Graph-based Deterministic Policy Gradient for Repetitive Combinatorial Optimization Problems

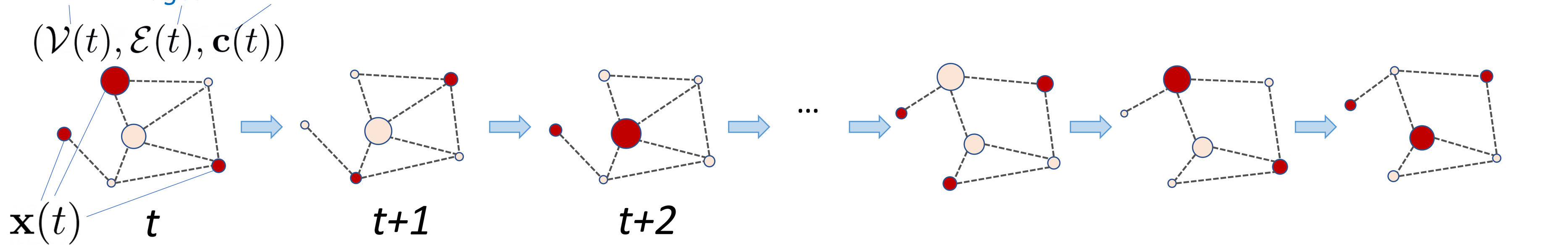
Zhongyuan Zhao*, Ananthram Swami †, Santiago Segarra* (*Rice University, † US Army's DEVCOM Army Research Laboratory, USA)

Paper ID: 4014

Problem: Repetitive Combinatorial Optimization Problems (R-COP)

State: weighted graph

Graph-based Markov decision process (MDP)

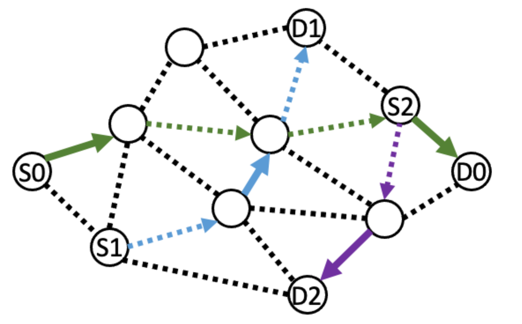


The **state** of t is a weighted graph $(\mathcal{V}(t), \mathcal{E}(t), \mathbf{c}(t))$

$\mathbf{x}(t)$ **Decisions** found by solving a combinatorial optimization problem (COP) on the weighted graph

Exemplary Applications

Routing & Scheduling in **communication networks**



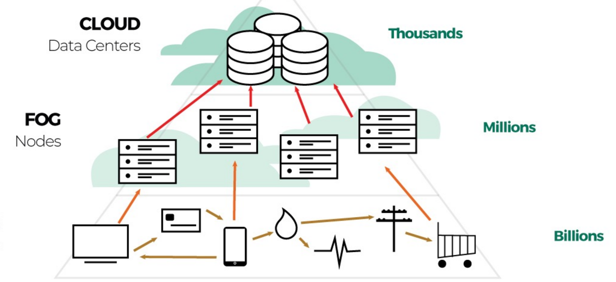
Source: zhongyuanzhao.com

Multi-object tracking in **computer vision**

Vehicle routing problems in **distribution networks**

Source: caliper.com

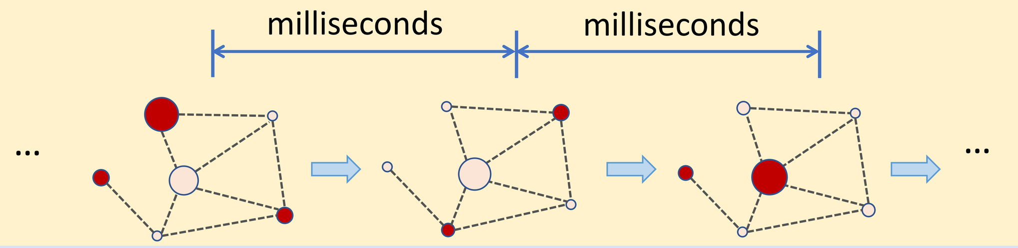
Resource allocation & job scheduling in **cloud, frog, edge computing**



