (relative to the arrival), and a weight w_m. Let $w_{\max}$ denote the maximum weight of a packet, $w_{\min}$ be the minimum weight (with $\rho = w_{\max}/w_{\min}$), and d_{max} be the maximum deadline. To receive reward w_m, the packet m has to be delivered to its destination before time slot $a_m + d_m$.

Definition 1 (Route-Schedule). A *route-schedule* k for packet m is a sequence of link-time slot tuples $(l_1, t_1), (l_2, t_2) \ldots, (l_n, t_n)$, such that by transmitting the packet over link l_i at time t_i, $i = 1, \cdots, n$, $a_m \leq t_1 < \cdots < t_n < a_m + d_m$, it can be successfully transmitted from its source to its destination within its deadline. We define the length of k as $|k| = n$. We use $\mathcal{K}_m$ to denote the set of all valid route-schedules for packet m. We define L to be the maximum length of any route-schedule, i.e., $L = \max_{m \in \mathcal{M}} \max_{k \in \mathcal{K}_m} |k|$. Note that by definition, $L \leq d_{max}$, where d_{max} is the maximum deadline.

See Figure 1 for an illustration of a route-schedule in a network.

As we mentioned earlier, we state our results in terms of resource augmented competitive ratios. Formally, suppose that the capacity of the network is scaled by a factor of $\mathrm{R} \geq 1$. Let μ denote an online scheduling algorithm. Let $P^{\mu}(\mathrm{R})$ be the total reward collected by applying μ on the packet sequence $\mathcal{M}$, with the resource augmentation R. Let $P^{\star}$ denote the total reward collected by an optimal offline algorithm without any resource augmentation. We say online algorithm μ is $(\gamma(\mathrm{R}), \mathrm{R})$-competitive if:

$$\frac{P^{\mu}(\mathrm{R})}{P^{\star}} \geq \frac{1}{\gamma(\mathrm{R})}, \tag{1}$$

for *any* sequence of packets $\mathcal{M}$, i.e., μ achieves a worst-case approximation ratio of $1/\gamma(\mathrm{R})$. Our goal is to design an online algorithm such that the approximation ratio $1/\gamma(\mathrm{R})$ is as large as possible, or its competitiveness $\gamma(\mathrm{R})$ is as small as possible, for $\mathrm{R} \geq 1$.

In the rest of the paper, we use $[n]$ to denote the set $\{1, 2, \cdots, n\}$, and use $\mathbb{N} = \{1, 2, \cdots\}$ to denote the set of positive integers.

2.1 General Interference Model

We assume a set of $\mathrm{C} \geq 1$ orthogonal frequency channels [9]. As a result of wireless interference, certain links cannot be activated at the same time in the same channel. We use an *interference graph* $\mathcal{G}_I = (\mathcal{L}, \mathcal{E}_I)$, e.g., [1, 2, 9], to represent the interference relationships between links. In $\mathcal{G}_I$, vertices correspond to communication links and there is an edge between any two links that interfere with each other. Hence, the set of links scheduled at the same time in the same channel should form an *independent set* of $\mathcal{G}_I$. We indicate the maximum degree of $\mathcal{G}_I$ with Δ.

We use $\mathbf{Y}_t^{\mathrm{c}}$ to denote the independent set used at time t in channel c, and define $Y_{lt}^{\mathrm{c}} \in \{0, 1\}$ to be 1 if link l at time t belongs to independent set $\mathbf{Y}_t^{\mathrm{c}}$, and 0 otherwise.

Definition 2 (Network-Schedule)**.** A valid network-schedule at time t is a set of RC independent sets $\{\mathbf{Y}_t^{\mathrm{c}}, \mathrm{c} \in [\mathrm{RC}]\}$ of the interference graph, over which packets can be transmitted on the corresponding links and channels.

We define $X_{mk} \in \{0, 1\}$ to be 1 if the route-schedule $k \in \mathcal{K}_m$ is selected for packet m (and there is a way to schedule the packet subject to the interference constraints), and is 0 otherwise.

Our goal is to maximize the total reward of packets that reach their destinations before they expire. Given a resource augmentation $\mathrm{R} \geq 1$, the timely reward maximization can be formulated as the following Integer Program which we refer to as $P_{\mathrm{WG}}(\mathrm{R})$:

$$\max_{\mathrm{X,Y}} \quad \sum_{m \in \mathcal{M}} w_m \sum_{k \in \mathcal{K}_m} X_{mk} \qquad (: P_{\mathrm{WG}}(\mathrm{R})) \tag{2a}$$

$$\text{s.t.} \quad Y_{lt}^{\mathrm{c}} + Y_{l't}^{\mathrm{c}} \leq 1, \quad \forall (l, l') \in \mathcal{E}_I, \forall \mathrm{c} \in [\mathrm{RC}], \forall t \tag{2b}$$

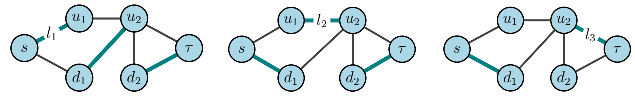
$$\sum_{m \in \mathcal{M}} \sum_{k \in \mathcal{K}_m : (l,t) \in k} X_{mk} \leq \sum_{\mathrm{c}=1}^{\mathrm{RC}} Y_{lt}^{\mathrm{c}}, \quad \forall l \in \mathcal{L}, \forall t \tag{2c}$$

$$\sum_{k \in \mathcal{K}_m} X_{mk} \leq 1, \quad \forall m \in \mathcal{M} \tag{2d}$$

$$X_{mk} \in \{0, 1\}, \quad \forall m \in \mathcal{M}, k \in \mathcal{K}_m \tag{2e}$$

$$Y_{lt}^{\mathrm{c}} \in \{0, 1\}, \quad \forall \mathrm{c} \in [\mathrm{RC}], \forall l \in \mathcal{L}, \forall t \tag{2f}$$

In the above, R is the augmentation factor. Constraints (2b) and (2f) state the requirement that the network schedule at any time t for every channel $\mathrm{c} \in [\mathrm{RC}]$ has to be an independent set $\mathbf{Y}_t^{\mathrm{c}}$. Constraints (2d) and (2e) state that at most one route-schedule k is selected for each arriving packet $m \in \mathcal{M}$. Constraint (2c) states that the number of transmitted packets over link l at time t cannot be more than the number of times that link l is scheduled considering all the channels, i.e., each packet transmission on link l at time t will be completed using a network independent set on a single channel. Hence, Constraints (2b)–(2f) imply that in order for a specific packet m to be admitted and scheduled using the route-schedule k, i.e., $X_{mk} = 1$, there must be a sequence of $|k|$ independent sets, one for each link that packet m traverses. With all the constraints satisfied, any association of packets to independent sets is equivalent.



(a) Matching $\mathrm{Y}_{t_1}^{\mathrm{c}_1}$ (b) Matching $\mathrm{Y}_{t_2}^{\mathrm{c}_2}$ (c) Matching $\mathrm{Y}_{t_3}^{\mathrm{c}_3}$

Figure 2: A sequence of matchings $\mathrm{Y}_{t_1}^{\mathrm{c}_1}, \mathrm{Y}_{t_2}^{\mathrm{c}_2}, \mathrm{Y}_{t_3}^{\mathrm{c}_3}$ at three time slots $t_1 < t_2 < t_3$, in channels $\mathrm{c}_1, \mathrm{c}_2, \mathrm{c}_3$, respectively. This schedule can deliver a packet m expiring at $d_m + a_m > t_3$ from s to τ. In this case, the route-schedule for m is $k_1 = \{(l_1, t_1), (l_2, t_2), (l_3, t_3)\}$, and $X_{mk_1} = 1$.

We refer to the unaugmented problem simply by $P_{\mathrm{WG}} \equiv P_{\mathrm{WG}}(1)$.

Note that in general there are exponentially many route-schedules in a network, hence the optimization problem $P_{\mathrm{WG}}(\mathrm{R})$ could have exponentially many variables. Furthermore, $P_{\mathrm{WG}}(\mathrm{R})$ is clearly NP-hard even in simple scenarios (e.g., with a reduction of the Maximum Independent Set problem to this problem).

Next, we present two special cases of the above model, namely, one-hop interference, and no interference (i.e., wired network).

