Long-Term Wireless Link Scheduling with State-Augmented Graph Neural Networks

Romina Garcia Camargo
University of Pennsylvania
rominag@seas.upenn.edu

Zhiyang Wang
University of California San Diego
zhw135@ucsd.edu

Navid NaderiAlizadeh
Duke University
navid.naderi@duke.edu

Alejandro Ribeiro
University of Pennsylvania
aribeiro@seas.upenn.edu

Abstract

We address the wireless link scheduling problem in large-scale wireless networks, subject to a minimum transmission requirement for each link. Unlike traditional approaches that aim to maximize instantaneous sum rate, we focus on maximizing long-term average performance, which better reflects the dynamic behavior of real-world networks. To this end, we formulate a constrained optimization problem that we solve by operating in the Lagrangian dual domain. The communication network is modeled as an undirected graph, from which we derive a conflict graph under a primary interference model, motivating the use of Graph Neural Networks (GNNs) to parameterize the scheduling policy. Since optimal scheduling decisions are inherently time-dependent when optimizing time averages, and GNNs are deterministic models, we incorporate state augmentation to handle the stochastic nature of the task. This augmentation enables the GNN to adapt scheduling decisions over time, balancing constraint satisfaction with performance maximization. We validate our approach through extensive numerical simulations, benchmarking against several baselines.

1 Introduction

The growing complexity and expanding application domains of large-scale wireless communication networks have underscored the importance of efficient resource management strategies. A central challenge is wireless link scheduling—determining which links should transmit and when. This problem plays a pivotal role in ensuring fair resource allocation, mitigating interference, and maintaining the quality of service (QoS) for end-users, ultimately shaping overall network performance.

The relevance of link scheduling has led to a wide range of proposed solutions, spanning information-theoretic, optimization-based, and learning-driven approaches [1, 2, 3, 4, 5, 6]. Algorithms from the information-theoretic line of work [1, 2] typically schedule links sequentially, ensuring that newly scheduled links do not interfere with those already selected. In contrast, [3] introduces a fractional programming-based method that achieves state-of-the-art performance. A common trend across recent learning-based approaches is the use of node geographic positions to inform scheduling decisions [4, 5, 6]. For instance, [5] constructs graph embeddings based on link distances, while [6] exploits the local geometric structure of wireless networks through Riemannian manifolds.

Communication networks naturally exhibit an underlying graph structure that facilitates their modeling and analysis. A common strategy for addressing link scheduling is to represent interference

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: AI and ML for Next-Generation Wireless Communications and Networking (AI4NextG).

using the conflict graph of the communication network, particularly under the primary interference model [7]. In this setting, the scheduling problem reduces to finding a Maximum Independent Set (MIS) of non-interfering links—a well-known NP-hard problem [8, 9]. Recent efforts, such as [10], focus on solving the scheduling task through efficient MIS computation, with frameworks designed to accelerate the discovery of viable transmission sets.

This work introduces a learning-based approach for link scheduling in large-scale wireless networks, designed to maximize the long-term average sum rate. Our formulation ensures a minimum transmission requirement for each link, thereby guaranteeing sustained performance across the network. In contrast to prior work, we shift the focus from instantaneous metrics to time-averaged objectives to better reflect the variability and fairness demands of dynamic network environments. The central distinction of our method lies in the incorporation of temporal dependence into the scheduling policy. We cast the problem as a constrained optimization task and address it through the Lagrangian dual framework. To accommodate the stochastic nature of the setting, we incorporate state augmentation, enabling otherwise static models to adapt over time [11]. A Graph Neural Network (GNN) is used to parameterize our policy, leveraging the underlying graph topology. We validate our method through extensive numerical experiments, demonstrating strong performance and generalization across a range of scenarios.

This article is structured as follows. In Section 2, we formulate the wireless link scheduling problem with time-averaged objective and constraints. The state augmentation strategy is introduced in Section 3, as well as the proposed algorithm and details on the architecture. We characterize the performance of our algorithm in Section 4 with several numerical experiments. In Section 5, we conclude with final thoughts and lines for future work.

2 A time-averaged formulation of link scheduling

We consider a device-to-device (D2D) network with K links. Let $s_i \in \{0, 1\}$ denote the status of link $i \in \{1, 2, \dots, K\}$, where $s_i = 1$ indicates that the link i is scheduled to transmit. Denote the scheduling status of all links by the vector $\mathbf{s} = [s_1; s_2; \cdots; s_K] \in \{0, 1\}^K$. Following the primary interference model, we define two links interfere each other if they share a common device. The interference patterns can be summarized in the conflict graph $\mathbf{G}(\mathcal{V}, \mathcal{E})$, where the node set $\mathcal{V} = \{1, 2, \dots, K\}$ corresponds to the links, and the edge set $\mathcal{E}$ includes (i, j) whenever links i and j interfere. The unweighted adjacency matrix $\mathbf{A} \in \{0, 1\}^{K \times K}$ encodes the structure of this graph. We define the per-link rate $\mathbf{r_0}$ as follows.

