Heterogeneous Graph Transformers for Simultaneous Mobile Multi-Robot Task Allocation and Scheduling under Temporal Constraints

Batuhan Altundas, Shengkang Chen,
Shivika Singh, Shivangi Deo,
Minwoo Cho and Matthew Gombolay

CORE Robotics Lab

NEURAL INFORMATION PROCESSING SYSTEMS

Georgia Tech Robotics & Intelligent Machines

Georgia Tech

Why Multi-Agent Teams are important?

Warehouse Management

Manufacturing

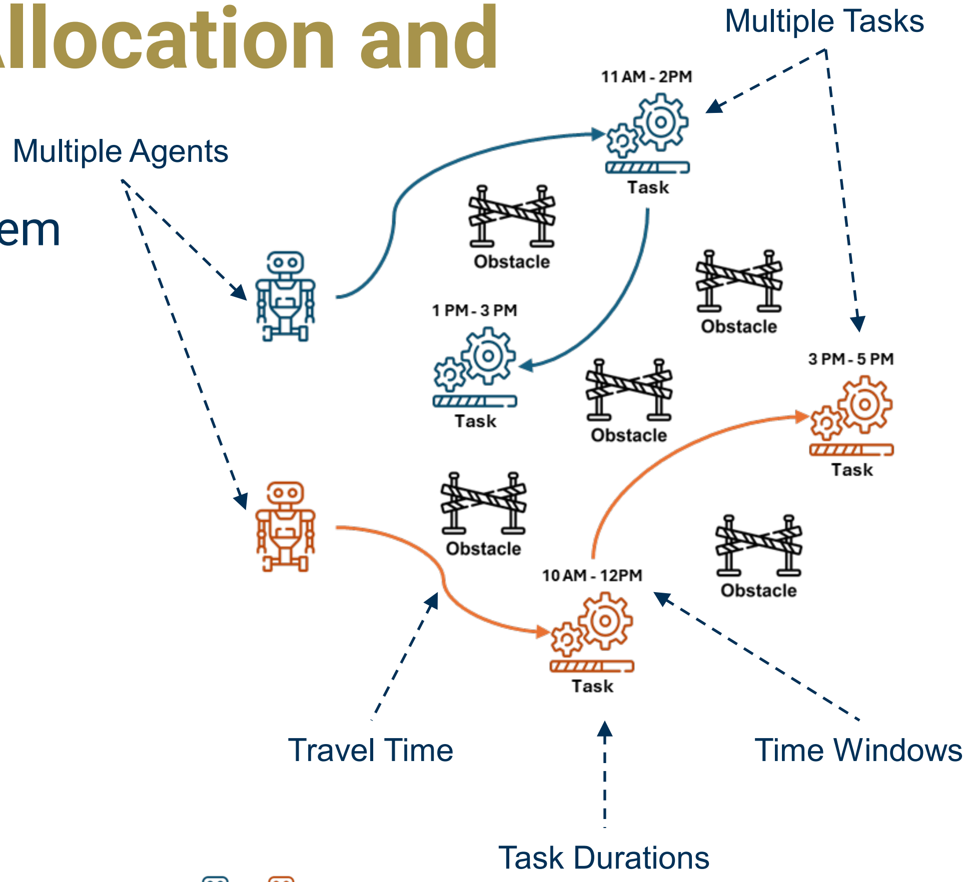
Logistics

Operations

Multi-Agent Coordination as a Task Allocation and Scheduling Problem

- Task Allocation and Scheduling is an NP-Hard Optimization Problem
- Heuristics are suboptimal in large scales
- Metaheuristics leverage search and are compute intensive
- Prior Learning Based Approaches:
  - Subset of the Problems
  - Subset of Constraints
  - Utilize Sequential Assignment

	Task Duration	Travel Time
Min-Max mTSP (Hu et al. 2020) [1]		✓
GPN (Ma et al. 2019) [2]		✓
ALMA (Iqbal et al 2022) [3]	✓	✓
HomGNN (Wang et al. 2019) [4]	✓	
ScheduleNet (Wang et al. 2022) [5]	✓	
HybridNET (Wang et al. 2022) [6]	✓	
Ours	✓	✓



Single-Task, Single-Agent, Time-extended Assignment (ST-SA-TA) Problems according to Korsah et al. 2013[7]