2.2 One-hop Interference Model

The one-hop (or node-exclusive) interference model, e.g. [17–20], assumes that two adjacent links cannot be scheduled at the same time in the same channel. Formally, define the links incident to node $v \in V$ as $\mathcal{A}(v) = \{l \in \mathcal{L} : l = (v, u), u \in V\}$.

One-hop interference model requires that for every $v \in V$, at most one link $l \in \mathcal{A}(v)$ is scheduled. Equivalently, at any time t and any channel c, the set of scheduled links should form a "*matching*" of the communication graph $\mathcal{G}$. Hence, the network-schedule $\mathbf{Y}_t^{\mathrm{c}}$ in this case corresponds to the matching used at time t on channel c. The reward maximization problem in this special case is as follows:

$$\max_{\mathrm{X,Y}} \quad \sum_{m \in \mathcal{M}} w_m \sum_{k \in \mathcal{K}_m} X_{mk} \qquad (: P_{\mathrm{WM}}(\mathrm{R})) \tag{3a}$$

$$\text{s.t.} \quad \sum_{l \in \mathcal{A}(v)} Y_{lt}^{\mathrm{c}} \leq 1, \quad \forall v \in V, \forall \mathrm{c} \in [\mathrm{RC}], \forall t \tag{3b}$$

$$\text{Constraints } (2c), (2d), (2e), (2f).$$

Refer to Figure 2 for an example illustrating transfer of a packet through matchings in the case of one-hop interference.

2.3 Wired Networks (No Interference)

Our general model trivially translates to this case. Since there is no interference, all links can be activated at the same time, i.e., we can set $\mathbf{Y}_t^{\mathrm{c}} = \mathbf{1}$. Then $\sum_{\mathrm{c}=1}^{\mathrm{RC}} Y_{lt}^{\mathrm{c}} = \mathrm{RC}$. In this case, we enrich the model slightly so that each link can have a different capacity, to obtain a model analogous to the one in [7, 8]. Let $\mathrm{C}_l \in \mathbb{N}$ be the capacity of link l, which is the maximum number of packets that can be transmitted over l at a time slot. The new optimization $P_{\mathrm{WD}}(\mathrm{R})$ is then as follows:

$$\max_{\mathbf{X}} \quad \sum_{m \in \mathcal{M}} w_m \sum_{k \in \mathcal{K}_m} X_{mk} \qquad (: P_{\mathrm{WD}}(\mathrm{R})) \tag{4a}$$

$$\text{s.t.} \quad \sum_{m \in \mathcal{M}} \sum_{k \in \mathcal{K}_m : (l,t) \in k} X_{mk} \le \mathrm{RC}_l, \quad \forall l \in \mathcal{L}, \forall t \tag{4b}$$

$$\text{Constraints } (2d), (2e).$$

Our algorithm in its general form relies on a clique cover of the interference graph and depends on the local properties of this cover, ψ, ξ. Hence, in the next section, we introduce some preliminaries and make a few definitions.

3 DEFINITIONS AND PRELIMINARIES: CLIQUE COVER

We define the (extended) *neighborhood* of link l as the set of links that interfere with l, including link l itself, i.e., $\mathcal{N}_l := \{l' \in \mathcal{L} : (l, l') \in \mathcal{E}_I\} \cup \{l\}$. We use $\mathcal{G}_I[\mathcal{L}']$ to denote the induced subgraph of $\mathcal{G}_I$ on a subset of links $\mathcal{L}' \subseteq \mathcal{L}$. A set of links Q is a *clique* for $\mathcal{G}_I$ if $\mathcal{G}_I[Q]$ is a complete graph. In other words, a clique for $\mathcal{G}_I$ is a set of links that all interfere with each other. We define a (vertex-) *clique cover* of $\mathcal{G}_I$ as a set $\mathcal{Q}$ of cliques in $\mathcal{G}_I$ that can cover the entire set $\mathcal{L}$, i.e., $\cup_{Q \in \mathcal{Q}} Q = \mathcal{L}$. When sets in $\mathcal{Q}$ are all disjoint, we refer to $\mathcal{Q}$ as a *disjoint clique cover*.

We define the local properties of the clique cover as follows. Given a clique cover $\mathcal{Q}$ of $\mathcal{G}_I$, we use $\mathcal{Q}_l$ to denote a minimal subset of $\mathcal{Q}$ needed to cover the neighborhood of l. Formally,

$$\mathcal{Q}_l = \arg\min_{\hat{\mathcal{Q}} \subseteq \mathcal{Q}} \left(|\hat{\mathcal{Q}}|, \ \text{s.t. } \mathcal{N}_l \subseteq \cup_{Q \in \hat{\mathcal{Q}}} Q \right).$$

We define the *local clique cover degree* (lccd) of $\mathcal{Q}$ as

$$\psi := \psi(\mathcal{Q}) = \max_{l \in \mathcal{L}} |\mathcal{Q}_l|.$$

We further define the *clique involvement degree* ξ as the maximum number of cliques any link belongs to. Formally,

$$\xi := \xi(\mathcal{Q}) = \max_{l \in \mathcal{L}} |\{Q \in \mathcal{Q} : l \in Q\}|$$

We omit the dependence of ψ and ξ on $\mathcal{Q}$ when there is no ambiguity. As it will become evident in the performance guarantees of our algorithm (Section 4.1), it is desirable to select a $\mathcal{Q}$ that primarily minimizes ψ. Refer to Figure 3 for an illustration of clique covers and properties ψ, ξ in a graph.

We define the optimal lccd $\psi^\star$ to be the minimum lccd over all possible clique covers. We refer to $\psi^\star$ as the *local clique cover number* of $\mathcal{G}_I$. The following lemma allows to compute $\psi^\star$ more efficiently by considering local clique covers.

Lemma 1. *Consider a clique cover obtained by the union of minimal clique covers for all local subgraphs $\mathcal{G}_I[\mathcal{N}_l], \forall l \in \mathcal{L}$. This clique cover minimizes lccd, i.e., it achieves $\psi^\star$.*

Finally, we mention that one can define an *edge clique cover* as a covering of all the edges rather than the vertices of a graph. Since covering the edges requires covering the end-vertices of every edge, an edge clique cover is also a valid vertex clique cover. An example of an edge clique cover is cover B of Figure 3, while clique covers A and C are not edge clique covers.

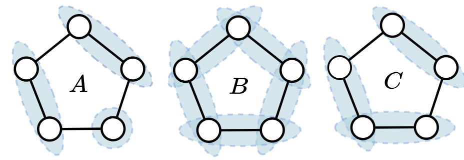


Figure 3: Three ways to perform a clique cover in the above graph. A is a minimal clique cover of the graph. B is a covering that would naturally arise by attempting to cover locally all neighborhoods. C is obtained from B by removing redundant cliques. In A: $\psi = 3, \xi = 1$, and in B and C: $\psi = 2, \xi = 2$. Hence, B or C give the minimum ψ. Note that for this graph $\psi^\star = \beta = 2$.

4 SCHEDULING UNDER GENERAL INTERFERENCE

In this section, we present the general form of our algorithm. We refer to this algorithm as GIMS (General Interference Multihop Scheduler), described in Algorithm 2. GIMS has three parts: (1) selection of a clique cover $\mathcal{Q}$, (2) admission and route-schedule assignment, and (3) channel assignment. We discuss these three parts below.

Selecting the Clique Cover. The choice of the clique cover $\mathcal{Q}$ directly depends on the interference graph. A predetermined clique cover can be specified for several families of graphs. For example, as we will see in Section 5, a natural clique cover arises in one-hop interference networks. In the case of arbitrary interference graphs, due to Lemma 1, we can attain $\psi^\star$ by identifying minimal local covers. Algorithm 1 describes one such algorithm based on a subroutine for finding the local clique covers at Line 1.4 (Line 4 of Algorithm 1). As a result of selecting a disjoint clique cover locally, we additionally have $\xi \le (\Delta + 1)$. For graphs where even finding a minimal clique cover locally is costly, we can simply use a greedy clique cover [21] at Line 1.4. The use of a greedy clique cover at Line 1.4 guarantees $\psi \le \Delta$ and $\xi \le \Delta + 1$. However, as we see in Section 10, for most graphs, even the greedy approach yields ψ close to β. We discuss the complexity in detail in Section 7.

