Distributed Scheduling using Graph Neural Networks

Zhongyuan Zhao (zhongyuan.zhao@rice.edu Rice University)
Gunjan Verma (gunjan.verma.civ@mail.mil US Army's CCDC Army Research Laboratory)
Chirag Rao (chirag.r.rao.civ@mail.mil US Army's CCDC Army Research Laboratory)
Ananthram Swami (ananthram.swami.civ@mail.mil US Army's CCDC Army Research Laboratory)
Santiago Segarra (segarra@rice.edu Rice University)

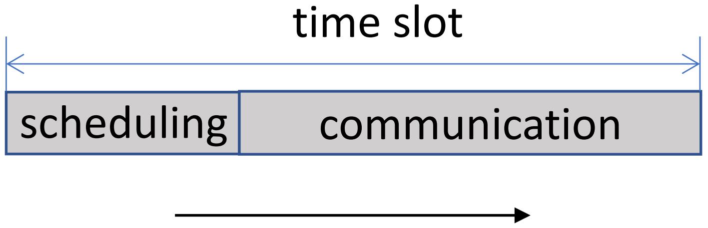
IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)
Toronto, Ontario, Canada
6-11 June 2021

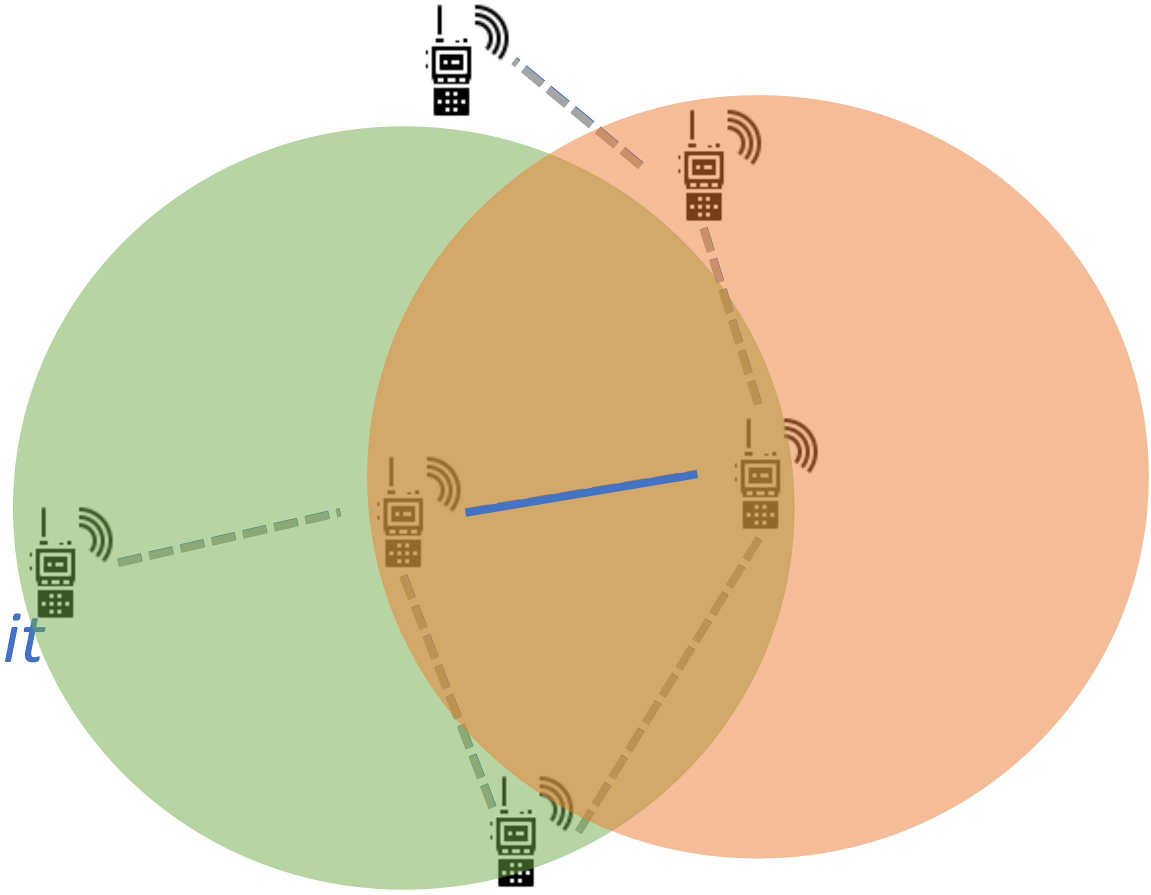
Outline

- Introduction
- Existing schedulers
- GCN-based distributed scheduler
- Numerical results
- Conclusion