[1] Yujiao Hu, Yuan Yao, and Wee Sun Lee. A reinforcement learning approach for optimizing multiple traveling salesman problems over graphs. Knowledge-Based Systems, 204:106244, September 2020. ISSN 09507051. doi: 10.1016/j.knosys.2020.106244.
[2] Ma, Qiang, Suwen Ge, Danyang He, Darshan D. Thaker and Iddo Drori. "Combinatorial Optimization by Graph Pointer Networks and Hierarchical Reinforcement Learning." ArXiv abs/1911.04936, 2019.
[3] Iqbal, S., Costales, R., & Sha, F. (2022). Alma: Hierarchical learning for composite multi-agent tasks. Advances in neural information processing systems, 35, 7155-7166.
[4] Zheyuan Wang and Matthew Gombolay. Learning to Dynamically Coordinate Multi-Robot Teams in Graph Attention Networks. arXiv preprint arXiv:1912.02059
[5] Zheyuan Wang, Chen Liu, and Matthew Gombolay. Heterogeneous graph attention networks for scalable multi-robot scheduling with temporospatial constraints. Autonomous Robots, 46 (1):249–268, January 2022. ISSN 1573-7527. doi: 10.1007/s10514-021-09997-2
[6] Batuhan Altundas, Zheyuan Wang, Joshua Bishop, and Matthew Gombolay. Learning Coordination Policies over Heterogeneous Graphs for Human-Robot Teams via Recurrent Neural Schedule Propagation. IROS 2022, pages 11679–11686, Kyoto, Japan, October 2022. IEEE. ISBN 978-1-66547-927-1. doi: 10.1109/IROS47612.2022.998174
[7] G. A. Korsah, A. Stentz, and M. B. Dias, "A comprehensive taxonomy for multi-robot task allocation," *The International Journal of Robotics Research*, vol. 32, no. 12, pp. 1495–1512, Oct. 2013, doi: 10.1177/0278364913496484.

Contributions

1. A Method of representing TARGETNET, a Heterogenous Mobile Multi-Agent Task Allocation and Scheduling using a relational Graph Model.
2. Integrate Edge Features and Attention into state-of-the-art Heterogenous Graph Transformers for the use of Optimization Problems.
3. Simultaneous Decision-Making using Graph-based Networks with Edge Features that allow for:
   - 13% better than the best performing metaheuristics in less than 1% of the time
   - 36% better performance than state-of-the-art GNN-based schedules that use Sequential Decision-Making in 5% the time in 10 agent-20 task problems
   - 232% better than partial schedules generated by exact solvers after 12 hours in less than 12 seconds, in 40 agent-200 task problems

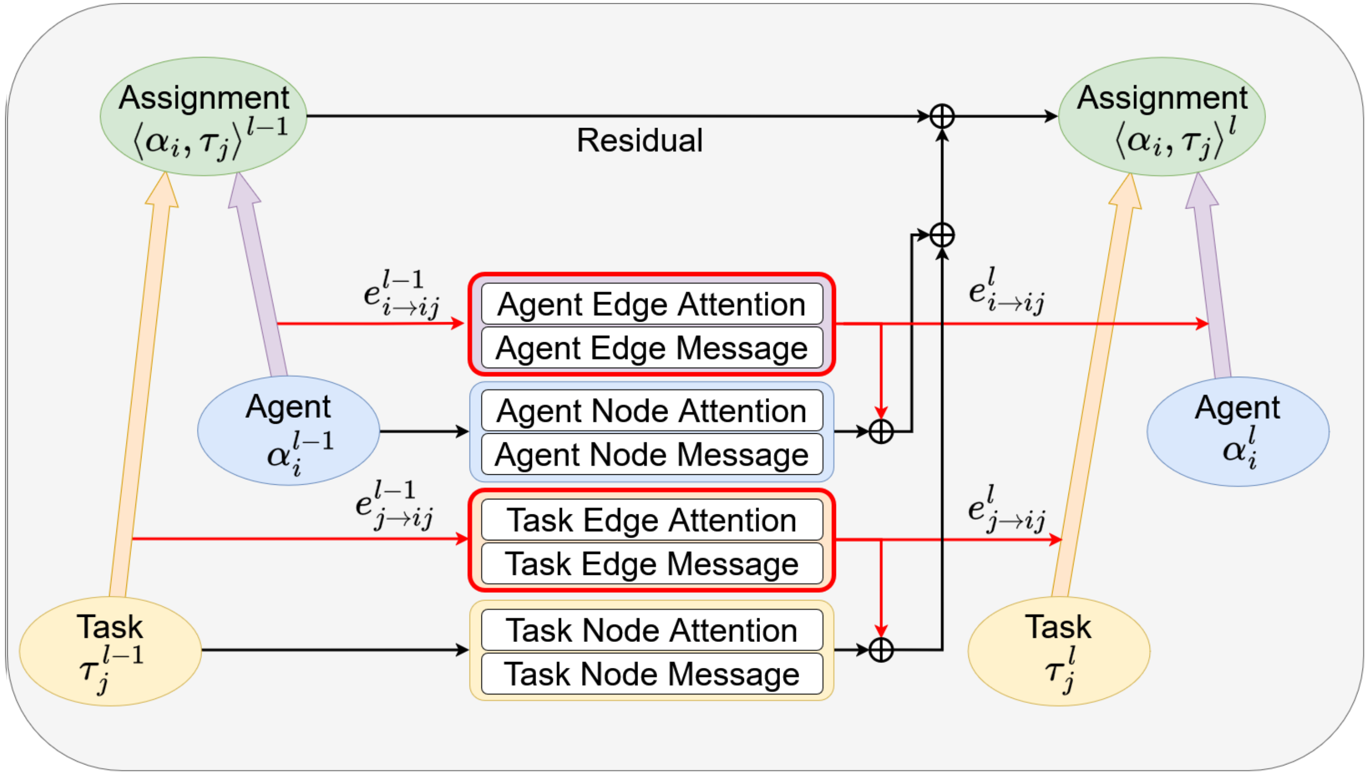
Integrating Relational Information through Edge Attention