The obtained clique cover $\mathcal{Q}$ might contain some redundant cliques. For example, cliques that are subsets of other cliques in the cover might arise. Algorithm 1 has a (optional) Prune step (Line 1.8) that iteratively removes cliques whose removal does *not* increase the value of ψ (as an attempt to reduce ξ).

Algorithm 1: Selecting Clique Cover $\mathcal{Q}$

1.1 **Input:** Interference graph $\mathcal{G}_I$
1.2 $\mathcal{Q} \leftarrow \emptyset$
1.3 **for** *each link* $l \in \mathcal{L}$ **do**
1.4 $\quad \mathcal{Q}_l \leftarrow$ Find a disjoint clique cover for $\mathcal{G}_I[\mathcal{N}_l]$
1.5 $\quad \mathcal{Q} = \mathcal{Q} \cup \mathcal{Q}_l$
1.6 **end**
1.7 **while** $\mathcal{Q} \ne \mathit{Prune}(\mathcal{Q})$ **do**
1.8 $\quad \mathcal{Q} \leftarrow \mathit{Prune}(\mathcal{Q})$
1.9 **end**
1.10 **Return:** $\mathcal{Q}$

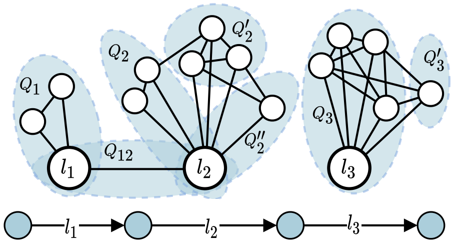


Figure 4: Consider a route-schedule $k = \{(l_1, t_1), (l_2, t_2), (l_3, t_3)\}$ in a graph $\mathcal{G}$ with interference graph $\mathcal{G}_I$. The bottom figure shows only part of $\mathcal{G}$ that includes the links of k. The top figure shows the interference subgraph $\mathcal{G}_I[\bigcup_{i=1}^{3} \mathcal{N}_{l_i}]$ and a corresponding clique cover. The cost of k is $q(k) = (b_{Q_1 t_1} + b_{Q_{12} t_1}) + (b_{Q_{12} t_2} + b_{Q_2 t_2} + b_{Q_2'' t_2}) + b_{Q_3 t_3}$. Only the cliques in which a link of k is included contribute to the cost.

Admission and Route-Schedule Assignment. GIMS maintains a *cost*, $b_{Qt} \geq 0$, for each clique $Q \in \mathcal{Q}$ at every time t, which depends on the load on the clique, i.e., the number of packets scheduled for links in Q at time t. A link that belongs to cliques that are not overloaded should be selected more favorably. Formally, define the cost for a route-schedule k as,

$$\mathrm{q}(k) = \sum_{(l,t)\in k} \sum_{Q:l\in Q} b_{Qt}, \tag{5}$$

i.e., the total cost of cliques of links included in the route-schedule at the respective time slots. See Figure 4 for an illustrative example.

Algorithm 2 finds the route-schedule $k^\star$ that has the minimum cost (Line 2.7). We will describe how this minimization can be solved efficiently through an expanded graph in Section 7.

When a packet m arrives, admission decision is made by comparing its reward w_m with the cost of the best route-schedule $q(k^\star)$ (Line 2.9). If $w_m > q(k^\star)$, the packet is admitted and scheduled on the route-schedule $k^\star$, otherwise it is rejected. If the packet is admitted and assigned to $k^\star$, we increase the corresponding clique costs (Line 2.11). Formally, for each $(l, t') \in k^\star$, we increase $b_{Qt'}$ for all cliques in $\mathcal{Q}$ that link l belongs to, i.e.,

$$b_{Qt'} \leftarrow \left(\frac{F}{\mathrm{C}} + 1\right) b_{Qt'} + \frac{w_{min}\phi}{L\mathrm{C}}, \ \forall Q \in \mathcal{Q} : l \in Q. \tag{6}$$

The cost increase depends on the chosen parameters $F \geq 1, \phi > 0$. We refer to GIMS with parameters F, ϕ, as $\mathrm{GIMS}_{F,\phi}$. We will later specify the value of these parameters such that the rate of increase of $b_{Qt'}$ guarantees that the cliques are not overloaded and consequently the capacity constraints are not violated.

Channel Assignment. In contrast to the admission and route-schedule decisions which are made upon arrival of a packet, the channel assignment that is used to transmit the packet is only determined at the time slot that the packet is to be transmitted.

After Line 2.21, we determine the network-schedule for that time t, namely, the collection of independent sets such that the packets decided to be scheduled at time t (denoted by $\bigcup_{l\in\mathcal{L}} S_{lt}$) are transmitted through these independent sets over the corresponding channels. The independent sets are chosen greedily. In every iteration c, we choose a maximal independent set $\mathrm{Y}_t^{\mathrm{c}}$ (Line 2.24) over the links with remaining packets to be transmitted ($\mathcal{L}_{\mathrm{c}}$ at Line 2.23) until all scheduled packets are transmitted. We will later show that the loop at Line 2.21 terminates in at most RC iterations, i.e., we transmit all the scheduled packets using the RC available channels.

Algorithm 2: General Interf. Multihop Scheduler (GIMS)

2.1 **Input:** Parameters $F \geq 1$, $\phi > 0$ and $\mathcal{Q}$.
2.2 $b_{Qt} \leftarrow 0, \forall Q \in \mathcal{Q}, \forall t$, and $S_{lt} \leftarrow \emptyset, \forall l \in \mathcal{L}, \forall t$.
2.3 **for** *each time* $t = 0, 1, \ldots$ **do**
2.5   **for** *each packet arrival m at time t* **do**
2.7 $k^\star \leftarrow \arg\min_{k\in\mathcal{K}_m} \mathrm{q}(k)$
2.9   **if** $\mathrm{q}(k^\star) < w_m$ **then**
2.11 $b_{Qt'} \leftarrow (\frac{F}{\mathrm{C}} + 1) b_{Qt'} + \frac{w_{min}\phi}{L\mathrm{C}}$,
2.12 $\forall(l, t') \in k^\star, \forall Q : l \in Q$
2.13 $S_{lt'} \leftarrow S_{lt'} \cup \{m\}, \forall(l, t') \in k^\star$
2.14 $X_{mk^\star} \leftarrow 1$
2.15   **else**
2.16   Drop packet m
2.17   **end**
2.18   **end**
2.19 $\mathrm{c} \leftarrow 0$
2.21   **while** $\exists l \in \mathcal{L}$ *with* $|S_{lt}| > 0$ **do**
2.22 $\mathrm{c} \leftarrow \mathrm{c} + 1$
2.23 $\mathcal{L}_{\mathrm{c}} \leftarrow \{l \in \mathcal{L} : |S_{lt}| > 0\}$
2.24 $\mathrm{Y}_t^{\mathrm{c}} \leftarrow$ A maximal independent set in $\mathcal{G}_I[\mathcal{L}_{\mathrm{c}}]$.
2.25 $m_{l,\mathrm{c}} \leftarrow$ An arbitrary packet from $S_{lt}, \forall l \in \mathrm{Y}_t^{\mathrm{c}}$
2.26   Schedule $m_{l,\mathrm{c}}$ at time t over matching $\mathrm{Y}_t^{\mathrm{c}}$
2.27 $S_{lt} \leftarrow S_{lt} \setminus \{m_{l,\mathrm{c}}\}, \forall l \in \mathrm{Y}_t^{\mathrm{c}}$
2.28   **end**
2.29 **end**

4.1 Performance Guarantees

The theorem below states the performance of GIMS in the most general form.

Theorem 1. *Consider a clique cover $\mathcal{Q}$ of $\mathcal{G}_I$ with parameters ψ, ξ. Then $\mathrm{GIMS}_{F,\phi}$ is $(F + \xi\phi)$-competitive for the problem P_{WG} (Eq. 2) with $\mathrm{R} = \psi \frac{\log\left(\frac{\rho L F}{\phi}+1\right)}{\mathrm{C}\log(F/\mathrm{C}+1)}$, where $F \geq 1$ and $\phi > 0$.*

Note that letting $\phi = \frac{\phi'}{\xi}$, the algorithm is $(F + \phi')$-competitive with $\mathrm{R} = \psi \frac{\log(\frac{\rho L F \xi}{\phi'}+1)}{\mathrm{C}\log(F/\mathrm{C}+1)}$. Hence, R is linear in ψ while logarithmic in ξ. Therefore, in selecting the clique cover, minimizing ψ should be prioritized (as in Algorithm 1).