Link scheduling in wireless ad-hoc networks

- Infrastructureless
- Orthogonal access
- Time-slotted





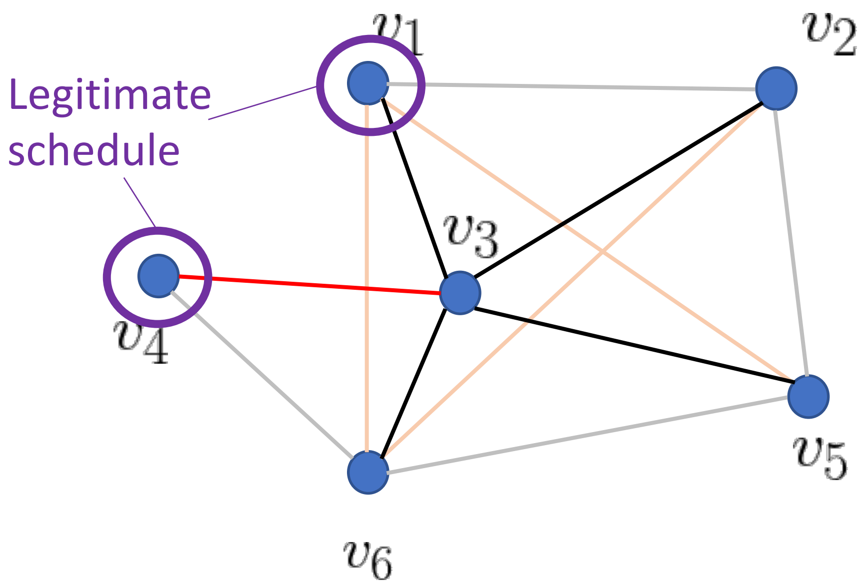
Scheduler decides which links to transmit in a <u>distributed</u> manner

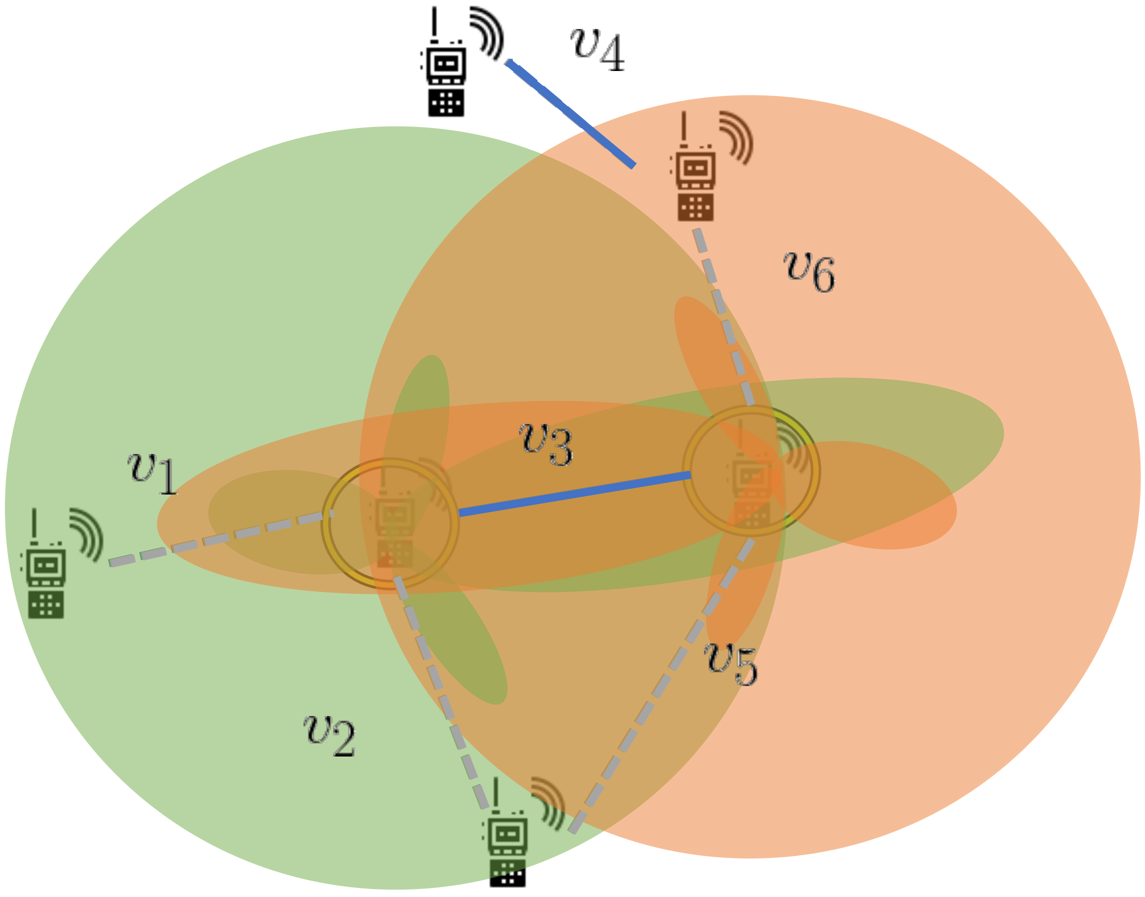
Assumptions
- Single channel
- Single radio interface per node
- Constant TX power

Icons made by photo3idea-studio from Flaticon

Conflict graph

- Vertices→ links in network
- Edges
  - Interface constraints
  - Interference relationship