Nodes store unique information

Edges store relational information

- e.g. Heterogenous Travel Duration from current location of Agent i to the location of Task j.

We expand on prior work to allow for Edge Attention and Message Passing in Heterogenous Graph Transformers



Update Node: $H^l[t] \xleftarrow[\substack{\forall s\in N(t)\\ \forall e\in E(s,t)}]{} Aggregate\big(Att_N(s,t)\cdot Msg_N(s) + Att_E(e,t)\cdot Msg_E(e)\big)$

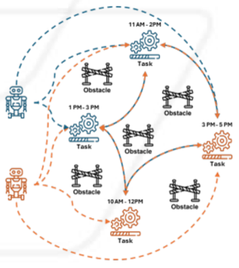
Update Edge: $H^l[e] \xleftarrow[\forall e\in E(s,t)]{} Aggregate\left(Att_N^E(s,t)\cdot Msg_N^E(s) + Att_E^E(e,t)\cdot Msg_E^E(e)\right)$

Agent α_i^l — Node Features of Agent i for Layer l

Task τ_j^l — Node Features of Task j for Layer l

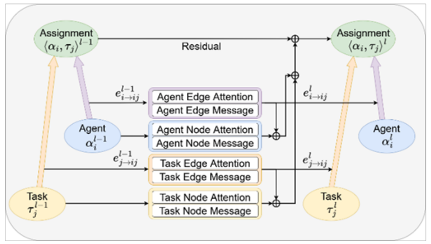
Assignment $\langle \alpha_i, \tau_j \rangle^l$ — Node Features of Assignment of Agent i to Task j for Layer l