$$\mathbf{r_0}(\mathbf{s}) = \mathbf{s} \odot [\mathbf{1} - \mathbf{A}\mathbf{s}]_+, \tag{1}$$

where $\mathbf{1}$ denotes a vector with ones in its K entries, $\odot$ is the element-wise product and $[\cdot]_+$ is the projection on the positive orthant. The vector term $[\mathbf{1} - \mathbf{A}\mathbf{s}]_+$ encodes the collisions resulting from the scheduling decisions $\mathbf{s}$, where a value of 1 in the i-th entry indicates that link i is scheduled without conflict. The vector $\mathbf{r}_0$, therefore, indicates which transmissions are successful. Denote with r the sum of all successful transmissions (i.e., those that do not collide):

$$r(\mathbf{s}) = \mathbf{1}^\mathsf{T}\mathbf{r_0}(\mathbf{s}) = \mathbf{s}^\mathsf{T}[\mathbf{1} - \mathbf{A}\mathbf{s}]_+. \tag{2}$$

Let us consider time steps $t \in \{0, 1, ..., T-1\}$ and add the time dependence to our scheduling vector, $\mathbf{s}(t)$. The objective of wireless link scheduling is to design a policy that satisfies the interference constraints dictated by the network topology, while maximizing the time average of the sum rate r. We restrict our attention to scheduling sequences $\{\mathbf{s}(t)\}_{t=0}^{T-1}$ that meet a minimum transmission requirement for each link, specified by the vector $\mathbf{\Delta} \in \mathbb{R}_+^K$, such that $\mathbf{r_0} \geq \mathbf{\Delta}$. With these definitions, the optimal scheduling problem can be formulated as the following constrained optimization problem:

$$\begin{aligned} \{\mathbf{s}^*(t)\}_{t=0}^{T-1} = \underset{\mathbf{s}(t) \in \{0,1\}^K}{\arg\max} \quad & \frac{1}{T} \sum_{t=0}^{T-1} \mathbf{s}^\mathsf{T}(t)[\mathbf{1} - \mathbf{A}\mathbf{s}(t)]_+ \\ \text{s.t.} \quad & \frac{1}{T} \sum_{t=0}^{T-1} \mathbf{s}(t) \odot [\mathbf{1} - \mathbf{A}\mathbf{s}(t)]_+ \geq \mathbf{\Delta}. \end{aligned} \tag{3}$$

We seek an optimal policy that maps the network's conflict structure to scheduling decisions, defined as $\mathbf{s}(t) = \mathbf{\Phi}(\mathbf{A}, \mathbf{H})$, where $\mathbf{H}$ are learnable parameters. This formulation provides a tractable and

controllable solution space. However, since the policy $\mathbf{\Phi}$ is deterministic, it produces the same scheduling decision at each time step for a given static graph. This lack of variability is misaligned with our goal of maximizing long-term average performance, which requires diversity in scheduling over time. To introduce the necessary stochasticity, we employ a state-augmentation approach, which is described in the following section.

3 State augmentation for non-deterministic policies

We reformulate the constrained optimization problem presented in Eq. (3) in the Lagrangian dual domain. We incorporate a non-negative dual variable vector $\boldsymbol{\lambda} \in \mathbb{R}_+^K$ to penalize violations of the minimum transmission requirement, introducing the constraint in the newly-defined objective. The Lagrangian function is constructed as follows:

$$\overline{\mathcal{L}}\left(\{\mathbf{s}(t)\}_{t=0}^{T-1}, \boldsymbol{\lambda}\right) = \frac{1}{T}\sum_{t=0}^{T-1}\left(\mathbf{s}^{\mathsf{T}}(t)[\mathbf{1}_K - \mathbf{A}\mathbf{s}(t)]_+\right) + \boldsymbol{\lambda}^{\mathsf{T}}\left(\frac{1}{T}\sum_{t=0}^{T-1}\mathbf{s}(t)\odot[\mathbf{1}_K - \mathbf{A}\mathbf{s}(t)]_+ - \boldsymbol{\Delta}\right). \tag{4}$$

We are interested in finding $\{\mathbf{s}^*(t)\}_{t=0}^{T-1}$ the scheduling decisions for all time steps. From now on, we omit the notation for $t = 0, 1, ..., T-1$ for clarity. With the maximization of $\overline{\mathcal{L}}$, we can obtain the following:

$$\mathbf{s}^\dagger(t, \boldsymbol{\lambda}) = \underset{\mathbf{s}(t)\in\{0,1\}^K}{\arg\max}\ \overline{\mathcal{L}}\left(\mathbf{s}(t), \boldsymbol{\lambda}\right). \tag{5}$$

Furthermore, we can define the dual function $g(\boldsymbol{\lambda})$ as the maximization of $\overline{\mathcal{L}}$ for different values of $\boldsymbol{\lambda}$. The global optimum for the dual variable $\boldsymbol{\lambda}^*$ can be found by minimizing $g(\boldsymbol{\lambda})$, i.e.,

$$g(\boldsymbol{\lambda}) = \max_{\mathbf{s}(t)\in\{0,1\}^K}\overline{\mathcal{L}}\left(\mathbf{s}(t), \boldsymbol{\lambda}\right), \quad \boldsymbol{\lambda}^* = \underset{\boldsymbol{\lambda}\in\mathbb{R}_+^K}{\arg\min}\, g(\boldsymbol{\lambda}). \tag{6}$$