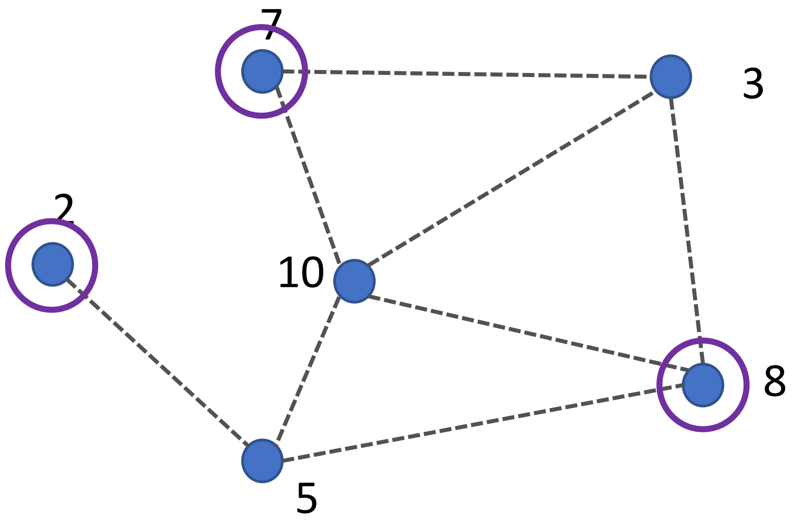




Icons made by photo3idea-studio from Flaticon

Optimal scheduling as MWIS solver

- Conflict graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$
- Utility function $u : \mathcal{V} \rightarrow \mathbb{R}_+$
- Optimal scheduling $\mathbf{v}^* \subseteq \mathcal{V}$



$$\mathbf{v}^* = \underset{\mathbf{v} \in \mathcal{V}}{\mathbf{argmax}} \sum_{v \in \mathbf{v}} u(v)$$

$$\text{s.t.} \quad e(v_i, v_j) \notin \mathcal{E}, \forall v_i, v_j \in \mathbf{v} .$$

Constraint of independent set

Maximum weighted independent set (MWIS)

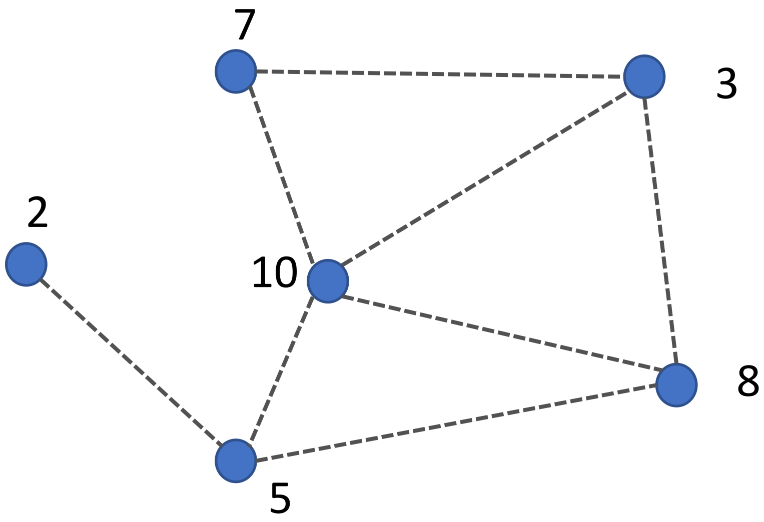
NP-hard → Heuristics

Per time slot, per channel

Existing distributed MWIS schedulers

- Per link utility function
  - q, queue length [Joo'12]
  - r , predicted link rate [Douik'18]
  - qr [Joo'15], q/r [Paschalidis'15]
  - Analytical form [Marques'11]
- Heuristic MWIS solver
  - Centralized greedy solver
  - Local greedy solver [Joo'12]
  - Threshold local greedy [Joo'15]
  - Message passing [Paschalidis'15]

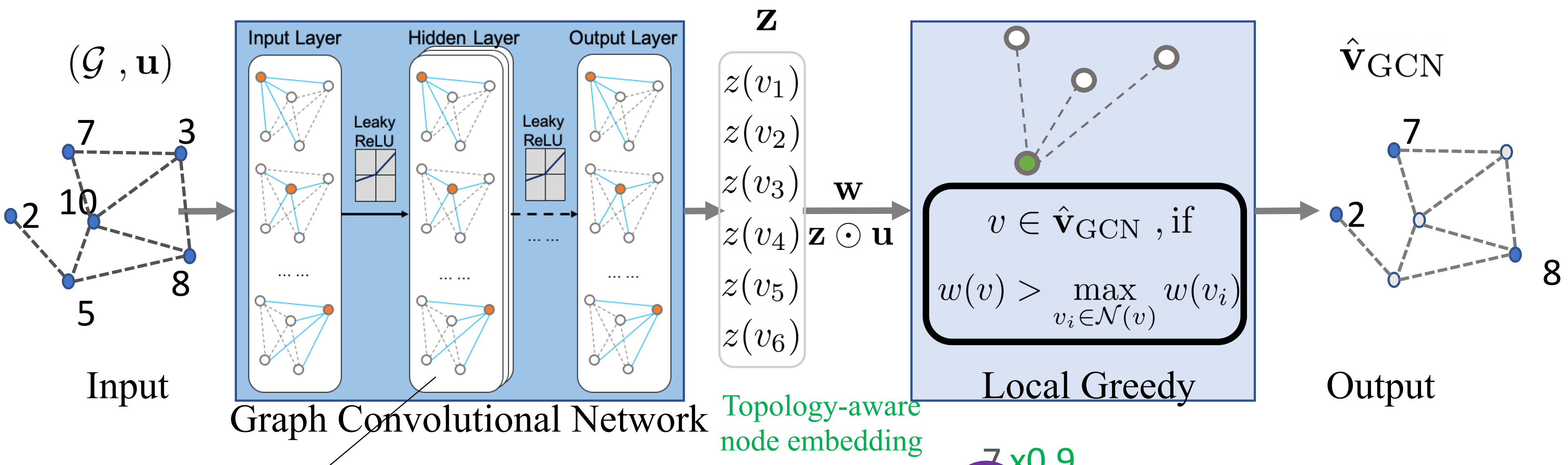
Average local complexity (Local exchanges)
$\log(\|\mathcal{V}\|)$
$\log_{\alpha}(\beta\|\mathcal{V}\|)$
$2\|\mathcal{V}\|$



