Adaptive Wireless Power Allocation with Graph Neural Networks

Navid NaderiAlizadeh[1], Mark Eisen[2], and Alejandro Ribeiro[1]

[1] University of Pennsylvania [2] Intel Labs

ICASSP 2022 SINGAPORE, 22-27 MAY 2022

System Model

- We consider an interference channel with m transmitter-receiver pairs $\{(Tx_i, Rx_i)\}_{i=1}^m$
- The channel gain between Tx_i and Rx_j is denoted by h_{ij}
  - The entire network channel matrix: $\mathbf{H} \in \mathbb{C}^{m \times m}$
  - The channel is modeled as $\mathbf{H} = \mathbf{H}^\ell \mathbf{H}^s$, comprising long-term ($\mathbf{H}^\ell$) and short-term ($\mathbf{H}^s$) fading components
- The signal-to-interference-plus-noise ratio (SINR) at Rx_i can be written as

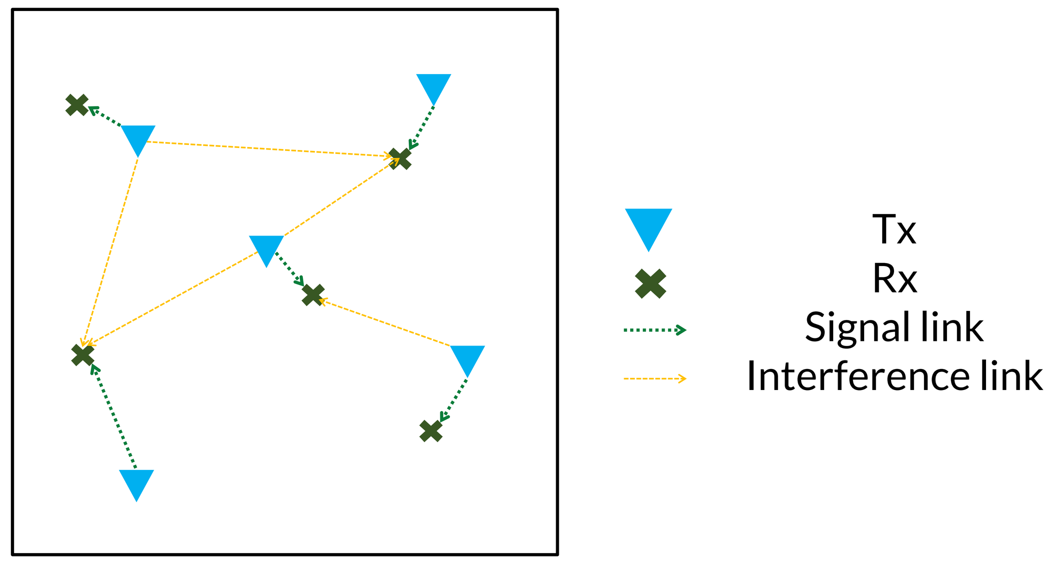
$$\mathsf{SINR}_i(\mathbf{H}, \mathbf{p}) = \frac{|h_{ii}|^2 p_i(\mathbf{H})}{\sigma^2 + \sum_{j \neq i} |h_{ji}|^2 p_j(\mathbf{H})}$$

Transmit power of Tx_i

Noise variance

- The Shannon capacity between Tx_i and Rx_i is then given by

$$f_i(\mathbf{H}, \mathbf{p}) = \log_2(1 + \mathsf{SINR}_i(\mathbf{H}, \mathbf{p}))$$



Resilient power allocation formulation

- Our goal is to learn a power allocation policy that manages the interference among these concurrent transmissions, and optimizes two metrics of interest:
  - Sum throughput, representing the "cell-center" performance, and
  - 5th percentile throughput, representing the "cell-edge" performance.