The non-convexity of the problem in Eq. (3) results in a positive duality gap. Consequently, the optimal scheduling decisions $\mathbf{s}^*(t)$ are not guaranteed to be recovered through iterative primal gradient ascent and dual gradient descent, but they are contained within the set of solutions obtained by this procedure.

$$\mathbf{s}^*(t) \subseteq \mathbf{s}^\dagger(t, \boldsymbol{\lambda}^*). \tag{7}$$

Additionally, as the Lagrangian can be optimized for each time step independently as $\mathcal{L}$ (see Appendix A), the resulting scheduling policy is time invariant, such that for all t, t' we have $\mathbf{s}^\dagger(t, \boldsymbol{\lambda}) = \mathbf{s}^\dagger(t', \boldsymbol{\lambda})$. This is not true for the optimal policy $\mathbf{s}^*(t)$. Nonetheless, since the dual is always convex, we can recover the optimal dual variables through dual gradient descent, where $\eta_{\boldsymbol{\lambda}}$ is the learning rate:

$$\boldsymbol{\lambda}_{k+1} = \left[\boldsymbol{\lambda}_k - \eta_{\boldsymbol{\lambda}}(\mathbf{s}(t, \boldsymbol{\lambda_k})\odot[\mathbf{1} - \mathbf{A}\mathbf{s}(t, \boldsymbol{\lambda_k})]_+ - \boldsymbol{\Delta})\right]_+. \tag{8}$$

Our formulation is such that $\boldsymbol{\lambda_k} \xrightarrow[k\to\infty]{} \boldsymbol{\lambda}^*$, but this does not imply that $\mathbf{s}^\dagger(t, \boldsymbol{\lambda}_k) \xrightarrow[k\to\infty]{} \mathbf{s}^*(t)$. Theoretical results show that the proposed primal-dual scheduling policy is both almost surely feasible and near-optimal when applied over a sufficiently large number of iterations κ [11, 12].

$$\lim_{\kappa\to\infty}\frac{1}{\kappa}\sum_{k=1}^{\kappa}\frac{1}{T}\sum_{t=0}^{T-1}\mathbf{s}^{\mathsf{T}}(t, \boldsymbol{\lambda_k})[\mathbf{1} - \mathbf{A}\mathbf{s}(t, \boldsymbol{\lambda_k})]_+ \geq P^* - \mathcal{O}(\eta_{\boldsymbol{\lambda}}) \tag{9}$$

$$\lim_{\kappa\to\infty}\frac{1}{\kappa}\sum_{k=1}^{\kappa}\frac{1}{T}\sum_{t=0}^{T-1}\mathbf{s}(t, \boldsymbol{\lambda_k})\odot[\mathbf{1} - \mathbf{A}\mathbf{s}(t, \boldsymbol{\lambda_k})]_+ \geq \boldsymbol{\Delta} \qquad a.s., \tag{10}$$

where P^* is the optimal average sum rate obtained with $\{s^*(t)\}_{t=0}^{T-1}$. The resulting scheduling policy is feasible and achieves performance within a constant additive gap of the optimum. Nonetheless, even with infinite iterations, we will not achieve the optimal policy as the function is not concave in $\mathbf{s}(t, \boldsymbol{\lambda_k})$. Moreover, the abrupt variations of the dual variables across iterations k induce equally abrupt changes in the scheduling policy, preventing stable training of a learning-based parameterization.

To overcome the previous limitations, we augment the scheduling policy with a corresponding set of dual variables $\boldsymbol{\lambda}$. This allows the policy to incorporate dynamic inputs that reflect constraint violations at each time step [11]. We parameterize the network policy using a GNN, which enables tractable and controllable policy representations.

Graph Neural Networks Graph Neural Networks consist of a cascade of layers, each comprising a graph convolutional filter followed by a pointwise nonlinearity [13, 14]. These filters are polynomials on a matrix representation of the graph, aggregating information from neighboring nodes. Formally, the graph convolutional filter at the l-th layer, with input graph signal $\mathbf{x}_{l-1}$, is expressed as follows:

$$\mathbf{y}_l = \sum_{k=0}^{K-1} h_{lk} \mathbf{A}^k \mathbf{x}_{l-1}, \qquad \mathbf{x}_{l-1}, \mathbf{y}_l \in \mathbb{R}^K, \tag{11}$$

where $\{h_{lk}\}_{k=0}^{K-1}$ are the learnable parameters of the filter in the l-th layer. The output of the graph convolutional filter is then passed through a pointwise nonlinearity $\sigma : \mathbb{R} \to \mathbb{R}$, resulting in $\mathbf{x}_l = \sigma(\mathbf{y}_l)$. GNNs possess several desirable properties that make them well-suited for real-world applications, such as permutation equivariance and transferability on and across scale [15, 16, 17, 18, 19]. To leverage these advantages, we parameterize our scheduling policy with a GNN. We denote the model $\mathbf{\Phi}(\mathbf{A}, \mathbf{H}; \mathbf{x})$, with the set of learnable parameters written as $\mathbf{H} \in \mathcal{H}$ and $\mathbf{x}$ as the input graph signal.