Source: e-ziggurat.com

Characters

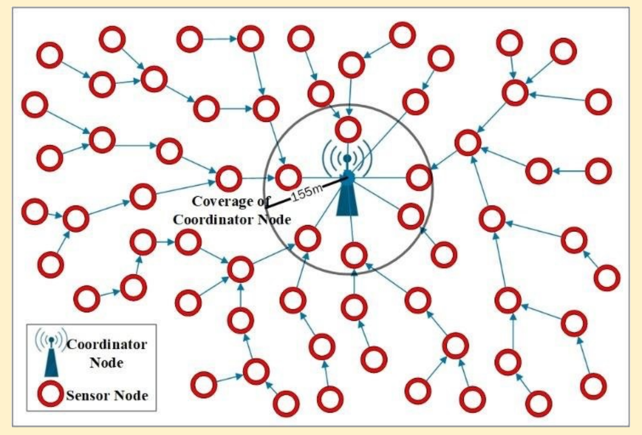
1. Network state of $t+1$ depends on the decisions at t
2. Cost vector $\mathbf{c}$ changes rapidly compared to network topology
3. Dynamic network topology (changes slowly)



Limited runtime: *in many R-COPs, such as wireless link scheduling or computer vision, the COP instances coming at data or video frame rates (tens per second). Centralized / neural solvers are not applicable, have to use fast heuristics (e.g. greedy) in practice*

Challenges & Restrictions

1. Practical: limited runtime
2. Practical: distributed execution
3. Theoretical: long-term goal seeking



Collecting network state to a server

Source: (D. Arı , M. Çibuk and F. Ağgün , 2017)

Why distributed execution?

Centralized COP solver

High communication overhead → Network state changes before being collected to a server

High computational complexity → Scales up quickly by network size

Single-point-of-failure

Distributed COP solver → only needs neighborhood information, fast, robust

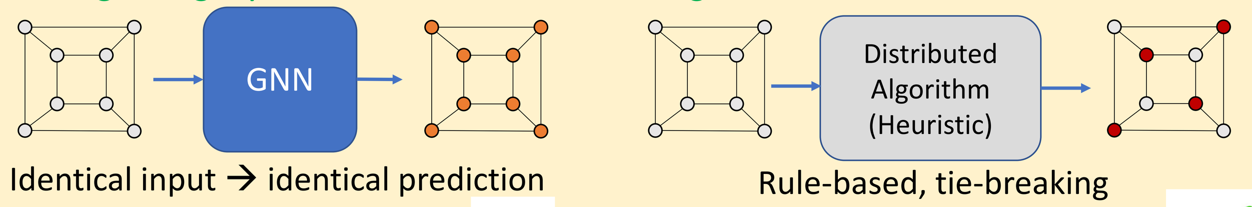
Graph neural networks (GNNs) v.s. Distributed Algorithms