Not fully leverage the graph topology

C. Joo and N. B. Shroff, "Local greedy approximation for scheduling in multihop wireless networks," IEEE Trans. on Mobile Computing, vol. 11, no. 3, pp. 414–426, 2012.
C. Joo, X. Lin, J. Ryu, and N. B. Shroff, "Distributed greedy approximation to maximum weighted independent set for scheduling with fading channels," IEEE/ACM Trans. on Networking, vol. 24, no. 3, pp. 1476–1488, 2015.
I. C. Paschalidis, F. Huang, and W. Lai, "A message-passing algorithm for wireless network scheduling," IEEE/ACM Trans. Netw., vol. 23, no. 5, pp. 1528–1541, Oct. 2015.
A. Douik, H. Dahrouj, T. Y. Al-Naffouri, and M. Alouini, "Distributed hybrid scheduling in multi-cloud networks using conflict graphs," IEEE Trans. on Communications, vol. 66, no. 1, pp. 209–224, 2018.
A. G. Marques, N. Gatsis, and G. B. Giannakis, "Optimal cross-layer design of wireless fading multi-hop networks," Cross Layer Designs in WLAN Systems, 2011.

GCN-based distributed MWIS solver

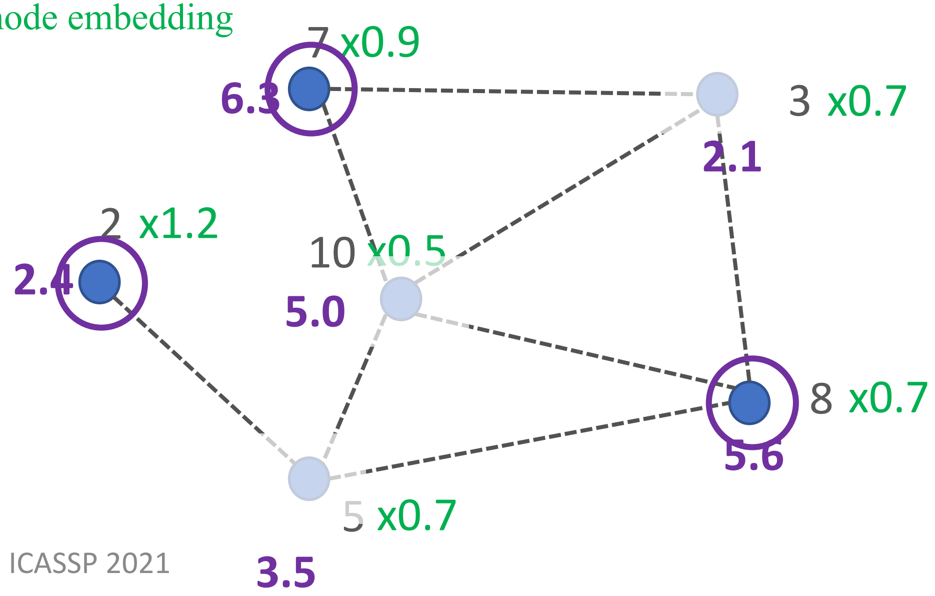


$$\mathbf{X}^{l+1} = \sigma\left(\mathbf{X}^l \mathbf{\Theta}_0^l + \mathcal{L}\mathbf{X}^l \mathbf{\Theta}_1^l\right)$$

$$\mathcal{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}$$

$$\mathbf{\Theta}_0^l, \mathbf{\Theta}_1^l \in \mathbb{R}^{g_l \times g_{l+1}}$$

$$\mathbf{X}^l \in \mathbb{R}^{|\mathcal{V}| \times g_l}, \mathbf{X}^{l+1} \in \mathbb{R}^{|\mathcal{V}| \times g_{l+1}}$$