3.1 State-augmented GNN (SAGNN) algorithm

We incorporate the dual variable $\boldsymbol{\lambda}$ as the input graph signal to our policy, effectively *augmenting* it to the form $\mathbf{\Phi}(\mathbf{A}, \mathbf{H}; \boldsymbol{\lambda})$. This introduces a notion of time into the solution, as the dual variable evolves over the iterations k, and the GNN is able to capture the stochastic nature of the problem. We rewrite the augmented Lagrangian with a set of dual variables $\boldsymbol{\lambda} \in \mathbb{R}_+^K$ as follows.

$$\mathcal{L}_{\boldsymbol{\lambda}}(\mathbf{H}) = \mathbf{\Phi}(\mathbf{A}, \mathbf{H}; \boldsymbol{\lambda})^{\mathsf{T}} [\mathbf{1} - \mathbf{A}\mathbf{\Phi}(\mathbf{A}, \mathbf{H}; \boldsymbol{\lambda})]_+ + \boldsymbol{\lambda}^{\mathsf{T}} \left(\mathbf{\Phi}(\mathbf{A}, \mathbf{H}; \boldsymbol{\lambda}) \odot ([\mathbf{1} - \mathbf{A}\mathbf{\Phi}(\mathbf{A}, \mathbf{H}; \boldsymbol{\lambda})]_+ - \boldsymbol{\Delta}) \right). \tag{12}$$

With this new definition, we can rewrite Equation (5).

$$\mathbf{H}^* = \arg\max_{\mathbf{H} \in \mathcal{H}} \mathbb{E}_{\boldsymbol{\lambda} \sim p_{\boldsymbol{\lambda}}} [\mathcal{L}_{\boldsymbol{\lambda}}(\mathbf{H})]. \tag{13}$$

During training, our algorithm optimizes the model parameters using randomly sampled vectors $\boldsymbol{\lambda} \sim p_{\boldsymbol{\lambda}}$, where $p_{\boldsymbol{\lambda}}$ denotes the probability distribution of the dual variable. We employ gradient ascent to solve the maximization of $\mathcal{L}_{\boldsymbol{\lambda}}$, iterating over training epochs using a learning rate $\eta_{\mathbf{H}}$. This algorithm results in optimal parameters $\mathbf{H}^*$ for the state-augmented policy in (13). At execution time, the trained model is used to compute the scheduling policy for the current $\boldsymbol{\lambda}_k$ and the dual variable is then updated:

$$\boldsymbol{\lambda}_{k+1} = \left[\boldsymbol{\lambda}_k - \eta_{\boldsymbol{\lambda}} (\mathbf{\Phi}^{\mathsf{T}}(\mathbf{A}, \mathbf{H}^*; \boldsymbol{\lambda}_k) \odot ([\mathbf{1} - \mathbf{A}\mathbf{\Phi}(\mathbf{A}, \mathbf{H}^*; \boldsymbol{\lambda}_k)]_+ - \boldsymbol{\Delta})) \right]_+. \tag{14}$$

This iterative process yields the optimal dual variable $\boldsymbol{\lambda}^*$ over time. Constraint satisfaction is monitored through the dynamics of the dual variables: when the scheduling decisions at a given time step help meet the transmission requirements, the corresponding dual variables decrease. Conversely, if constraints are violated, the dual variables increase. This feedback mechanism steers the scheduling policy toward feasible solutions over time. The training and execution procedures are detailed in Appendix B.

4 Numerical experiments

We conduct an extensive set of experiments to characterize the algorithm's performance.[1] The evaluation focuses on constraint satisfaction under varying transmission requirements $\boldsymbol{\Delta}$, and the achieved time-averaged sum-rates $\overline{r}$. We assign the same minimum transmission requirement to all links, i.e., $\boldsymbol{\Delta} = \Delta \mathbf{1}$. Choosing an appropriate value for Δ is critical. To guide this choice, we use the average degree of the conflict graph as a proxy for network interference, which provides a rough upper bound on feasible scheduling capacity [1].

[1]The code used for the experiments can be found in `https://github.com/romm32/SAGNN`.

Dataset generation Wireless networks can be naturally modeled and analyzed using graph-based representations. Among these, Random Geometric Graphs (RGGs) offer a powerful and intuitive framework. In an RGG, nodes are uniformly distributed over a 2D area, and an edge is drawn between any pair of nodes separated by a distance less than a specified communication threshold d_c. This structure can be viewed as a noisy variant of a grid graph, where nodes originally placed on a regular lattice are perturbed by Gaussian noise. We generate RGGs by perturbing grid graphs and compute the corresponding conflict graphs, which model wireless interference. This approach provides controllability in network complexity. Three sets of 100 graphs each are used for training, validation and testing. Each graph contains approximately $K \simeq 500$ links on average.