Both: Fast & distributed execution; generalize to different topologies

GNN: learnable but poor in encoding relational constraints

Algorithm: valid solution but large optimality gap

***Counter example**: Maximum weighted independent set (MWIS) problem on a regular graph with identical node weights*



Identical input → identical prediction

Constraint violations

Rule-based, tie-breaking

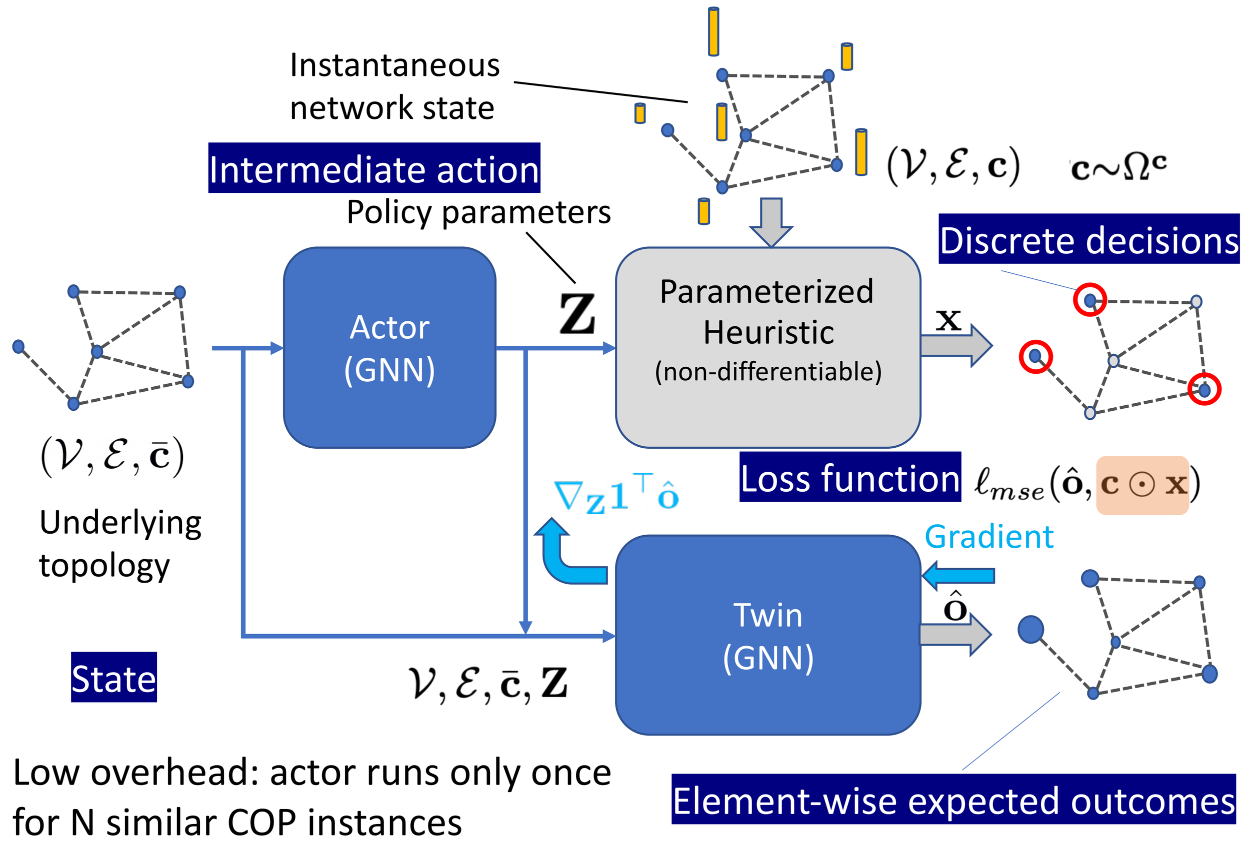
Always find valid solutions

Approach: GDPG-Twin – *a hybrid ML pipeline for distributed control in networks*

1. Independent R-COP

Optimize each instance individually

Goal: reduce optimality gap with minimal overhead



Low overhead: actor runs only once for N similar COP instances

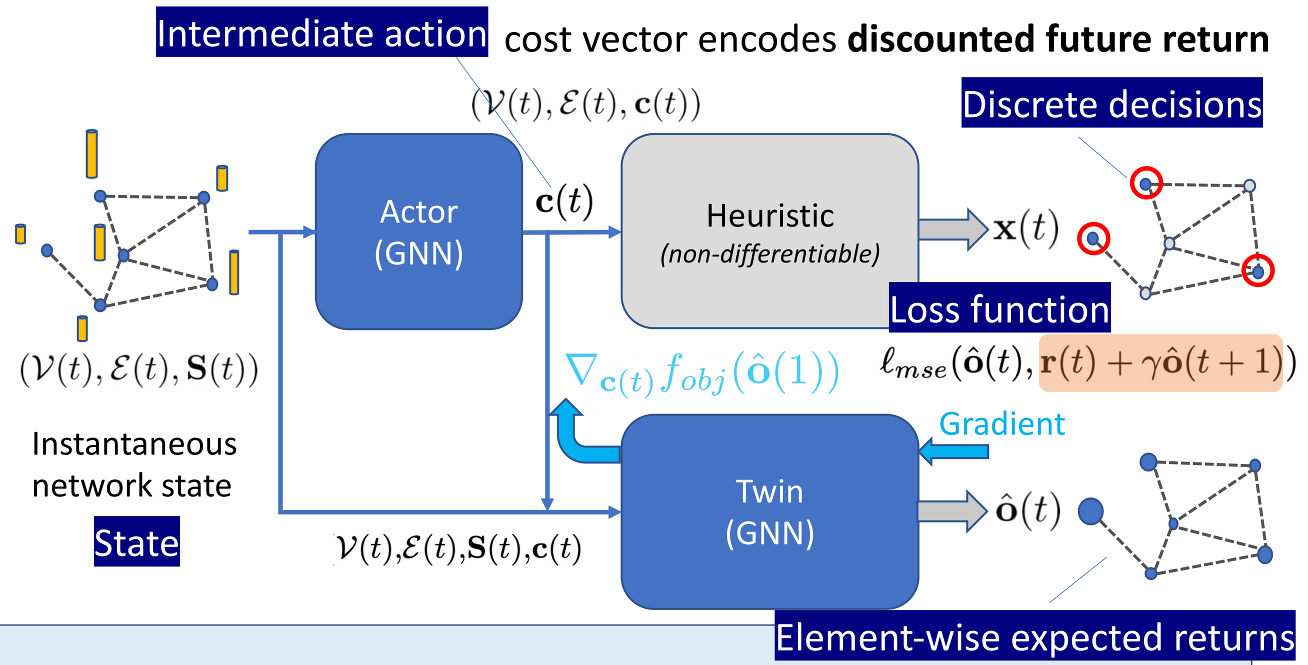
Key features

1. All distributable components (Actor, Critic, Heuristic)
2. Pipeline can generalize to different network topologies
3. **Constraints guaranteed by the heuristic**