Distributed GCN: a minimal example

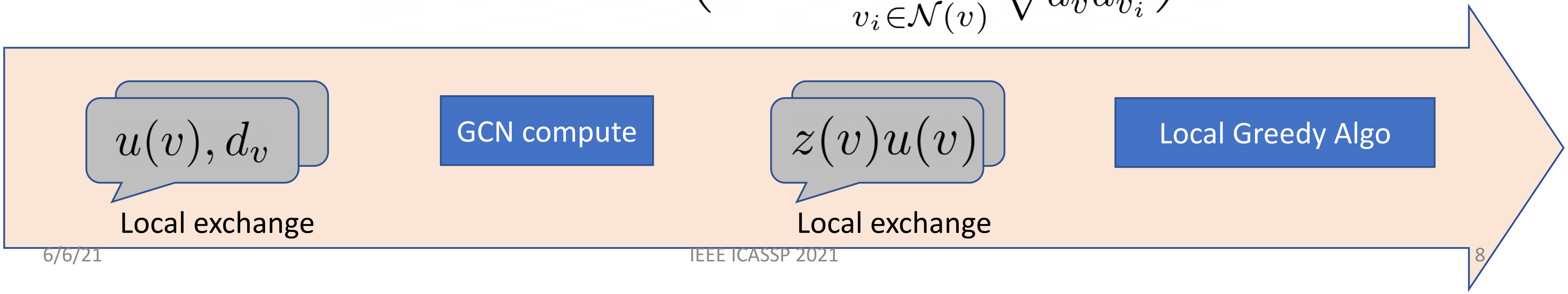
$$\mathbf{X}^{l+1} = \sigma\left(\mathbf{X}^{l}\boldsymbol{\Theta}_0^{l} + \mathcal{L}\mathbf{X}^{l}\boldsymbol{\Theta}_1^{l}\right)$$

- 1-layer GCN
- Input and output dimensions
- Local computing

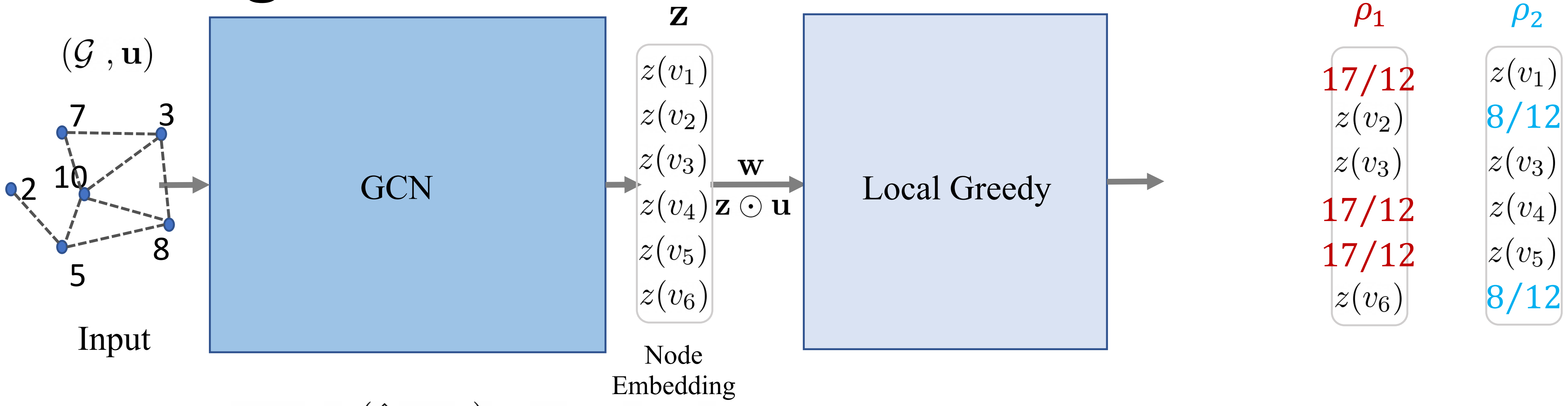
$$\mathbf{X}^0 = \mathbf{u}\,,\mathbf{X}^1 = \mathbf{z}$$

$$g_0 = g_1 = 1$$

$$z(v) = u(v)\theta_0 + \left(u(v) - \sum_{v_i \in \mathcal{N}(v)} \frac{u(v_i)}{\sqrt{d_v d_{v_i}}}\right)\theta_1$$



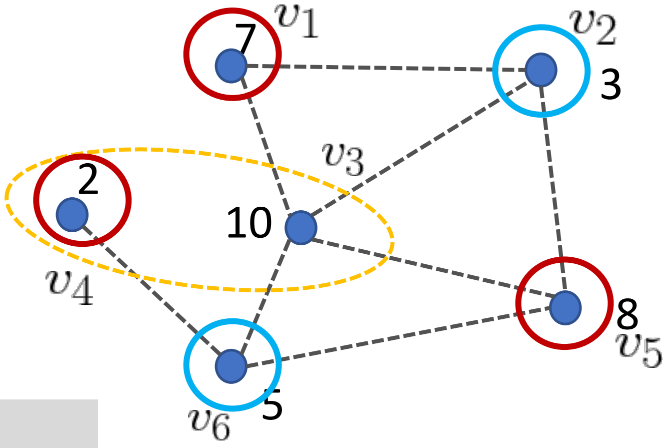
Learning with non-differentiable module



Reward $$\rho(v) = \frac{u(\hat{\mathbf{v}}_{GCN})}{u(\hat{\mathbf{v}}_{Gr})}, \text{ if } v \in \hat{\mathbf{v}}_{GCN}$$

Loss function (root mean square) $$\ell(\mathcal{O}\,; \mathcal{G}, \mathbf{u}) = \sqrt{\sum_{v \in \mathcal{V}} \frac{[z(v) - \rho(v)]^2}{|\mathcal{V}|}}$$

Greedy: 12
GCN (1): 17
GCN (2): 8



Adam optimizer, Learning rate 0.0001
Exponential decaying exploration ϵ
Periodic gradient reset

No labelled datasets required!