$$\max_{\mathbf{p},\mathbf{x},\mathbf{z}} \quad \mathbb{E}_{\mathbf{H}^\ell}\left[\mathcal{U}(\mathbf{x}(\mathbf{H}^\ell)) - \frac{\alpha}{2}\|\mathbf{z}(\mathbf{H}^\ell)\|_2^2\right],$$
$$\text{s.t.} \quad \mathbf{x}(\mathbf{H}^\ell) = \mathbb{E}_{\mathbf{H}^s}\left[\mathbf{f}(\mathbf{H}^\ell\mathbf{H}^s, \mathbf{p}(\mathbf{H}^\ell\mathbf{H}^s))\right], \ \forall \mathbf{H}^\ell$$
$$\mathbf{x}(\mathbf{H}^\ell) \geq f_{\min} - \mathbf{z}(\mathbf{H}^\ell), \ \forall \mathbf{H}^\ell$$
$$\mathbf{p}(\mathbf{H}) \in [0, P_{\max}]^m, \ \mathbf{x}(\mathbf{H}^\ell) \geq \mathbf{0}, \ \mathbf{z}(\mathbf{H}^\ell) \geq \mathbf{0}.$$

- $\mathbf{x}(\mathbf{H}^\ell)$: Ergodic average rate
- $\mathcal{U}(\cdot)$: Concave utility
- $P_{\max}$: Maximum transmit power
- $f_{\min}$: Minimum ergodic rate constraint
- $\mathbf{z}(\mathbf{H}^\ell)$: Slack variables

- The slack variables adapt the constraints to the underlying channel conditions, hence helping to treat cell-edge and cell-center users fairly

Parameterized primal-dual unsupervised learning

- The Lagrangian function can be written as follows:

$$\mathcal{L}(\mathbf{p}, \mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\mu})$$
$$:= \mathbb{E}_{\mathbf{H}^\ell}\left[\mathcal{U}(\mathbf{x}(\mathbf{H}^\ell)) - \frac{\alpha}{2}\|\mathbf{z}(\mathbf{H}^\ell)\|_2^2\right]$$
$$- \int_{\mathcal{H}^\ell} \Big\{ \boldsymbol{\lambda}(\mathbf{H}^\ell)^T \left[\mathbf{x}(\mathbf{H}^\ell) - \mathbb{E}_{\mathbf{H}^s}\left[\mathbf{f}(\mathbf{H}, \mathbf{p}(\mathbf{H}))\right]\right]$$
$$+ \boldsymbol{\mu}(\mathbf{H}^\ell)^T \left[f_{\min} - \mathbf{z}(\mathbf{H}^\ell) - \mathbf{x}(\mathbf{H}^\ell)\right] \Big\} d\mathbf{H}^\ell$$

Dual multiplier functions

- We then replace each policy $\mathbf{y}(\cdot)$ with a parameterized version $\mathbf{y}(\cdot; \boldsymbol{\theta}^{\mathbf{y}})$, leading to the parameterized Lagrangian

$$\mathcal{L}_\theta\left(\boldsymbol{\theta}^{\mathbf{p}}, \boldsymbol{\theta}^{\mathbf{x}}, \boldsymbol{\theta}^{\mathbf{z}}, \boldsymbol{\theta}^{\boldsymbol{\lambda}}, \boldsymbol{\theta}^{\boldsymbol{\mu}}\right) := \mathcal{L}\left(\mathbf{p}(\cdot; \boldsymbol{\theta}^{\mathbf{p}}), \mathbf{x}(\cdot; \boldsymbol{\theta}^{\mathbf{x}}), \mathbf{z}(\cdot; \boldsymbol{\theta}^{\mathbf{z}}), \boldsymbol{\lambda}(\cdot; \boldsymbol{\theta}^{\boldsymbol{\lambda}}), \boldsymbol{\mu}(\cdot; \boldsymbol{\theta}^{\boldsymbol{\mu}})\right)$$

And the parameterized dual problem, $D_\theta^* := \min_{\boldsymbol{\theta}^{\boldsymbol{\lambda}}, \boldsymbol{\theta}^{\boldsymbol{\mu}}} \max_{\boldsymbol{\theta}^{\mathbf{p}}, \boldsymbol{\theta}^{\mathbf{x}}, \boldsymbol{\theta}^{\mathbf{z}}} \mathcal{L}_\theta\left(\boldsymbol{\theta}^{\mathbf{p}}, \boldsymbol{\theta}^{\mathbf{x}}, \boldsymbol{\theta}^{\mathbf{z}}, \boldsymbol{\theta}^{\boldsymbol{\lambda}}, \boldsymbol{\theta}^{\boldsymbol{\mu}}\right)$