The following corollary states the required R to achieve a near-optimal performance.

Corollary 1. *Given a clique cover $\mathcal{Q}$, and any $\epsilon > 0$, $\mathrm{GIMS}_{1,\epsilon/\xi}$ is $(1 + \epsilon)$-competitive for the problem P_{WG} when the resource augmentation is $\mathrm{R} = \psi \frac{\log\left(\frac{\rho L \xi}{\epsilon}+1\right)}{\mathrm{C}\log(1/\mathrm{C}+1)}$.*

Given $\mathcal{Q}$ and R, we can optimize for the competitive ratio by solving the following simple single-variable optimization:

$$\min \quad F + \xi\phi(F) \tag{7a}$$

$$\text{s.t.} \quad F \geq 1 \tag{7b}$$

$$\phi(F) = \frac{L\rho F}{(1 + F/\mathrm{C})^{\mathrm{C}\frac{\mathrm{R}}{\psi}} - 1}. \tag{7c}$$

Let us denote the optimal solution of this optimization problem as $F^\star(\mathrm{R})$ and its corresponding ϕ by $\phi^\star(\mathrm{R}) = \phi(F^\star(\mathrm{R}))$. Further, we denote $\mathrm{GIMS}_\star = \mathrm{GIMS}_{F^\star(\mathrm{R}),\phi^\star(\mathrm{R})}$. We provide a bound on the performance of $\mathrm{GIMS}_\star$ below based on a suboptimal choice of F, ϕ.

Corollary 2. *Given a resource augmentation* $\mathrm{R} \geq 1$, $\mathrm{GIMS}_{F,\phi}$ *is* $\frac{4\psi}{\mathrm{R}}\ln(\xi\rho L + 1)$*-competitive when* $\mathrm{C} \geq \frac{2\psi}{\mathrm{R}}\ln(\xi\rho L + 1)$ *by setting* $F = \frac{2\psi}{\mathrm{R}}\ln(\xi\rho L + 1)$ *and* ϕ *as in (7c).*

The special case of $\mathrm{C} = 1, \mathrm{R} = 1$ is not covered by Corollary 2 as $\mathrm{CR} = 1$. We provide a result for this case in the following Corollary.

Corollary 3. *For* $\mathrm{C} = 1$ *and* $\mathrm{R} = 1$, $\mathrm{GIMS}_{1,\rho L}$ *is* $(1 + \xi\rho L)$*-competitive given an edge clique cover* $\mathcal{Q}$.

Corollary 3 requires an edge clique cover $\mathcal{Q}$. Note that in this case we seek to choose a cover with only a small ξ. For certain graph families, there are known bounds for ξ. For example, in linear interval graphs $\xi = O(\log\Delta)$, where Δ is the maximum degree of any node in the interference graph [22].

Remark 1 (*Relation to interference degree*). As we saw, e.g., in Corollary 2, our results are linear in ψ while logarithmic in ξ, hence primarily impacted by ψ. We connect the optimal $\psi^\star$ with the previously studied notion of the interference degree β [1–3]. It is not hard to show (see our technical report [23]) that, in general,

$$\beta \leq \psi^\star \leq \Delta.$$

For many families of interference graphs however we have $\beta = \psi^\star$, e.g., graphs that are perfect [24] in every neighborhood (i.e., for each $\mathcal{G}_I[\mathcal{N}_l]$). One such case is the graph in Figure 3. In this case $\psi^\star = \beta = 2$. Note that the entire graph is not perfect. In practical graphs, we expect $\psi^\star$ to be close to β. In Section 10, we verify that this is indeed true for random graphs.

5 SCHEDULING UNDER NODE-EXCLUSIVE INTERFERENCE

The general framework described in Section 4 is directly applicable to the special case of node-exclusive interference model. However, in this case we can describe an effective edge clique covering. In fact under node-exclusive interference, all the links incident to a node $v \in \mathcal{V}$ form a valid clique $Q_v = \mathcal{A}(v)$ in $\mathcal{G}_I$. Furthermore, we can locally cover $\mathcal{N}_l$ of link $l = (u, v)$, with $\mathcal{Q}_l = \{\mathcal{A}(u), \mathcal{A}(v)\}$. Hence, $\mathcal{Q} = \{\mathcal{A}(v), v \in \mathcal{V}\}$. Therefore, we can cover the neighborhood of every link in $\mathcal{G}_I$ with exactly two cliques and get $\psi = \xi = 2$. We further simplify our notation for the costs to refer to the cost of nodes instead of cliques, since each clique is associated with a node. Let the costs be $b_{vt} \geq 0$ for every *node* $v \in V$ at every time t. The cost for a route-schedule k is then

$$\mathrm{q}(k) = \sum_{((u,v),t)\in k} (b_{vt} + b_{ut}). \tag{8}$$

We refer to the algorithm for one-hop interference with the node-induced clique cover as NEMS which stands for *Node-Exclusive Multihop Scheduler.* Then Theorem 1 and Corollaries 1, 2 and 3 hold for $\psi = 2, \xi = 2$. To facilitate our discussion, we state the following bound on the performance of $\mathrm{NEMS}_\star$ (defined analogously to $\mathrm{GIMS}_\star$) for the suboptimal choice of F, ϕ in the following corollary.

Corollary 4. *Given a resource augmentation* $\mathrm{R} \geq 1$, $\mathrm{NEMS}_{F,\phi}$ *is* $\frac{8}{\mathrm{R}}\ln(2\rho L + 1)$*-competitive when* $\mathrm{C} \geq \frac{4}{\mathrm{R}}\ln(2\rho L + 1)$ *by setting* $F = \frac{4}{\mathrm{R}}\ln(2\rho L + 1)$ *and* ϕ *as in (7c) for* $\psi = 2$.

6 SCHEDULING WITH NO INTERFERENCE

In this section, we mention the special case of wired networks. This time, we associate a cost $b_{lt} \geq 0$ with every link l at each time t. The cost of a route-schedule k reduces to the sum of the cost of link-time tuples (l, t) in k, i.e.,

$$\mathrm{q}(k) = \sum_{(l,t)\in k} b_{lt}. \tag{9}$$

We refer to the algorithm for this setting as *Weighted Multihop Scheduler* (WEMS). The simplified Algorithm 2 for WEMS is provided in our technical report [23]. Following similar notations of the general algorithm, we define $\mathrm{WEMS}_{F,\phi}$ and $\mathrm{WEMS}_\star$. In this case, Theorem 1 and Corollaries 1, 2 and 3 hold with $\psi = \xi = 1$.

We provide a bound on the performance of $\mathrm{WEMS}_\star$ below based on a suboptimal choice of F, ϕ.

Corollary 5. *Given a resource augmentation* $\mathrm{R} \geq 1$, $\mathrm{WEMS}_{F,\phi}$ *is* $\frac{4}{\mathrm{R}}\ln(\rho L + 1)$*-competitive when* $C_{\min} \geq \frac{2}{\mathrm{R}}\ln(\rho L + 1)$ *by setting* $F = \frac{2}{\mathrm{R}}\ln(\rho L + 1)$ *and* $\phi = \phi(F)$ *as in (7c) for* $\psi = 1$.

Remark 2. As seen in Corollary 5, WEMS is $O(\log(\rho L))$-competitive for $\mathrm{R} = 1$, if $C_{\min} = \Omega(\log(\rho L))$. This is the first logarithmic result for general ρ, improving over [8]. Corollary 5 further improves the result in [7] for the values $\mathrm{R} = o(\log(\rho L))$. For example, for $\mathrm{R} = 2$, Corollary 5 still provides logarithmic competitiveness as opposed to the linear competitiveness of [7]. Finally our result is not asymptotic in $\mathrm{C}_{\min}$ but only requires $\mathrm{C}_{\min}$ to be logarithmic.

7 COMPLEXITY OF ALGORITHMS

In this section, we elaborate on the complexity of the algorithms.