4. Future returns encoded in cost vector input to heuristic

2. R-COP in graph-based MDP

Inter-state dependency MUST be considered

Goal: achieve long-term system-level objective



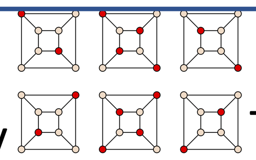
Key novelty

1. *Defines **vectorized (element-wise) reward and return** for network settings*
2. *Critic based on a **twin network** that predicts element-wise expected returns/outcomes*

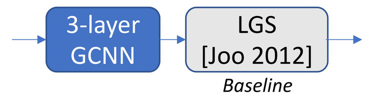
Test Results

Independent R-COP

Maximum Weighted Independent Set



Tested on four different R-COPs



C. Joo and N. B. Shroff, "Local Greedy Approximation for Scheduling in Multihop Wireless Networks," in *IEEE Transactions on Mobile Computing*, vol. 11, no. 3, pp. 414-426, March 2012, doi: 10.1109/TMC.2011.33.

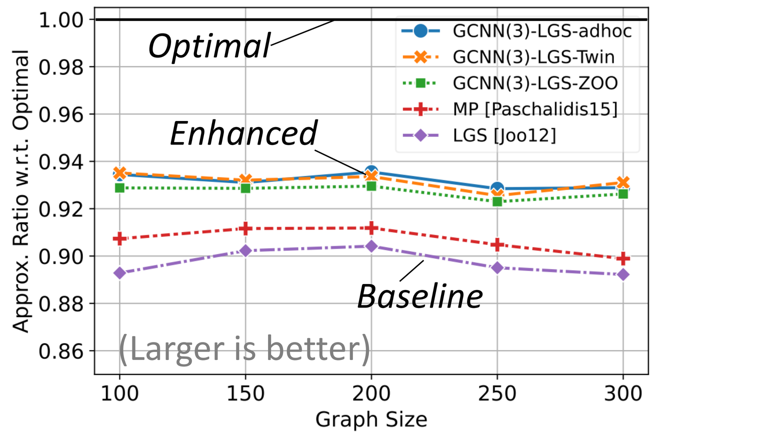


Figure 1: Approximation ratios (Larger is better) of the vanilla and GCNN-enhanced distributed heuristics for MWIS problem (max), w.r.t. the optimal solver.