- We then update the primal and dual parameters by iteratively performing gradient ascent/descent steps as follows:

$$\boldsymbol{\theta}_{k+1}^{\mathbf{p}} = \boldsymbol{\theta}_k^{\mathbf{p}} + \eta_{\mathbf{p}} \nabla_{\boldsymbol{\theta}^{\mathbf{p}}} \mathcal{L}_\theta\left(\boldsymbol{\theta}^{\mathbf{p}}, \boldsymbol{\theta}^{\mathbf{x}}, \boldsymbol{\theta}^{\mathbf{z}}, \boldsymbol{\theta}^{\boldsymbol{\lambda}}, \boldsymbol{\theta}^{\boldsymbol{\mu}}\right)$$
$$\boldsymbol{\theta}_{k+1}^{\mathbf{x}} = \boldsymbol{\theta}_k^{\mathbf{x}} + \eta_{\mathbf{x}} \nabla_{\boldsymbol{\theta}^{\mathbf{x}}} \mathcal{L}_\theta\left(\boldsymbol{\theta}^{\mathbf{p}}, \boldsymbol{\theta}^{\mathbf{x}}, \boldsymbol{\theta}^{\mathbf{z}}, \boldsymbol{\theta}^{\boldsymbol{\lambda}}, \boldsymbol{\theta}^{\boldsymbol{\mu}}\right)$$
$$\boldsymbol{\theta}_{k+1}^{\mathbf{z}} = \boldsymbol{\theta}_k^{\mathbf{z}} + \eta_{\mathbf{z}} \nabla_{\boldsymbol{\theta}^{\mathbf{z}}} \mathcal{L}_\theta\left(\boldsymbol{\theta}^{\mathbf{p}}, \boldsymbol{\theta}^{\mathbf{x}}, \boldsymbol{\theta}^{\mathbf{z}}, \boldsymbol{\theta}^{\boldsymbol{\lambda}}, \boldsymbol{\theta}^{\boldsymbol{\mu}}\right)$$
$$\boldsymbol{\theta}_{k+1}^{\boldsymbol{\lambda}} = \boldsymbol{\theta}_k^{\boldsymbol{\lambda}} - \eta_{\boldsymbol{\lambda}} \nabla_{\boldsymbol{\theta}^{\boldsymbol{\lambda}}} \mathcal{L}_\theta\left(\boldsymbol{\theta}^{\mathbf{p}}, \boldsymbol{\theta}^{\mathbf{x}}, \boldsymbol{\theta}^{\mathbf{z}}, \boldsymbol{\theta}^{\boldsymbol{\lambda}}, \boldsymbol{\theta}^{\boldsymbol{\mu}}\right)$$
$$\boldsymbol{\theta}_{k+1}^{\boldsymbol{\mu}} = \boldsymbol{\theta}_k^{\boldsymbol{\mu}} - \eta_{\boldsymbol{\mu}} \nabla_{\boldsymbol{\theta}^{\boldsymbol{\mu}}} \mathcal{L}_\theta\left(\boldsymbol{\theta}^{\mathbf{p}}, \boldsymbol{\theta}^{\mathbf{x}}, \boldsymbol{\theta}^{\mathbf{z}}, \boldsymbol{\theta}^{\boldsymbol{\lambda}}, \boldsymbol{\theta}^{\boldsymbol{\mu}}\right)$$

Graph neural network- (GNN-)based parameterizations

- We model the wireless network as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, r, w)$
  - $\mathcal{V} = \{1, \ldots, m\}$: Set of graph nodes
  - $\mathcal{E} = \mathcal{V}^2 \backslash \{(i,i)\}_{i \in \mathcal{V}}$: Set of graph edges
  - $r: \mathcal{V} \to \mathbb{R}^{F_0}$: Initial node feature function
  - $w: \mathcal{E} \to \mathbb{R}$: Edge weight function
- For each node $v \in \mathcal{V}$, we let $\mathbf{y}_v^0 = r(v)$ denote its initial feature vector
- Then, the features get transformed through multiple layers, where the feature vector of node v at layer l can be written as

$$\mathbf{y}_v^l = \mu\left(\mathbf{y}_v^{l-1}\boldsymbol{\theta}_1^l + \sum_{u:(u,v) \in \mathcal{E}} w(u,v)\left(\mathbf{y}_v^{l-1}\boldsymbol{\theta}_2^l - \mathbf{y}_u^{l-1}\boldsymbol{\theta}_3^l\right)\right)$$