Selecting the clique cover. Recall that we need to find a clique cover of the given interference graph. We discuss the complexity of finding a clique cover through a local method such as in Algorithm 1. For locally perfect graphs, finding an optimal clique cover can be solved in polynomial time [25]. In more general graphs, we can use an algorithm like Eppstein's algorithm [25] to find an optimal clique cover in each neighborhood, yielding a total complexity $O(|\mathcal{L}|2.415^\Delta)$. If $\Delta = O(\log|\mathcal{L}|)$, this complexity is polynomial. In the case of large Δ, when the complexity becomes prohibitive, one can adopt experimentally tested algorithms that are shown to obtain the optimal coloring in most graphs [26] or other greedy techniques [21] (note that a disjoint clique cover of a graph corresponds to the coloring of the complement graph).

Admission and route-schedule selection. Now we discuss the running time of Algorithm 2. We show how the minimization at Line 2.7 can be solved in polynomial time. We will reduce the problem to finding a minimum cost path between two nodes in a

Directed Acyclic Graph (DAG). The minimum path problem can be solved even more efficiently in DAGs, with linear complexity $O(V+E)$ in a DAG (V,E) [27].

Consider the minimization for packet m. We use $T=\{a_m, a_m+1,\cdots,a_m+d_m\}$ to denote the time slots during which packet m can be in the system. Let us construct an *Expanded graph*, over which we will find minimum paths, as follows. Define the set of vertices as $\mathcal{V}\times T$, i.e., each node of the network's graph is copied $|T|=d_m+1$ times. For each link $l=(u,v)\in\mathcal{L}$, we add edges $\big((u,t),(v,t+1)\big)$ if $t\in T$ and $t+1\in T$, with weight $\sum_{Q:l\in Q} b_{Qt}$. Finally, for each node $v\in\mathcal{V}$, we add edges $\big((v,t),(v,t+1)\big)$ if $t\in T$ and $t+1\in T$, with weight 0. It is easy to verify that all paths from (s_m,a_m) to (τ_m, a_m+d_m) correspond to all the route-schedules $\mathcal{K}_m$. Consequently finding the optimal route-schedule is reduced to finding a minimum cost path in the expanded graph. Finding a minimum path and processing a packet in this graph can thus be completed in $O(|\mathcal{L}|d_{max}\xi)$.

Channel assignment. For Algorithm 2, at every time slot, we need to also find at most RC maximal independent sets. Each maximal independent set can be found in $O(\beta|\mathcal{L}|)$ [28], so the complexity per time slot is bounded by $O(\beta|\mathcal{L}|\mathrm{RC})$.

8 PROOFS OF MAIN RESULTS

We provide the proof of Theorem 1. The proofs of Corollary 2 and Corollary 3 can be found in our technical report [23].

8.1 Outline of Proof Technique

Consider a (primal) maximization problem P, such as P_{WG} (Eq. 2) or P_{WM} (Eq. 3). Let P^{μ} denote the objective value of the solution provided by our algorithm μ, and $P^{\star}$ denote the optimal value. Let $P(\mathrm{R})$ denote the augmented optimization problem when $\mathrm{R}\geq 1$ (or the contracted optimization problem when $\mathrm{R}<1$). Note that, in (2) (or (3) and (4)), R appears in the constraints, not in the objective function. To show (1), we first obtain a relaxation $\widetilde{P}$ of the original problem P. Suppose for some $\lambda\in\mathbb{N}$, our algorithm μ finds a feasible solution for the relaxed augmented problem $\widetilde{P}(\mathrm{R}/\lambda)$ that is also feasible for $P(\mathrm{R})$, then it holds that $\widetilde{P}^{\mu}(\mathrm{R}/\lambda)=P^{\mu}(\mathrm{R})$, since the two optimization problems have the same objective function. Then if we show

$$\widetilde{P}^{\mu}(\mathrm{R}/\lambda)\geq\gamma(\mathrm{R})^{-1}\widetilde{P}^{\star},\tag{10}$$

we get

$$P^{\mu}(\mathrm{R})\overset{(a)}{=}\widetilde{P}^{\mu}(\mathrm{R}/\lambda)\overset{(b)}{\geq}\gamma(\mathrm{R})^{-1}\widetilde{P}^{\star}\overset{(c)}{\geq}\gamma(\mathrm{R})^{-1}P^{\star},$$

and hence (1) follows. Here, (a) is due to feasibility of solution of μ for $\widetilde{P}(R/\lambda)$ and $P(R)$, (b) is due to (10), and (c) is the immediate result of the relaxation. Hence, our proof for each main theorem has three steps:

Step 1: We construct $\widetilde{P}$ as a Linear Program (LP).

Step 2: We show (10) through a primal-dual approach. Let $\widetilde{D}$ be the dual of $\widetilde{P}$. Let $\widetilde{D}^{\mu}$ be the dual objective value under algorithm μ and $\widetilde{D}^{\star}$ be the optimal dual value. We construct a primal-dual pair through our algorithm μ such that

$$\widetilde{P}^{\mu}(R/\lambda)\geq\gamma(R)^{-1}\widetilde{D}^{\mu}.\tag{11}$$

Then (10) follows since $\widetilde{D}^{\mu}\geq\widetilde{D}^{\star}=\widetilde{P}^{\star}$ by LP duality.

Step 3: We show that the solution of μ for $\widetilde{P}(R/\lambda)$ is also a feasible solution for $P(R)$.

In the following, for simplicity of exposition and to avoid technicalities we assume in our analysis that R is such that $\frac{\mathrm{RC}}{\psi}$ is an integer, unless RC = 1. As we see in our evaluations in Section 10, this assumption does not have a negative impact on our algorithm.

8.2 Proof of Theorem 1

Step 1. We provide an LP relaxation of $P_{\mathrm{WG}}(R)$, given a clique cover $\mathcal{Q}$ of $\mathcal{G}_I$. We refer to the relaxation as $\widetilde{P}_{\mathrm{WG}}(R)$:

$$\max\quad \sum_{m\in\mathcal{M}} w_m\sum_{k\in\mathcal{K}_m} X_{mk}\qquad (:\widetilde{P}_{\mathrm{WG}}(\mathrm{R}))\tag{12a}$$

$$\text{s.t.}\quad \sum_{l\in Q}\sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{K}_m:(l,t)\in k} X_{mk}\leq\mathrm{RC}\quad\forall Q\in\mathcal{Q},\forall l,\forall t\tag{12b}$$

$$\sum_{k\in\mathcal{K}_m} X_{mk}\leq 1\quad\forall m\in\mathcal{M}\tag{12c}$$

$$X_{mk}\geq 0\quad\forall m\in\mathcal{M},\forall k\in\mathcal{K}_m,\quad Y^{c}_{lt}\geq 0\quad\forall l,t,c\tag{12d}$$

To see that this is indeed a relaxation, we will show that any feasible solution for $P_{\mathrm{WG}}(\mathrm{R})$ is also a feasible solution for $\widetilde{P}_{\mathrm{WG}}(\mathrm{R})$. Consider a particular clique $Q\in\mathcal{Q}$. Constraints (2b), (2f) enforce that each $\mathbf{Y}^{\mathrm{c}}_t$ is an independent set, and consequently, for any clique, we can have at most one link in the clique activated, i.e., $\sum_{l\in Q} Y^{\mathrm{c}}_{lt}\leq 1$ for all $c\in[\mathrm{RC}]$ and for all t. Then it follows

$$\sum_{l\in Q}\sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{K}_m:(l,t)\in k} X_{mk}\overset{(a)}{\leq}\sum_{l\in Q}\sum_{i=1}^{\mathrm{RC}} Y^{i}_{lt}=\sum_{i=1}^{\mathrm{RC}}\sum_{l\in Q} Y^{i}_{lt}\leq\mathrm{RC},$$

where (a) is due to Constraint (2c). Hence, (12b) holds.

Step 2. Next, we show (10) in our case, by proving the following Lemma through a primal-dual technique.

Lemma 2. *Under $\mu=\mathrm{GIMS}_{F,\phi}$, relation (10) is satisfied as follows: $\widetilde{P}^{\mu}_{\mathrm{WG}}(\frac{\mathrm{R}}{\psi})\geq(F+\xi\phi)^{-1}\widetilde{P}^{\star}_{\mathrm{WG}}$ for $\frac{\mathrm{R}}{\psi}=\log_{(F/C+1)^C}\left(\frac{\rho L}{\phi}F+1\right)$.*

Proof. We provide a sketch of the proof here. For the complete proof, refer to our technical report [23].