Numerical Evaluations

- GCN config
  - No. of Layers $L = 1, 3, 20$
  - Feature dimensions

$$g_l = \begin{cases} 1, & l = 0, L \\ 32, & \text{otherwise} \end{cases}$$

  - Activations

$$\sigma_l(\cdot) = \begin{cases} \text{linear}, & l = L \\ \text{Leaky ReLU}, & 1 \leq l \leq L \end{cases}$$

Training for 25 epochs

- Training dataset
  - Erdő́s–Rényi(ER) (N, p)
  - Barabási–Albert(BA) (N, m)

  Choose one (ER or BA)

  - Graph size

    $N \in \{100, 150, 200, 250, 300\}$
  - Average degree

    $m = Np \in \{2, 5, 7.5, 10, 12.5\}$

  5000 ($5 \times 5 \times 20$)

  - Small graphs

    $N \in \{30, 100\}$

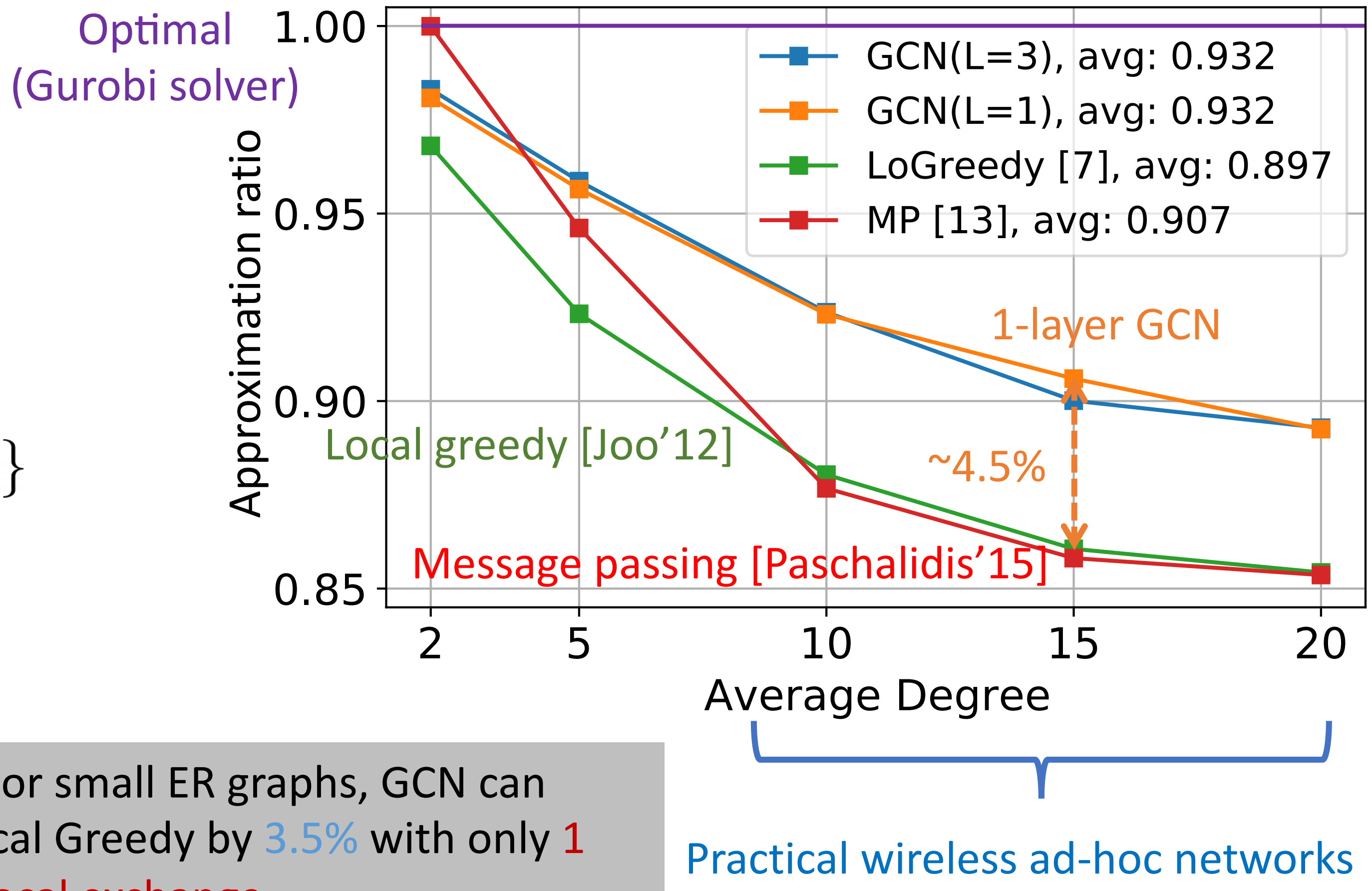
    $p \in \{0.1, 0.2, \ldots, 0.9\}$

  800

  - Weight dist. $u(v) \sim \mathcal{U}(0, 1)$

Performance on Erdő́s–Rényi(ER) graphs

- Training dataset
  - 5800 ER graphs
- Testing dataset
  - 500 ER graphs (N, p)
  - Graph size $N \in \{100, 150, 200, 250, 300\}$
  - Average degree $Np \in \{2, 5, 10, 15, 20\}$
  - Weight dist. $u(v) \sim \mathcal{U}(0, 1)$