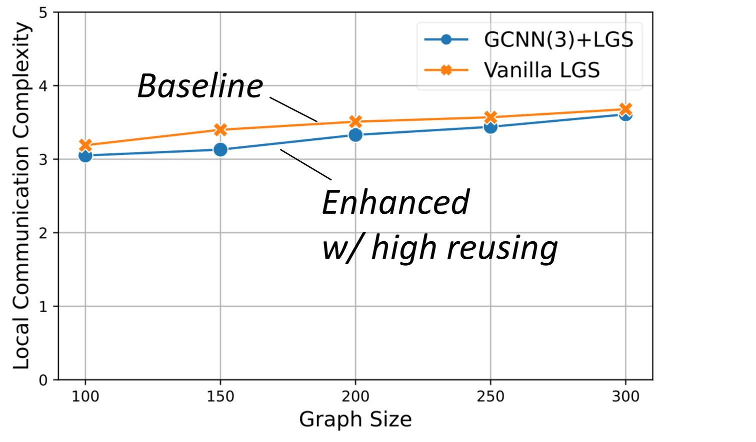


Figure 3: Average local communication complexity of GCNN-enhanced and vanilla LGS-MWIS solvers per instance, in rounds, excluding the GCNN ($N = \infty$).

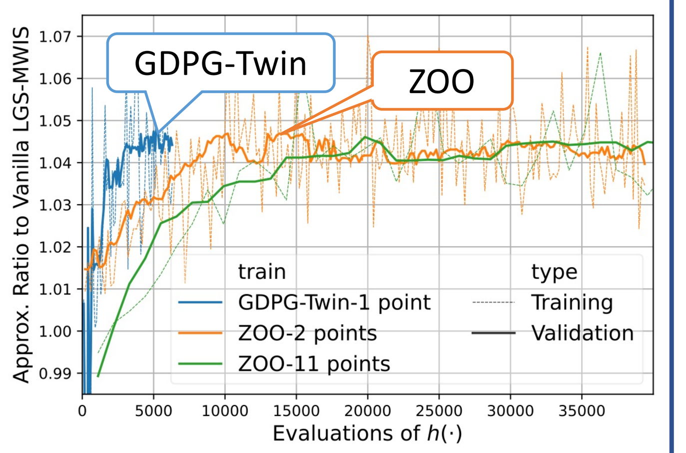
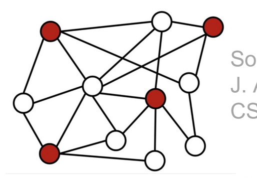


Figure 8: Performance trajectories of GCNN-enhanced LGS-MWIS trained by GDPG-Twin and ZOOs with 2-point and 11-point gradient estimations. Larger is better. GDPG-Twin needs fewer evaluations of $h(\cdot)$.

GDPG-Twin

1. Generalized ML pipeline for different R-COPs
2. Close optimality gap of fast and distributed heuristics under different graph distributions
3. Slightly reduce execution time with high reusing factor
4. Improve training efficiency by 2~3 times compared to Zeroth-order optimization (ZOO)