The following LP is the dual of $\widetilde{P}_{\mathrm{WG}}(1)$:

$$\min_{\alpha_m,b_{Qt}}\quad\sum_{m\in\mathcal{M}}\alpha_m+\sum_{Q,t}\mathrm{C}b_{Qt}\qquad(:\widetilde{D}_{\mathrm{WM}})\tag{13a}$$

$$\text{s.t.}\quad\alpha_m+\sum_{(l,t)\in k}\sum_{Q:l\in Q} b_{Qt}\geq w_m\quad\forall m,\forall k\in\mathcal{K}_m\tag{13b}$$

$$\alpha_m\geq 0,\forall m,\quad b_{Qt}\geq 0,\forall Q,t\tag{13c}$$

Note that in the above, the dual variables b_{Qt} correspond to the clique costs, i.e., our algorithm constructs an appropriate solution for dual variables b_{Qt}. We will also define a solution for α_m. We, then, show (10) by proving that Algorithm 2 yields a primal-dual pair with $\widetilde{P}^{\mu}_{\mathrm{WG}}(\frac{\mathrm{R}}{\psi})\geq(F+\xi\phi)^{-1}\widetilde{D}^{\mu}_{\mathrm{WG}}$.

Consider a given packet m processed by Algorithm 2. Assign to the dual variable α_m the value

$$\alpha_m=\mathbb{1}(\mathrm{q}(k^{\star})<w_m)(w_m-q(k^{\star})),\tag{14}$$

where the values $k^{\star}$ and b_{Qt} used in the evaluation of α_m are the values at the time packet m is processed at Line 2.9.

Satisfying Dual Constraints. The dual constraints are satisfied as a result of the assignment in (14) for each m.

Satisfying Primal Constraints. For the primal constraints we need to make sure that the capacity constraints in Equation 12b are satisfied. After n packet admissions to links in Q we have $b_{Qt}[n] = \frac{w_{min}\phi}{LC}\frac{(\frac{F}{C}+1)^{n}-1}{F/C}$. Further, to admit a packet on a link in a clique Q, it must hold that $b_{Qt}[n] \leq q(k^\star) < w_m \leq w_{max}$. From that inequality on b_{Qt}, we can bound the maximum number of packets n' on any clique. It follows that $n' \leq \mathrm{C}\frac{\mathrm{R}}{\psi}$ for the R stated in the statement of the lemma, i.e., the capacity can be violated by at most a factor of $\frac{\mathrm{R}}{\psi}$.

Primal-Dual Analysis. We finish the proof of the Lemma and thus of Step 2 by showing (11) for $\lambda = \psi$. Let $\Delta\widetilde{P}_{\mathrm{WG}}$ and $\Delta\widetilde{D}_{\mathrm{WG}}$ be respectively the change of the primal objective value and the dual objective value after a packet arrival m has been rejected or admitted and assigned to a route-schedule k. For the nontrivial case of when a packet is admitted, $\Delta\widetilde{P}_{\mathrm{WG}} = w_m$. For the dual we have

$$\Delta\widetilde{D}_{\mathrm{WG}} = w_m - \mathrm{q}(k^\star) + \sum_{(l,t)\in k^\star} C \sum_{Q:l\in Q} \Delta b_{Qt}$$
$$\leq F w_m + \phi\xi w_{min} \leq (F+\phi\xi) w_m.$$

This implies that $\widetilde{D}_{\mathrm{WG}} \leq (F+\xi\phi)\widetilde{P}_{\mathrm{WG}}$ and hence for the relaxation we have a competitive ratio of $(F+\xi\phi)^{-1}$ for augmentation $\frac{\mathrm{R}}{\psi}$. ◻

Step 3. With Lemma 2 in hand, we complete the proof by arguing that the obtained solution is feasible for $P_{\mathrm{WG}}(\mathrm{R})$. To do that we show that Algorithm 2 yields a set of independent sets that satisfy Constraints (2b) and (2c) for each time slot t.

Let Λ_t denote the number of iterations of the loop at Line 2.21 of Algorithm 2 before it terminates for time t. Note that after processing the arrivals at time t (i.e, after exiting the loop at Line 2.5), S_{lt} is the set of packets scheduled to be transmitted on link l at time slot t. Let $S_{lt}(\mathrm{c})$ denote the set S_{lt} at the end of iteration c at Line 2.21, and $S_{lt}(0)$ be the initial set. Note that for $\mathrm{c} = \Lambda_t$, $S_{lt}(\Lambda_t) = \varnothing$, by the condition of Line 2.21 and the definition of Λ_t. We extend the definition of S_{lt} by letting $S_{lt}(\mathrm{c}) = \varnothing$ for $\mathrm{c} \geq \Lambda_t$. Note that initially

$$|S_{lt}(0)| = \sum_{m\in\mathcal{M}} \sum_{k\in\mathcal{K}_m:(l,t)\in k} X_{mk}. \tag{15}$$

In every iteration, a packet is removed from S_{lt} if $Y^{\mathrm{c}}_{lt} = 1$ (Line 2.27). Thus it follows that for $n \in [\Lambda_t]$

$$|S_{lt}(n)| = \sum_{m\in\mathcal{M}} \sum_{k\in\mathcal{K}_m:(l,t)\in k} X_{mk} - \sum_{\mathrm{c}=1}^{n} Y^{\mathrm{c}}_{lt}. \tag{16}$$

By (16) and since $|S_{lt}(\Lambda_t)| = 0$, we have

$$\sum_{m\in\mathcal{M}} \sum_{k\in\mathcal{K}_m:(l,t)\in k} X_{mk} = \sum_{\mathrm{c}=1}^{\Lambda_t} Y^{\mathrm{c}}_{lt}.$$

If we show that $\Lambda_t \leq \mathrm{RC}$, then Constraints (2c) will be satisfied as

$$\sum_{m\in\mathcal{M}} \sum_{k\in\mathcal{K}_m:(l,t)\in k} X_{mk} = \sum_{\mathrm{c}=1}^{\Lambda_t} Y^{\mathrm{c}}_{lt} \leq \sum_{\mathrm{c}=1}^{\mathrm{RC}} Y^{\mathrm{c}}_{lt}.$$

Since we take a maximal independent set at Line 2.24 in every iteration for at most RC iterations, it will also follow that Constraints (2b) are satisfied. Therefore the $(F+\xi\phi)$-competitive solution obtained Lemma 2 for $\widetilde{P}_{\mathrm{WM}}(\mathrm{R}/\psi)$ would be a feasible solution for $P_{\mathrm{WM}}(\mathrm{R})$ and, as we have discussed in Section 8.1, this will imply the theorem.

Thus it remains to show that $\Lambda_t \leq \mathrm{RC}$ or alternatively $|S_{lt}(\mathrm{RC})| = 0$. Let $y_{lt}(\mathrm{c}) = |S_{lt}(\mathrm{c})|$. Note that by Lemma 2, we obtain a solution to $\widetilde{P}_{\mathrm{WG}}$, and hence at the end of slot t it must satisfy

$$\sum_{l\in Q} \sum_{m\in\mathcal{M}} \sum_{k\in\mathcal{K}_m:(l,t)\in k} X_{mk} \leq \mathrm{CR}/\psi,\ \forall Q \in \mathcal{Q},$$

where $\mathrm{R}/\psi = \log_{(F/\mathrm{C}+1)^{\mathrm{C}}}\left(\frac{w_{max}L}{w_{min}\phi}F + 1\right)$. Using (15), we can rewrite these constraints as

$$\sum_{l\in Q} y_{lt}(0) \leq \mathrm{RC}/\psi. \tag{17}$$

Take any link l'. We will show $y_{l't}(\mathrm{RC}) = 0$. To that end, consider the graph $\mathcal{G}_I[\mathcal{L}_{\mathrm{c}}]$ at time t where $\mathcal{L}_{\mathrm{c}}$ is defined at Line 2.23. Recall that the neighborhood of link l' is covered by definition by at most ψ cliques, $\mathcal{Q}_{l'}$. For each such clique there is a constraint in $\widetilde{P}_{\mathrm{WG}}(R/\psi)$ such as in (17). Further, since $y_{lt}(\mathrm{c})$ is non-increasing with c, these constraints can be written as

$$\sum_{l\in Q} y_{lt}(\mathrm{c}) \leq \mathrm{CR}/\psi,\ \forall Q \in \mathcal{Q}_{l'}. \tag{18}$$