Experimental setup We use a GNN architecture with $L = 3$ layers, where the convolutional stages use TagConv filters [20] of order 3, followed by a leaky ReLU activation as the pointwise nonlinearity. A sigmoid activation is applied at the output to produce continuous values in the range $[0, 1]$ that allow propagation of gradients. Since scheduling decisions are binary, a threshold of 0.5 is used at evaluation time to determine which links are scheduled for transmission. The GNN parameters are optimized using the Adam optimizer [21] solve (13), with a learning rate of 5×10^{-5}. Evaluation is carried out considering $T = 200$ time steps. The minimum transmission requirement varies in $\Delta = \{0.1, 0.125, 0.15\}$ for different models, with the dual learning rate set to 2. Further details can be seen in Appendix B.

Baselines Several baselines are considered for comparison. A naive p-persistent algorithm determines a probability of transmission for each link, correlated to its degree of conflict, and samples Bernoulli trials according to these probabilities. A collision avoidance (CA) version further observes which conflicting links have been scheduled and chooses one to turn off. FPlinQ [3] is a state-of-the-art baseline for link scheduling based on fractional programming. It maximizes the instantaneous sum rate. Finally, the Maximum Weight Independent Set (MWIS) algorithm is the optimal solution to the problem, where the links are weighted by the dual variable vector.

4.1 Basic performance of the SAGNN algorithm

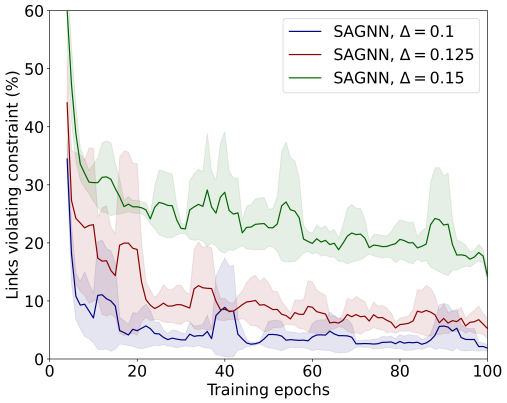
Figure 1a shows the evolution of constraint violations on the validation dataset, specifically the percentage of links for which $\overline{\mathbf{r_0}} < \boldsymbol{\Delta}$, where $\overline{\mathbf{r_0}}$ represents the time-averaged rates achieved per link. For each value of Δ, we report the mean and standard deviation across three independently trained models, applying a running average with a window of 5 for smoother visualization. As expected, higher values of Δ increase the problem's difficulty, approaching the boundary of feasibility. The results indicate that the algorithm learns to effectively schedule links, ensuring a fair share of channel access time. Nevertheless, a small fraction of links fail to meet the minimum transmission requirement.

To assess the severity of the remaining constraint violations, we analyze the violation magnitude $\boldsymbol{\Delta} - \overline{\mathbf{r_0}}$. We present in Figure 1b the distribution of these violation levels, normalized such that a value of 1 corresponds to a complete shortfall. This normalization enables consistent comparison across different values of Δ. The results show that, across all transmission requirements, the vast majority of violations are minor—typically below 15%—with only a few outliers. These findings provide strong evidence that the proposed algorithm effectively learns to satisfy the transmission constraints. Moreover, the incorporation of resilience into the model ensures that links at risk of violating constraints still receive high transmission rates, rather than being neglected due to potential infeasibility [22, 23].

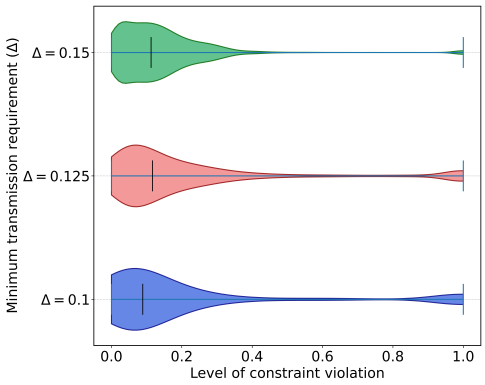
4.2 Comparison against baselines

Table 1 presents the performance of our algorithm alongside baseline methods, averaged over $T = 200$ time steps on 10 graphs from the testing dataset.[2] Our algorithm achieves the lowest constraint violation, demonstrating effective and fair resource allocation across all links. Although it does not attain the highest average sum rate, it remains competitive while offering inference runtimes that are orders of magnitude faster (see Appendix C).

[2]Only 10 graphs are used due to the high computational cost of running the MWIS benchmark over 200 time steps per graph.



(a) Constraint violation in percentage of links.



(b) Normalized distribution of violation levels.

Figure 1: (a): Constraint violation during training averaged across 3 experiments, evaluating on the validation dataset. (b): Level of violation for the testing dataset, normalized by Δ.

Table 1: Comparison against baselines. Mean and standard deviation across 10 graphs.