Minimum Weighted Dominating Set



Source: J. Abernethy, CS 3510

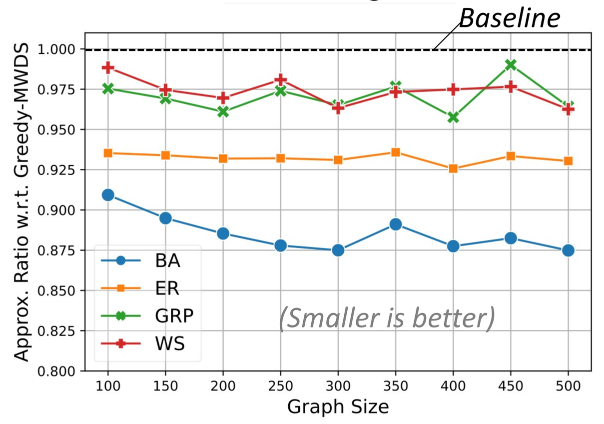
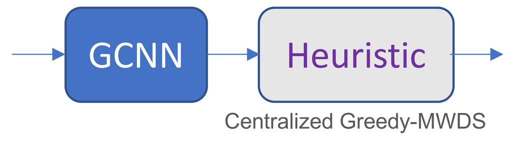
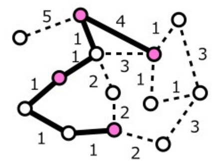


Figure 2: Approximation ratio (Smaller is better) of the GCNN-enhanced w.r.t. the vanilla Greedy-MWDS for MWDS problem (min) on 4 sets of random graphs.



Node Weighted Steiner Tree



Source: (M. Fujita, T. Kimura, & K. Jin'no, 2016)

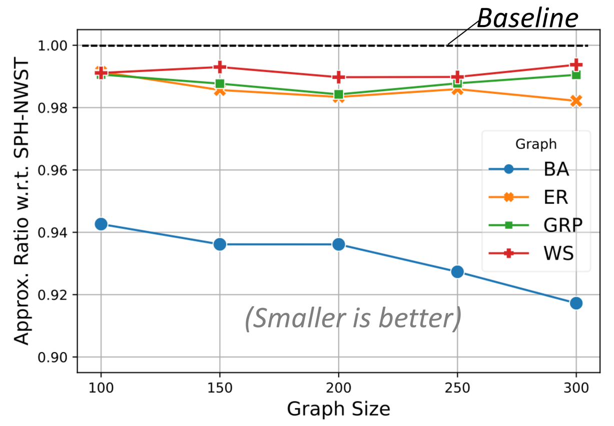
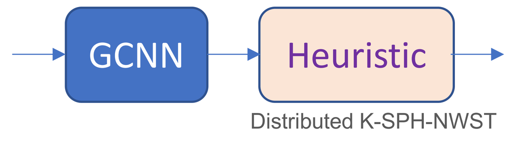
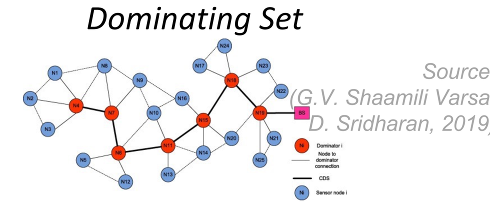


Figure 5: Approximation ratio (Smaller is better) of the GCNN-enhanced w.r.t. vanilla K-SPH-NWST for NWST problem on 4 sets of random graphs. NWST is a minimization (min) problem.



Minimum Weighted Connected Dominating Set



Source: (V. Shaamili Varsa, D. Sridharan, 2019)