Since $\mathcal{N}_{l'} = \cup_{Q\in\mathcal{Q}_{l'}} Q$, the above implies that

$$\sum_{l\in\mathcal{N}_{l'}} y_{lt}(\mathrm{c}) \leq \sum_{Q\in\mathcal{Q}_{l'}} \sum_{l\in Q} y_{lt}(\mathrm{c}) \leq \mathrm{CR}.$$

In every iteration c, while $y_{l't}(\mathrm{c}) > 0$, the value of $\sum_{l\in\mathcal{N}_{l'}} y_{lt}(\mathrm{c})$ is reduced by one. This is because we select a maximal independent set and at least one of the links $\hat{l} \in \mathcal{N}_{l'}$ with $y_{\hat{l}t}(\mathrm{c}) > 0$ has to be included in the independent set, otherwise it is not maximal. Therefore we have $y_{lt'}(\mathrm{RC}) = 0$, or $\sum_{l\in\mathcal{N}_{l'}} y_{lt}(\mathrm{RC}) = 0$ which in turn implies $y_{l't}(\mathrm{RC}) = 0$. Since l' was arbitrary, it follows that for all links $l \in \mathcal{L}$, we have $y_{lt}(\mathrm{RC}) = 0$.

9 LOWER BOUNDS ON COMPETITIVENESS

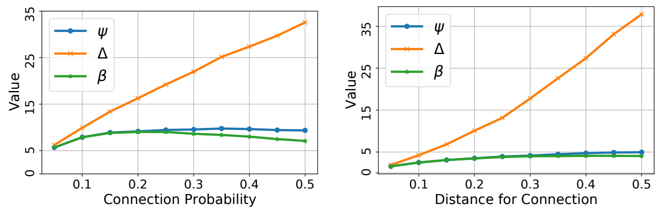
In this section we discuss lower bounds on competitiveness, or equivalently upper bounds on the competitive ratio, of any online algorithm. There are many parameters of interest and hence there is a variety of the lower bounds one can state.

The following result follows directly from the results stated for wireline networks in [6, 8].

Theorem 2. *Any online algorithm, subject to a general interference graph, cannot be better than $\Omega(\max(L, \log(\rho L)))$-competitive with a single channel ($\mathrm{C} = 1$).*

Therefore our algorithms are optimal up to constants for $\mathrm{C} = 1$, $\rho = 1$, for one-hop interference and no-interference networks. It remains an open question whether our online algorithms are also optimal for general ρ and ψ. Further, we remark that inapproximability results for the Maximum Independent Set (MIS) problem [29] apply to our problem P_{WG} as well. Hence, it is not likely (unless $P = NP$) to improve competitiveness significantly over a linear β factor with any polynomial-time algorithm.

Recall that our algorithms are *admission-control* algorithms, i.e., at the time slot a packet arrives, the algorithm determines if the packet should be admitted or not, and if admitted, the packet should



(a) Erdős–Rènyi with 50 nodes. **(b) Random Geometric Graph**

Figure 5: Comparison of greedy ψ, maximum degree Δ and interference degree β, averaged over 100 random networks.

necessarily be scheduled. The following result indicates the optimality of our results in this class in the single channel case for general ρ.

Theorem 3. *Any admission-control algorithm in general interference graphs with any interference degree β cannot be better than $\rho\beta(L-1)$-competitive for a single channel ($\mathrm{C}=1$) and any $L>1$.*

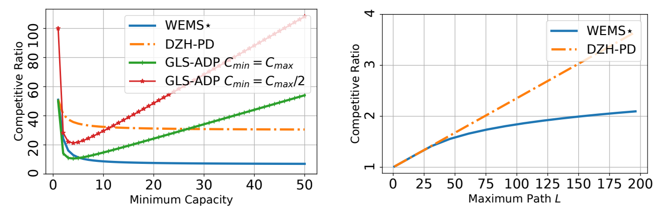
We provide the proof in the technical report [23]. Theorem 3 implies that our algorithms are optimal up to multiplicative constants for $\mathrm{C}=1$ in the class of online admission and scheduling algorithms in all cases where $\beta=\psi^\star$, e.g., in the one-hop interference case, locally perfect graphs and other families.

10 SIMULATION RESULTS

In this section, we provide simulation results to evaluate different aspects of our algorithms and compare with the past algorithms.

Clique Cover Selection. We evaluate the ψ obtained by Algorithm 1 using a locally greedy clique cover [21], on random graphs of different densities. In Figure 5, we compare the obtained ψ with the lower bound β and upper bound Δ. In Figure 5a, we used Erdős–Rènyi graphs of 50 nodes, with an edge probability ranging from 0.05 to 0.5 between every pair of nodes. The results are obtained by averaging over 100 random graphs. In Figure 5b, we used random geometric graphs, where points are thrown in a unit cube, and pairs of points with distance less than the specified threshold are connected. As we observe, the greedily obtained ψ is very close to β and hence to the optimal value $\psi^\star$. Both ψ,β are much smaller than Δ for denser graphs.

Comparing Competitive Ratios in Wired Networks. We illustrate the advantage of the guarantees obtained by WEMS$_\star$ in Figure 6, compared to two past algorithms for wired networks, namely, the PD algorithm from [7] that we refer to as DZH-PD and GLS-ADP algorithm from [8]. As we see in Figure 6a in almost all scenarios the guarantee of WEMS$_\star$ is considerably better than those by [7, 8]. Similar results are obtained for different values of L. In the case of GLS-ADP [8], the competitive ratio depends on the ratio $\mathrm{C}_{\max}/\mathrm{C}_{\min}$. In our plot, we include the ideal case of $\mathrm{C}_{\max}=\mathrm{C}_{\min}$ and the case when the ratio of the two is 2. In our comparison, we use the results of the PD variant from [7] which provides known constants. In Figure 6b, we compare the guarantees for $\mathrm{R}>1$ for various values of L. We observe that for larger L, the gap between WEMS$_\star$ and DZH-PD becomes larger. Finally, we emphasize that our results hold for general ρ in contrast to these works.



(a) For $L=50$ and $R=1$ **(b) For $\mathrm{C}_{\min}=3$ and $R=5$**

Figure 6: Comparison of competitive ratios for $\rho=1$ in wired networks.

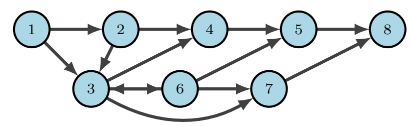
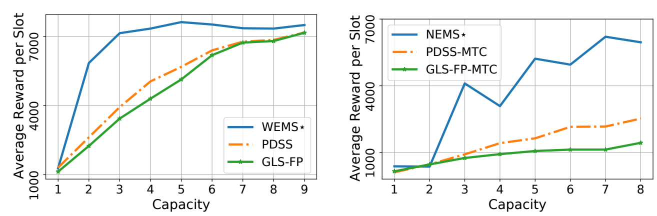


Figure 7: Network topology used in the simulations of Figure 8.



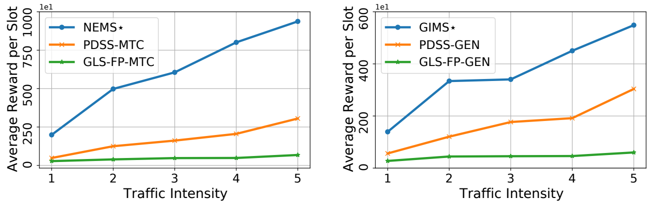
(a) No Interference **(b) One-hop Interference**

Figure 8: Average reward per time slot obtained by various algorithms for the network in Figure 7.

Wired Multihop Networks. We simulate our algorithms and compare them with prior algorithms in the network in Figure 7. We simulated the following flows: $(1,5),(1,8),(2,7),(6,7),(2,5)$ with i.i.d. weights, deadlines and number of arrivals (refer to [23] for details). We observe that the average reward obtained from our Algorithm NEMS$_\star$ is significantly higher than PDSS and GLS-FP (we used the PDSS variant of [7] as it performed better in our simulations, and GLS-FP which is the variant used in the simulations of [8] for multiple routes). We simulated many other topologies, such as the grid topology, and other traffic patterns, and observed similar results [23].