- We denote the final feature vector of node v, i.e., its node embedding, as $\mathbf{s}_v$
- We then use three GNNs to parameterize the primal/dual policies:
  - Main GNN (responsible for $\mathbf{p}(\cdot)$): The power level of Tx_i is given by

$$p_i(\mathbf{H}) = P_{\max} \cdot \sigma\left(\mathbf{b}_{\mathbf{p}}^T \mathbf{s}_i\right)$$

  - Auxiliary GNN (responsible for $\mathbf{x}(\cdot)$ and $\mathbf{z}(\cdot)$): The variables for (Tx_i, Rx_i) are given by

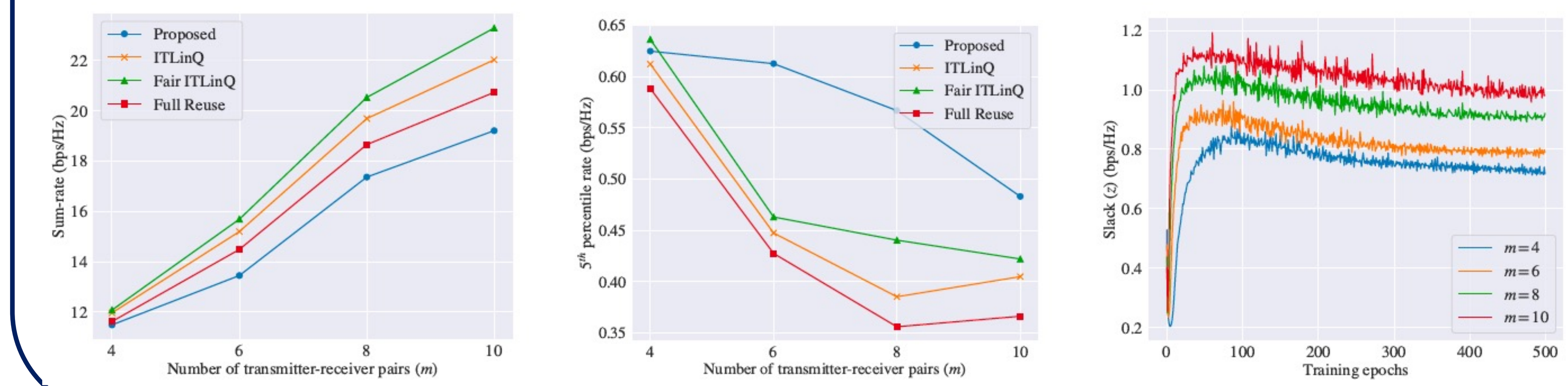
$$x_i(\mathbf{H}^\ell) = \mathbf{b}_{\mathbf{x}}^T \mathbf{s}_i',$$
$$z_i(\mathbf{H}^\ell) = \left[\mathbf{b}_{\mathbf{z}}^T \mathbf{s}_i'\right]_+$$

  - Dual GNN (responsible for $\boldsymbol{\lambda}(\cdot)$ and $\boldsymbol{\mu}(\cdot)$): The variables for (Tx_i, Rx_i) are given by

$$\lambda_i(\mathbf{H}^\ell) = \left[\mathbf{b}_{\boldsymbol{\lambda}}^T \mathbf{s}_i''\right]_+$$
$$\mu_i(\mathbf{H}^\ell) = \left[\mathbf{b}_{\boldsymbol{\mu}}^T \mathbf{s}_i''\right]_+$$

Experimental evaluation

- We evaluate the proposed method on networks with 6-14 transmitter-receiver pairs
- Our proposed algorithm learns how to adaptively elevate the slack variable for larger and denser networks to make the optimization problem feasible and maximize the sum-rate utility function.
- This leads to a much fairer resource allocation policy, where a superior trade-off between the sum-rate and the 5th percentile rate can be achieved.



References

E. Ranjan, S. Sanyal, and P. Talukdar, "ASAP: Adaptive structure aware pooling for learning hierarchical graph representations," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 04, pp. 5470– 5477, 2020.

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.

GitHub