	Constraint violation (%)	Objective function (%)
p-persistent	97.57±0.62	2.69±0.08
p-persistent CA	85.47±1.44	6.27±0.14
FPLinQ	74.91±0.75	25.09±0.75
MWIS	1.55±0.34	**27.12±0.21**
SAGNN	**0.75±0.45**	21.91±0.26

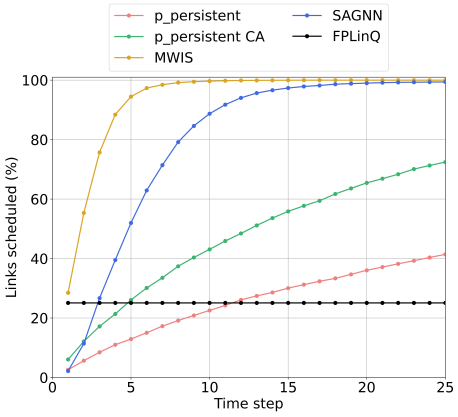


Figure 2: Percentage of links scheduled at least once during the first 25 evaluation steps (averaged over 10 test graphs).

We show in Figure 2 the evolution of the percentage of links scheduled at least once during the initial evaluation steps. SAGNN rapidly approaches the performance of strong baselines such as MWIS. Unlike FPLinQ, which repeatedly schedules the same subset of links and thus limits diversity in channel access, SAGNN distributes channel access more evenly across links. Beyond its strong performance, SAGNN is also naturally suited for decentralized deployment since it relies only on local information [24]. Moreover, its foundation on GNNs provides valuable transferability properties—both within and across network scales [19, 25]— strengthening its practical relevance.

5 Conclusions

We presented a novel formulation and approach to the wireless link scheduling problem that emphasizes long-term performance metrics. By introducing state augmentation, we endowed the scheduling policy with temporal adaptability, enabling it to achieve strong and consistent performance. This technique proves effective in managing stochastic tasks and warrants further exploration. Comparative evaluations against several baselines demonstrate that the proposed algorithm achieves fair channel access while maintaining sustained high sum-rate performance through effective interference avoidance.

Future work could explore several promising directions. One is the analysis of the distribution of dual variables, with the goal of designing models that explicitly learn and exploit this distribution.

Another challenge lies in learning discrete probability distributions for the primal variables: in this work, we relied on the GNN's ability to approximate binary decisions, a naive but effective strategy. More advanced techniques—such as policy gradient methods (e.g., REINFORCE) or differentiable relaxations like the Gumbel-Softmax—may provide significant improvements.

References

[1] S. A. Jafar, "Topological interference management through index coding," *IEEE Transactions on Information Theory*, vol. 60, no. 1, pp. 529–568, 2013.

[2] N. Naderializadeh and A. S. Avestimehr, "Itlinq: A new approach for spectrum sharing in device-to-device communication systems," *IEEE journal on selected areas in communications*, vol. 32, no. 6, pp. 1139–1151, 2014.

[3] K. Shen and W. Yu, "FPLinQ: A cooperative spectrum sharing strategy for device-to-device communications," in *2017 IEEE international symposium on information theory (ISIT)*, pp. 2323–2327, IEEE, 2017.

[4] W. Cui, K. Shen, and W. Yu, "Spatial deep learning for wireless scheduling," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1248–1261, 2019.

[5] M. Lee, G. Yu, and G. Y. Li, "Graph embedding-based wireless link scheduling with few training samples," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2282–2294, 2021.

[6] R. Shelim and A. S. Ibrahim, "Geometric machine learning over riemannian manifolds for wireless link scheduling," *IEEE Access*, vol. 10, pp. 22854–22864, 2022.

[7] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, MobiCom '03, (New York, NY, USA), p. 66–80, Association for Computing Machinery, 2003.

[8] M. R. Garey and D. S. Johnson, "Strong np-completeness results: Motivation, examples, and implications," *J. ACM*, vol. 25, p. 499–508, July 1978.

[9] J. Yu, B. Huang, X. Cheng, and M. Atiquzzaman, "Shortest link scheduling algorithms in wireless networks under the sinr model," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2643–2657, 2017.

[10] Z. Zhao, G. Verma, C. Rao, A. Swami, and S. Segarra, "Link scheduling using graph neural networks," *IEEE Transactions on Wireless Communications*, vol. 22, no. 6, pp. 3997–4012, 2023.

[11] N. NaderiAlizadeh, M. Eisen, and A. Ribeiro, "State-augmented learnable algorithms for resource management in wireless networks," *IEEE Transactions on Signal Processing*, vol. 70, pp. 5898–5912, 2022.

[12] A. Ribeiro, "Ergodic stochastic optimization algorithms for wireless communication and networking," *IEEE Transactions on Signal Processing*, vol. 58, no. 12, pp. 6369–6386, 2010.

[13] F. Gama, A. G. Marques, G. Leus, and A. Ribeiro, "Convolutional neural network architectures for signals supported on graphs," *IEEE Transactions on Signal Processing*, vol. 67, no. 4, pp. 1034–1049, 2019.

[14] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[15] L. Ruiz, F. Gama, and A. Ribeiro, "Graph neural networks: Architectures, stability, and transferability," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 660–682, 2021.