Simultaneous Decision Making



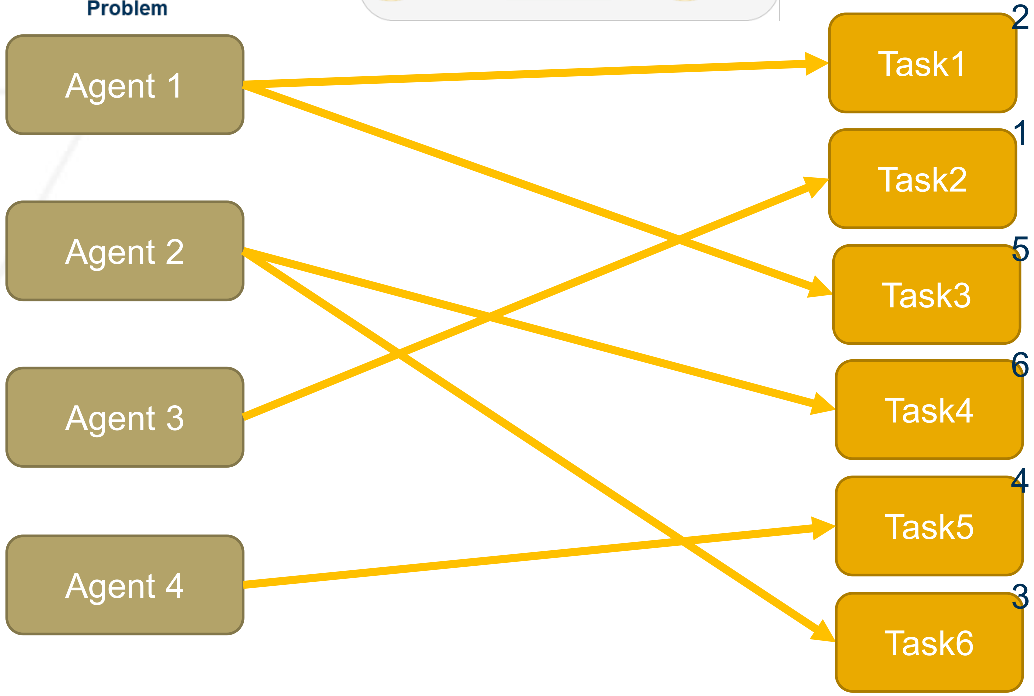
Problem

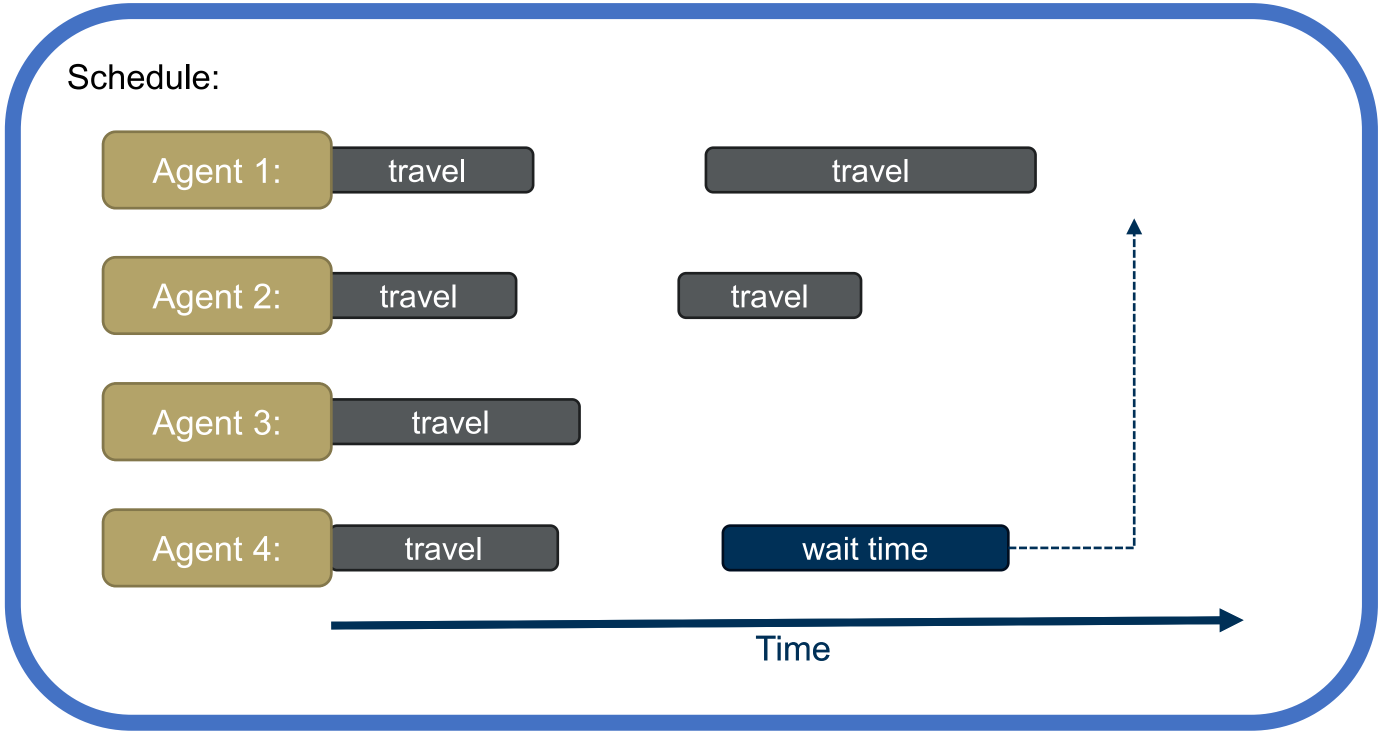
TARGETNET



$$\begin{array}{cc} & \text{Tasks} \\ \text{Agents} & \begin{bmatrix} x_{00} & \cdots & x_{0j} \\ \vdots & \ddots & \vdots \\ x_{i0} & \cdots & x_{ij} \end{bmatrix} \end{array}$$