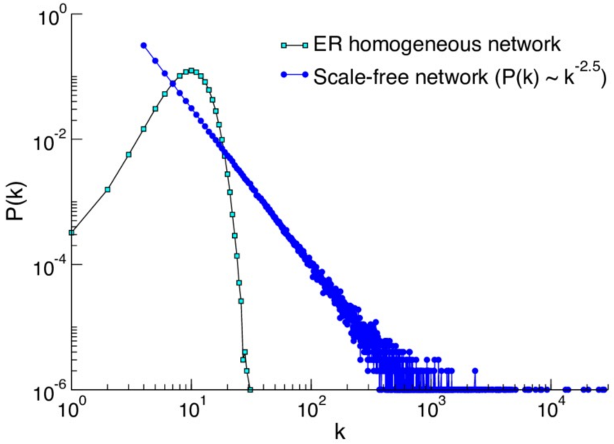
Remark 1: For small ER graphs, GCN can improve Local Greedy by 3.5% with only 1 additional local exchange

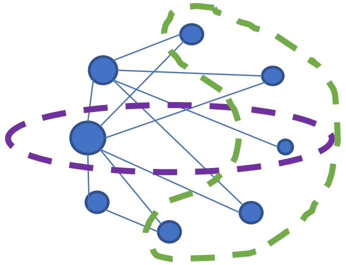
Remark 2: No. of layers has tiny influence

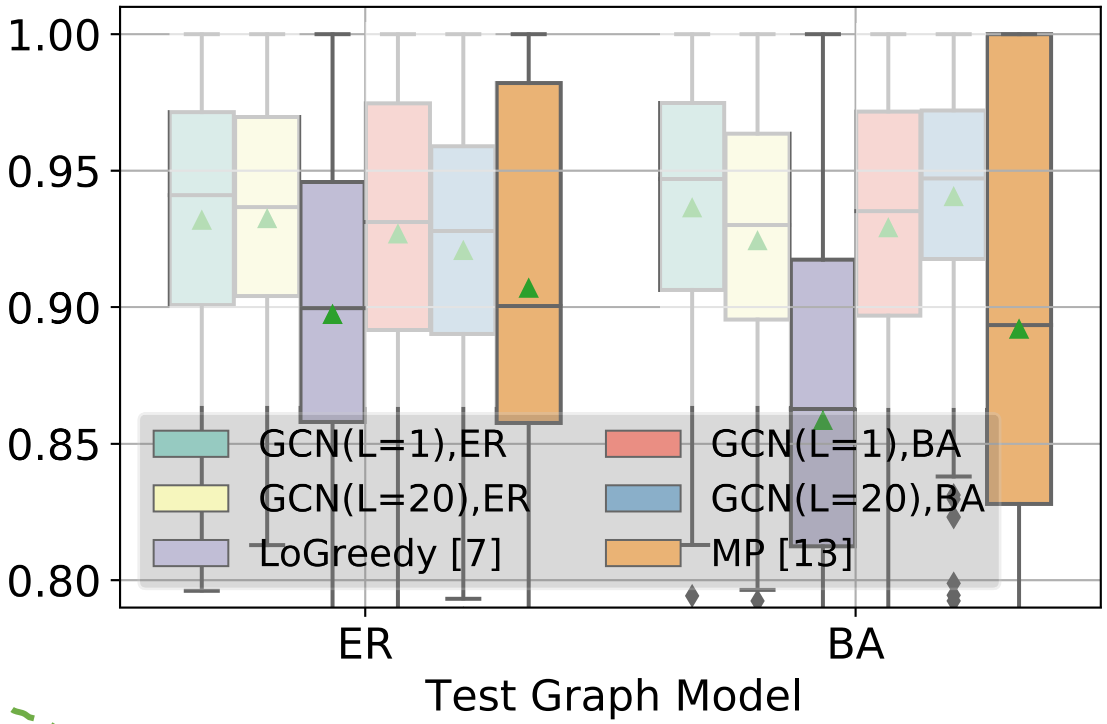
Generalizability across graph types

		Testing	
		ER (500)	BA (500)
Training	ER (5800)	0.932	0.936
	BA (5800)	0.925	0.928

For BA graph $m = Np$
Other settings identical







Remark 1: Local Greedy and MP perform worse on BA graphs (high-degree nodes)