[16] F. Gama, A. Ribeiro, and J. Bruna, "Stability of graph scattering transforms," *Advances in Neural Information Processing Systems*, vol. 32, pp. 8038–8048, 2019.

[17] L. Ruiz, L. F. Chamon, and A. Ribeiro, "Transferability Properties of Graph Neural Networks," *arXiv preprint arXiv:2112.04629*, 2021.

[18] Z. Wang, L. Ruiz, and A. Ribeiro, "Geometric graph filters and neural networks: Limit properties and discriminability trade-offs," *IEEE Transactions on Signal Processing*, 2024.

[19] S. Fernández, R. G. Camargo, M. Eisen, A. Ribeiro, and F. Larroca, "On the transferability of graph neural networks for resource allocation in wireless networks," in *2024 IEEE URUCON*, pp. 1–5, 2024.

[20] J. Du, S. Zhang, G. Wu, J. M. F. Moura, and S. Kar, "Topology adaptive graph convolutional networks," *CoRR*, vol. abs/1710.10370, 2017.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[22] I. Hounie, A. Ribeiro, and L. F. O. Chamon, "Resilient constrained learning," in *Advances in Neural Information Processing Systems* (A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, eds.), vol. 36, pp. 71767–71798, Curran Associates, Inc., 2023.

[23] N. NaderiAlizadeh, M. Eisen, and A. Ribeiro, "Learning resilient radio resource management policies with graph neural networks," *IEEE Transactions on Signal Processing*, vol. 71, pp. 995–1009, 2023.

[24] Z. Wang, M. Eisen, and A. Ribeiro, "Decentralized wireless resource allocation with graph neural networks," in *2020 54th Asilomar Conference on Signals, Systems, and Computers*, pp. 299–303, 2020.

[25] M. Eisen and A. Ribeiro, "Optimal wireless resource allocation with random edge graph neural networks," *ieee transactions on signal processing*, vol. 68, pp. 2977–2991, June 2020.

A Separability properties of $\overline{\mathcal{L}}$

We summarize the property in the following proposition.

Proposition 1 *The scheduling decisions from Eq.* (5) *can be recovered from the maximization of a time-decomposed Lagrangian* $\mathcal{L}$*.*

$$\mathbf{s}^{\dagger}(t, \boldsymbol{\lambda}) = \underset{\mathbf{s}(t)\in\{0,1\}^K}{\arg\max}\, \mathcal{L}\left(\mathbf{s}(t), \boldsymbol{\lambda}\right), \tag{15}$$

where we define $\mathcal{L}$ *as follows.*

$$\mathcal{L}(\mathbf{s}(t), \boldsymbol{\lambda}) = \mathbf{s}^{\mathsf{T}}(t).[\mathbf{1} - \mathbf{A}.\mathbf{s}(t)]_+ + \boldsymbol{\lambda}^{\mathsf{T}}\left(\mathbf{s}(t) \odot [\mathbf{1} - \mathbf{A}.\mathbf{s}(t)]_+ - \boldsymbol{\Delta}\right). \tag{16}$$

Proof. Observe that the sum of averages is equal to the average of sums when the two quantities have the same number of elements (T, in this case).

$$\begin{aligned}
\overline{\mathcal{L}}\left(\{\mathbf{s}(t)\}_{t=0}^{T-1}, \boldsymbol{\lambda}\right) &= \frac{1}{T}\sum_{t=0}^{T-1}\left(\mathbf{s}^{\mathsf{T}}(t)[\mathbf{1}_K - \mathbf{A}\mathbf{s}(t)]_+\right) + \boldsymbol{\lambda}^{\mathsf{T}}\left(\frac{1}{T}\sum_{t=0}^{T-1}\mathbf{s}(t) \odot [\mathbf{1}_K - \mathbf{A}\mathbf{s}(t)]_+ - \boldsymbol{\Delta}\right) \\
&= \frac{1}{T}\sum_{t=0}^{T-1}\left(\mathbf{s}^{\mathsf{T}}(t).[\mathbf{1} - \mathbf{A}.\mathbf{s}(t)]_+ + \boldsymbol{\lambda}^{\mathsf{T}}\left(\mathbf{s}(t) \odot [\mathbf{1} - \mathbf{A}.\mathbf{s}(t)]_+ - \boldsymbol{\Delta}\right)\right) \qquad (17) \\
&= \frac{1}{T}\sum_{t=0}^{T-1}\mathcal{L}(\mathbf{s}(t), \boldsymbol{\lambda}). \qquad (18)
\end{aligned}$$

■

B Proposed algorithms for training and execution

The proposed algorithms for training and execution are presented in Algorithms 1 and 2, respectively.

Algorithm 1 Training Phase for the SAGNN Link Scheduling Algorithm

Input: Number of training iterations M, primal learning rate $\eta_{\mathbf{H}}$.
Initialize: $\mathbf{H}_0$ randomly
for $m = 0, \cdots, M-1$ **do**
    Randomly sample $\boldsymbol{\lambda} \sim p_{\boldsymbol{\lambda}}$.
    Generate scheduling decisions $\boldsymbol{\Phi}(\mathbf{A}, \mathbf{H}_m; \boldsymbol{\lambda})$.
    Calculate the augmented Lagrangian according to (12).
    Update the primal parameters

$$\mathbf{H}_{m+1} = \mathbf{H}_m + \eta_{\mathbf{H}} \nabla_{\mathbf{H}} \mathcal{L}_{\boldsymbol{\lambda}}(\mathbf{H}_m).$$

end for
Return: Optimal model parameters $\mathbf{H}^* = \mathbf{H}_{M-1}$.