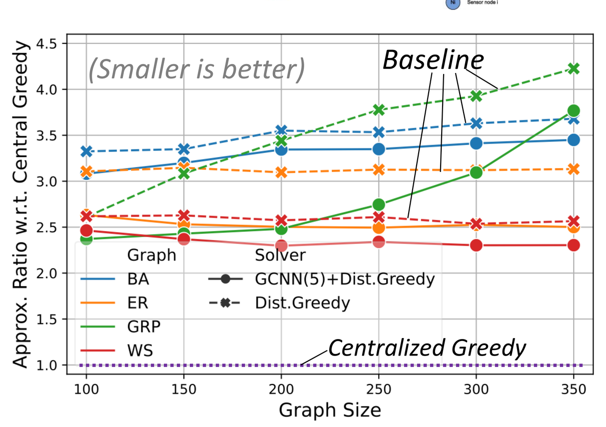
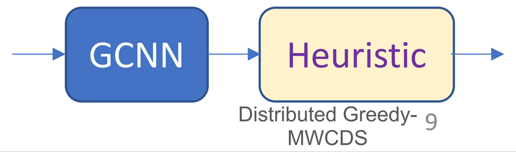
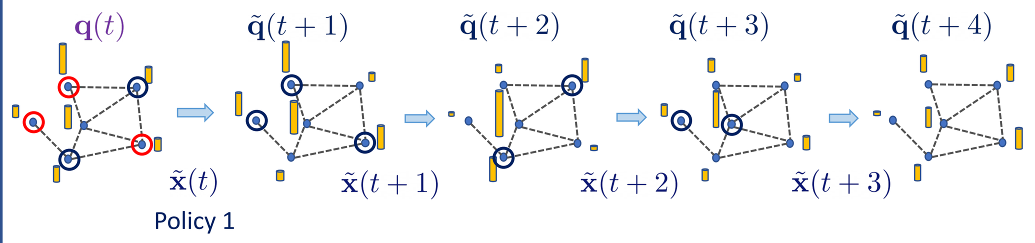


Figure 6: Approximation ratios (Smaller is better) of the vanilla and GCNN-enhanced distributed heuristics w.r.t. a centralized heuristic for MWCDS problem on 4 sets of random graphs. MWCDS is a min. problem.



R-COP in graph-based MDP: Delay-oriented link scheduling*



$$\mathbf{q}(t+1) = \mathbf{q}(t) + \mathbf{a}(t) - \mathbf{x}(t) \odot \min(\mathbf{l}(t), \mathbf{q}(t))$$

Minimize $\mathbb{E}_{i \in \mathcal{V}, t \leq T}\left[\mathbf{q}_i(t)\right]$

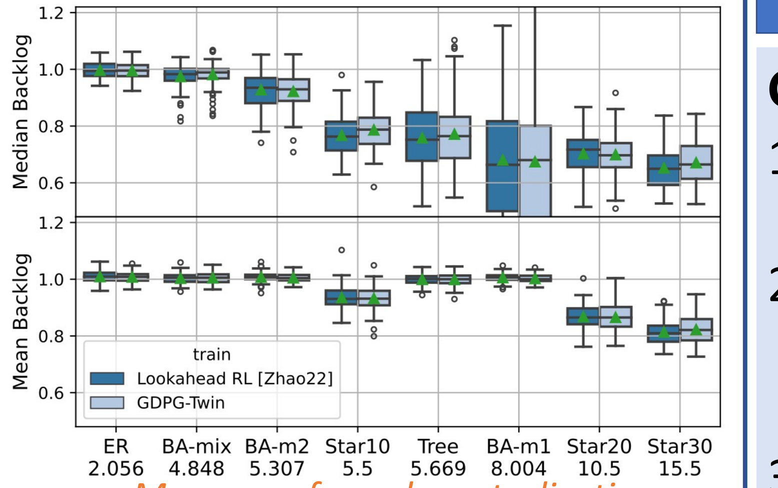


Figure 7: GDPG-Twin achieves similar network-wide mean and medium backlogs (smaller is better) of lookahead RL (Zhao et al., 2022b) in training a distributed link scheduler, using only 1/5 evaluations of $h(\cdot)$ of it.

GDPG-Twin can

1. enable long-term goal seeking for R-COP in MDP
2. do the same job of an ad-hoc RL scheme* at 1/5 computational cost

* Z. Zhao, G. Verma, A. Swami and S. Segarra, "Delay-Oriented Distributed Scheduling Using Graph Neural Networks," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Singapore, Singapore, 2022, pp. 8902-8906, doi: 10.1109/ICASSP43922.2022.9746926.

Conclusion

GDPG-Twin

1. A general actor-critic framework
2. Applicable to R-COPs with limited runtime and distributed execution
3. Reduce optimality gap with min overhead
4. Enable long-term goal seeking
5. *Beyond COP → applicable to general network processes*