Remark 2: GCN holds across graph models

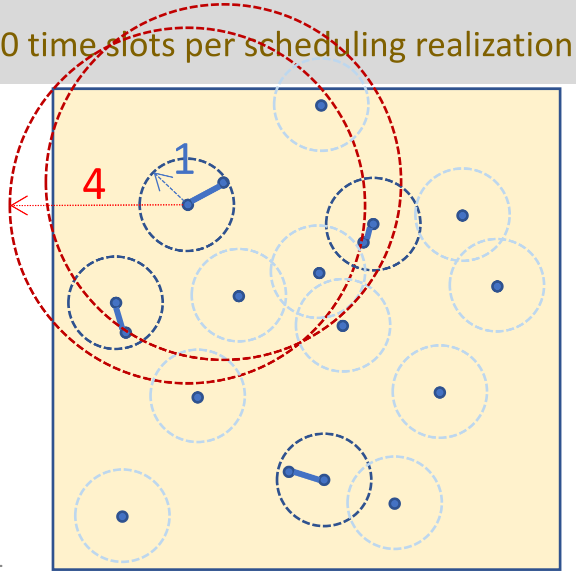
Remark 3: Deeper GCN → lower variance

Scheduling in mid-sized 1-hop wireless networks

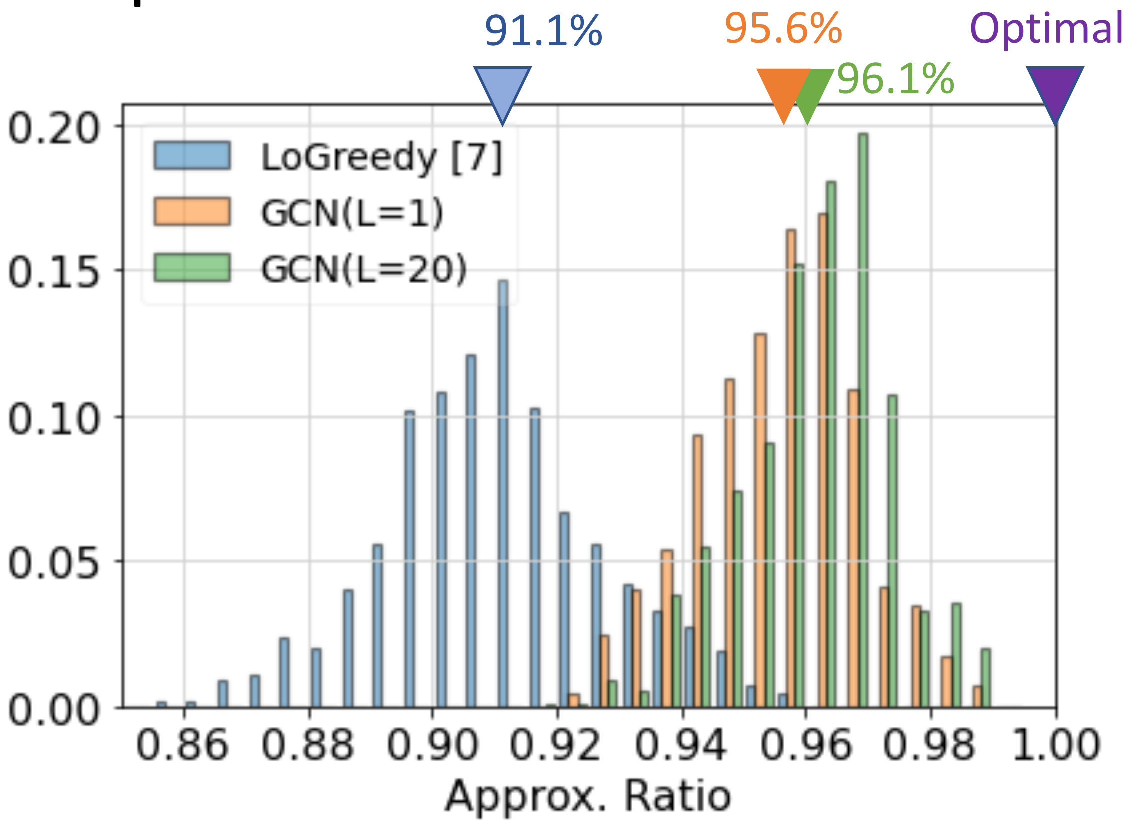
- Randomly located users
- 100 users, 40~60 links, single hop flow
- Average degree 13.1
- Link rate $r(v) \sim \mathcal{U}(0, 100)$ fading
- Utility function $u(v) = \min(q(v), r(v))$
- Flooding traffic

packets could be delivered

- 100 network instances
- 10 scheduling realizations per instance
- 200 time slots per scheduling realization



Average throughput



Remark 1: On small-mid ad-hoc networks (100 users), GCN can close half suboptimality gap of Local Greedy with only 1 additional local exchange

Remark 2: 1-layer GCN is a good choice

Conclusion & future work

- Link scheduling for orthogonal multiple access
- GCN-based heuristic scheduler
  - Leverage network topology
  - Efficiency of local greedy solver
  - Reinforcement learning
  - Centralized training, distributed deployment
- Performance gain: 91.1% -> 95.6%
- Scalability: $L + log(V)$
- Generalizability
- Scalar embedding → vectorized embeddings
- Memoryless agent → strategic agent
- Non-orthogonal access

Thank You!

Zhongyuan Zhao (zhongyuan.zhao@rice.edu Rice University)

Gunjan Verma (gunjan.verma.civ@mail.mil US Army's CCDC Army Research Laboratory)

Chirag Rao (chirag.r.rao.civ@mail.mil US Army's CCDC Army Research Laboratory)

Ananthram Swami (ananthram.swami.civ@mail.mil US Army's CCDC Army Research Laboratory)

Santiago Segarra (segarra@rice.edu Rice University)