Algorithm 2 Execution Phase for the SAGNN Link Scheduling Algorithm

Input: Optimal model parameters $\mathbf{H}^*$, number of time steps T, dual learning rate $\eta_{\boldsymbol{\lambda}}$
Initialize: $\boldsymbol{\lambda}_0 = \mathbf{0}$
for $t = 0, \cdots, T-1$ **do**
    Generate scheduling decisions $\mathbf{s}(t) = \boldsymbol{\Phi}(\mathbf{A}, \mathbf{H}^*; \boldsymbol{\lambda}_t)$.
    Update the dual parameters

$$\boldsymbol{\lambda}_{t+1} = \left[\boldsymbol{\lambda}_t - \eta_{\boldsymbol{\lambda}} \nabla_{\boldsymbol{\lambda}} \mathcal{L}_{\boldsymbol{\lambda}}(\mathbf{H}^*)\right]_+ .$$

end for
Return: Optimal Scheduling decisions $\mathbf{s}(t)$, $t = 0, 1, \cdots, T-1$.

Resilience A resilient formulation of the problem [22] is considered for the experiments to mitigate the effects that few links with infeasible constraints might have on the overall solution. This introduces a new term to the Lagrangian function as follows.

$$\mathcal{L}_{\boldsymbol{\lambda}}(\mathbf{H}) = \boldsymbol{\Phi}(\mathbf{A}, \mathbf{H}; \boldsymbol{\lambda})^{\mathsf{T}} [\mathbf{1} - \mathbf{A}\boldsymbol{\Phi}(\mathbf{A}, \mathbf{H}; \boldsymbol{\lambda})]_+ \tag{19}$$

$$+ \boldsymbol{\lambda}^{\mathsf{T}} \left(\boldsymbol{\Phi}(\mathbf{A}, \mathbf{H}; \boldsymbol{\lambda}) \odot ([\mathbf{1} - \mathbf{A}\boldsymbol{\Phi}(\mathbf{A}, \mathbf{H}; \boldsymbol{\lambda})]_+ - \boldsymbol{\Delta})\right) \tag{20}$$

$$+ \frac{1}{\alpha} \boldsymbol{\lambda} \tag{21}$$

While the update of the primal variables is not affected by this change, the update of the dual variables is adapted:

$$\boldsymbol{\lambda}_{k+1} = \left[\boldsymbol{\lambda}_k - \eta_{\boldsymbol{\lambda}} \Big(\boldsymbol{\Phi}^{\mathsf{T}}(\mathbf{A}, \mathbf{H}^*; \boldsymbol{\lambda}_k) \odot ([\mathbf{1} - \mathbf{A}\boldsymbol{\Phi}(\mathbf{A}, \mathbf{H}^*; \boldsymbol{\lambda}_k)]_+ - \boldsymbol{\Delta}) + \frac{1}{\alpha} \boldsymbol{\lambda} \Big) \right]_+ . \tag{22}$$

The resilience factor α is fixed at 0.05.

Probability distribution of $\boldsymbol{\lambda}$ During the first epoch, dual variables are sampled from an arbitrary uniform distribution $\mathcal{U}[0, 2]^K$, where the upper bound corresponds to the dual learning rate $\eta_{\boldsymbol{\lambda}}$. Since each epoch includes evaluations on both the training and validation sets, we collect the dual variable vectors encountered during the execution of Algorithm 2 on the training graphs. In subsequent epochs, instead of sampling from the uniform distribution, we sample dual variable vectors from this empirical distribution p_λ.

C Runtimes for inference

The performance of our algorithm and the baselines are summarized in Table 2, averaging results over $T = 200$ time steps for 10 unseen graphs. While SAGNN does not achieve the highest average rates, it demonstrates strong overall performance and effectively minimizes constraint violations. The scheduling time per graph per time step is also reported, highlighting the efficiency of our method and further supporting its suitability as a practical solution.

	Constraint violation (%)	Objective function (%)	Runtime
p-persistent	97.57±0.62	2.69±0.08	$0.04 \pm 4.88 \times 10^{-3}$ ms
p-persistent CA	85.47±1.44	6.27±0.14	$0.04 \pm 3.73 \times 10^{-3}$ ms
FPLinQ	74.91±0.75	25.09±0.75	15.3 ± 0.50 ms
MWIS	1.55±0.34	**27.12±0.21**	14.22 ± 2.30 s
SAGNN	**0.75±0.45**	21.91±0.26	1.11 ± 0.03 ms

Table 2: Performance evaluation averaged across 10 graphs from the testing dataset, showing constraint violation and objective function in percentage of links. We also present average runtime required for generating a link scheduling for one time step, for one graph.