Wireless Multihop Networks. Since the algorithms in [7, 8] are for wired networks, in our evaluation, we extend these policies to our wireless setting as follows: Apply the regular wired policy in these works. To assign packets to channels, apply the packet assignment to channels implemented in our algorithm. When a packet admitted and scheduled for an (l,t) cannot be allocated to one of the C available channels on that link, we drop the packet.

First, we consider the network in Figure 7 under one-hop interference, and traffic as in our simulations for the wired multihop network. This allows us to observe the effect of interference on the performance. The results are presented in Figure 8b. NEMS$_\star$ significantly outperforms the other policies for $\mathrm{C}\geq 3$.



(a) One-Hop Interference **(b) More General Interference**

Figure 9: Average reward per time slot obtained by various algorithms for a random geometric wireless network, for different traffic intensities.

We further evaluated the algorithms on random geometric networks with one-hop interference and more general interference. The network is formed by distributing 25 nodes uniformly at random in a unit square and connecting nodes that have a distance less than 0.2. We kept the largest connected component of this graph, which included 20 nodes. We generated traffic by selecting 10 random pairs of nodes, and assigning a random traffic flow to that pair from a set of 4 different traffic sources. We further varied the number of flows to 20, 30, 40, and 50 (traffic intensity of 1 to 5). The results are presented in Figure 9 under one-hop interference. For these results, $C = 10$ channels were used, however we obtained analogous results for $C = 5, 15, 20$.

To evaluate a more general interference model, in the random geometric graph, each link (u, v) was assigned a "location" as the midpoint of the locations of u and v. Any two links in distance less than 0.4 are then selected to interfere with each other. The resulting interference graph has 185 edges, $\Delta = 20$, and a greedily selected $\psi = 3$. Before pruning $\xi = \Delta + 1 = 21$. After a simple pruning of removing the redundant subset cliques we obtained $\xi = 16$. The average reward performance is shown in Figure 9b. As expected, there is a small reduction in performance compared to Figure 9a. However our algorithms preserve good performance overall, validating our theoretical results. For details on the simulation parameters, refer to our technical report [23]. In our simulations, we verified that all packets admitted by NEMS$_\star$ and GIMS$_\star$ are successfully allocated to the C available channels, as predicted by our analysis.

11 CONCLUSION

In this paper, we introduced a framework for scheduling packets with end-to-end deadlines in multihop wireless networks with general interference graphs, an important problem that has been largely unexplored due to its difficulty. Our framework is directly applicable to special cases such as one-hop interference models, or even wired networks where the obtained competitive ratios outperform prior results. Following our work, many interesting problems arise. Tightening the lower bounds on competitiveness in general regimes could be an interesting future work. Moreover, investigating alternate relaxations can potentially yield improved results.

ACKNOWLEDGMENTS

This work was supported in part by NSF grant CNS-1652115 and ARO grant W911NF1910379.

REFERENCES

[1] X. Kang, W. Wang, J. J. Jaramillo, and L. Ying, "On the performance of largest-deficit-first for scheduling real-time traffic in wireless networks," *IEEE/ACM Transactions on Networking*, vol. 24, no. 1, pp. 72–84, 2014.
[2] C. Tsanikidis and J. Ghaderi, "On the power of randomization for scheduling real-time traffic in wireless networks," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 59–68.
[3] ——, "Randomized scheduling of real-time traffic in wireless networks over fading channels," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021.
[4] I.-H. Hou and P. Kumar, "Utility maximization for delay constrained qos in wireless," in *2010 Proceedings IEEE INFOCOM*. IEEE, 2010, pp. 1–9.
[5] R. Singh, I.-H. Hou, and P. Kumar, "Pathwise performance of debt based policies for wireless networks with hard delay constraints," in *52nd IEEE Conference on Decision and Control*. IEEE, 2013, pp. 7838–7843.
[6] Z. Mao, C. E. Koksal, and N. B. Shroff, "Optimal online scheduling with arbitrary hard deadlines in multihop communication networks," *IEEE/ACM Transactions on Networking*, vol. 24, no. 1, pp. 177–189, 2014.
[7] H. Deng, T. Zhao, and I.-H. Hou, "Online routing and scheduling with capacity redundancy for timely delivery guarantees in multihop networks," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1258–1271, 2019.
[8] Y. Gu, B. Liu, and X. Shen, "Asymptotically optimal online scheduling with arbitrary hard deadlines in multi-hop communication networks," *IEEE/ACM Transactions on Networking*, 2021.
[9] R. Singh, P. Kumar, and E. Modiano, "Throughput optimal decentralized scheduling of multi-hop networks with end-to-end deadline constraints: II wireless networks with interference," *arXiv preprint arXiv:1709.01672*, 2017.
[10] F. Y. Chin, M. Chrobak, S. P. Fung, W. Jawor, J. Sgall, and T. Tichỳ, "Online competitive algorithms for maximizing weighted throughput of unit jobs," *Journal of Discrete Algorithms*, vol. 4, no. 2, pp. 255–276, 2006.
[11] Ł. Jeż, "A universal randomized packet scheduling algorithm," *Algorithmica*, vol. 67, no. 4, pp. 498–515, 2013.
[12] R. Li and A. Eryilmaz, "Scheduling for end-to-end deadline-constrained traffic with reliability requirements in multihop networks," *IEEE/ACM Transactions on Networking*, vol. 20, no. 5, pp. 1649–1662, 2012.
[13] H. Li, Y. Cheng, C. Zhou, and W. Zhuang, "Minimizing end-to-end delay: A novel routing metric for multi-radio wireless mesh networks," in *IEEE INFOCOM 2009*. IEEE, 2009, pp. 46–54.
[14] X. Liu, W. Wang, and L. Ying, "Spatial–temporal routing for supporting end-to-end hard deadlines in multi-hop networks," *Performance Evaluation*, vol. 135, p. 102007, 2019.
[15] Q. Wang, P. Fan, D. O. Wu, and K. B. Letaief, "End-to-end delay constrained routing and scheduling for wireless sensor networks," in *2011 IEEE International Conference on Communications (ICC)*. IEEE, 2011, pp. 1–5.
[16] N. Buchbinder and J. Naor, *The design of competitive online algorithms via a primal-dual approach*. Now Publishers Inc, 2009.
[17] C. Joo, X. Lin, and N. B. Shroff, "Greedy maximal matching: Performance limits for arbitrary network graphs under the node-exclusive interference model," *IEEE Transactions on Automatic Control*, vol. 54, no. 12, pp. 2734–2744, 2009.
[18] B. Hajek and G. Sasaki, "Link scheduling in polynomial time," *IEEE transactions on Information Theory*, vol. 34, no. 5, pp. 910–917, 1988.
[19] E. Modiano, D. Shah, and G. Zussman, "Maximizing throughput in wireless networks via gossiping," *ACM SIGMETRICS Performance evaluation review*, vol. 34, no. 1, pp. 27–38, 2006.
[20] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, "Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks," *IEEE INFOCOM 20016– IEEE Conference on Computer Communications*, 2006.
[21] A. Kosowski and K. Manuszewski, "Classical coloring of graphs," *Contemporary Mathematics*, vol. 352, pp. 1–20, 2004.
[22] R. Javadi, Z. Maleki, and B. Omoomi, "Local clique covering of graphs," *arXiv preprint arXiv:1210.6965*, 2012.
[23] C. Tsanikidis and J. Ghaderi, "Online scheduling and routing with end-to-end deadline constraints in multihop wireless networks- technical report," https://tinyurl.com/mobihoc-technical-report, 2022.
[24] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas, "The strong perfect graph theorem," *Annals of mathematics*, pp. 51–229, 2006.
[25] A. M. de Lima and R. Carmo, "Exact algorithms for the graph coloring problem," *Revista de Informática Teórica e Aplicada*, vol. 25, no. 4, pp. 57–73, 2018.
[26] P. San Segundo, "A new dsatur-based algorithm for exact vertex coloring," *Computers & Operations Research*, vol. 39, no. 7, pp. 1724–1733, 2012.
[27] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2009.
[28] S. Assadi and S. Solomon, "When algorithms for maximal independent set and maximal matching run in sublinear-time," *arXiv preprint arXiv:2006.07628*, 2020.
[29] S. O. Chan, "Approximation resistance from pairwise-independent subgroups," *Journal of the ACM (JACM)*, vol. 63, no. 3, pp. 1–32, 2016.