Binary Assignment Matrix





Step 1: Assign an Agent to each Task.
Step 2: Determine the Order of Tasks
Step 3: Construct and execute Schedule

Agent	Agent Node contains Agent Info (eg. Location, Speed)
Task	Task Node contains Task Info (eg. Location, Time Window)
travel	Travel Time from Agent's Current Location to Task Location
wait time	Constraint for Wait Time between execution of two Tasks

Georgia Tech

Policy Training using REINFORCE[1]

For the Task Allocation and Scheduling of $|T|$ tasks to $|A|$ agents:

- Policy $\pi(a_t|s_t) = \pi_{TaskAllocation}(\langle\alpha_i, \tau_j\rangle|s_0) \times \pi_{Schedule}\left(\tau_{j_t}|\alpha_i, s_0\right)$ where,
  - $\pi_{TaskAllocation}$ is Task Allocation Policy. Assigns an agent to each Task
  - $\pi_{Schedule}$ is the Scheduling policy. Determines order of Tasks.
- Combined generates a full schedule $\{a_t\}_{t\in\{1:|T|\}}$ where $a_t = \langle\alpha_i, \tau_j\rangle_t$ is the assignment of Agent, α_i , to Task, τ_i, at time step, t.

$$\text{Reward: } R_t = \begin{cases} \left(1 - \frac{t_{ms}}{t_{max}}\right) + \sum_{i=1}^{T} \mathbb{1}_{feasible}(s_i, a_i) & \text{, if } t = t_{max} \\ +1 & \text{, if feasible} \\ -1 & \text{, otherwise} \end{cases}$$

Task Allocation and Scheduling Policy outputs a probability mass function: $\pi(a_t|s_t)$

$$Loss_{Policy} = \left(\sum_{t=1}^{|T|} \gamma^{|T|-t} R_t\right) \log \pi(a_t|s_t)$$

Discount factor: γ

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, vol. 1, no. 1.

Experiments

Training:

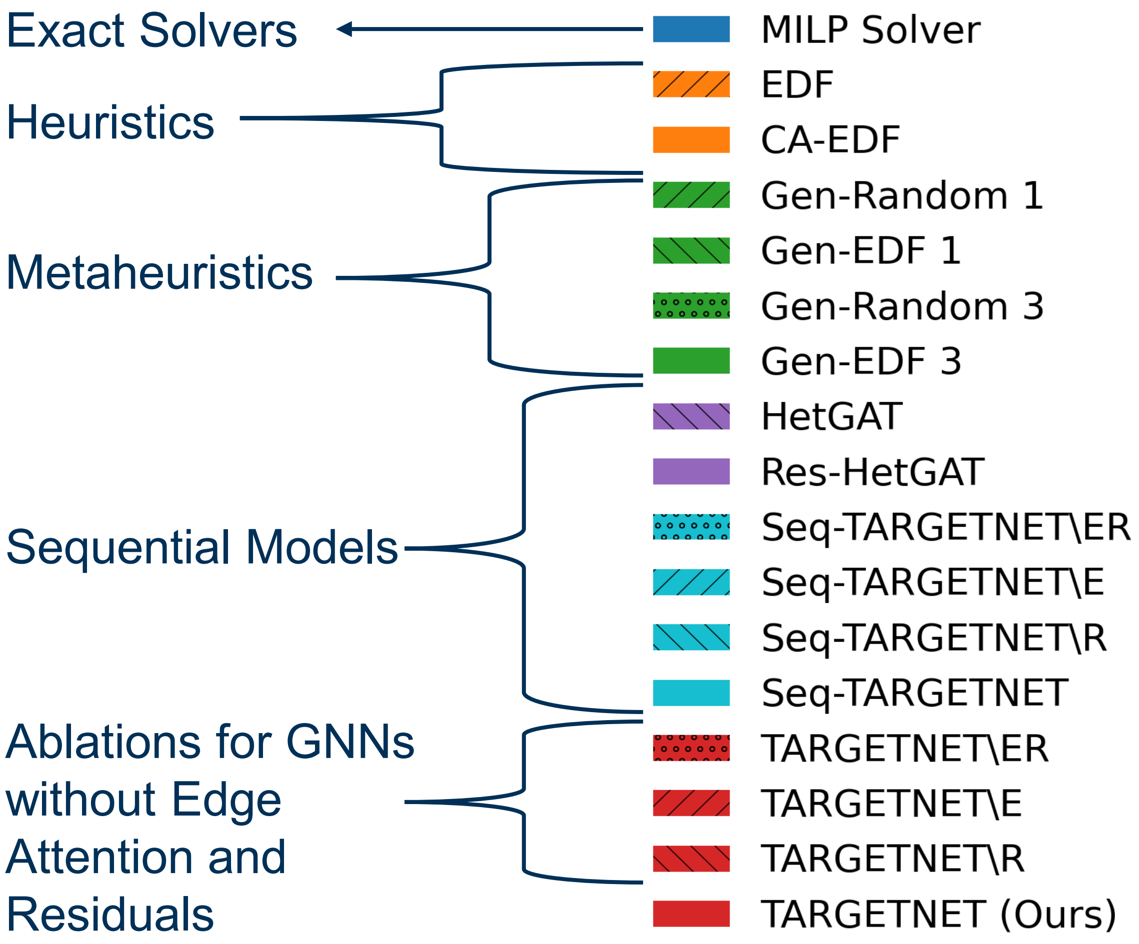
- 200 Small-Scale Problems

Testing on:

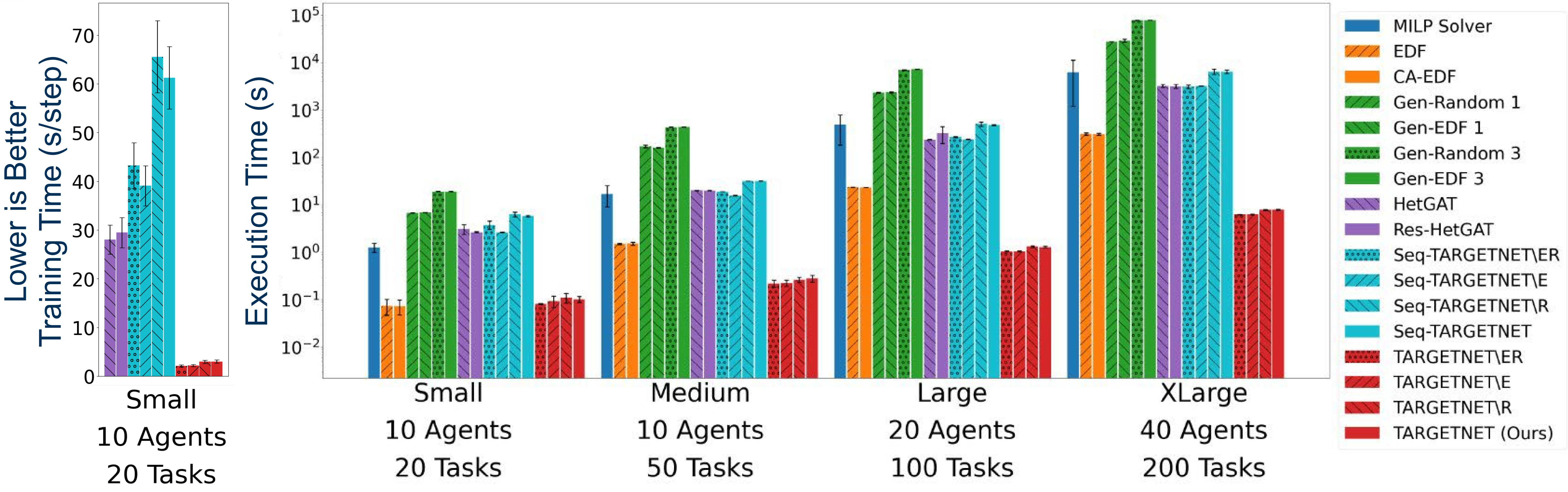
- Small: 10 Agents, 20 Tasks
- Medium: 10 Agents, 50 Tasks
- Large: 20 Agents, 100 Tasks
- Xlarge: 40 Agents, 200 Tasks

Metrics:

- Schedule Quality Relative to the MILP Solver $= \frac{R_{policy}}{R_{MILP}} \times 100\%$
- Feasibility Percentage = $\frac{Num\ Feasible}{Num\ Task} \times 100\%$
- Training and Computation time

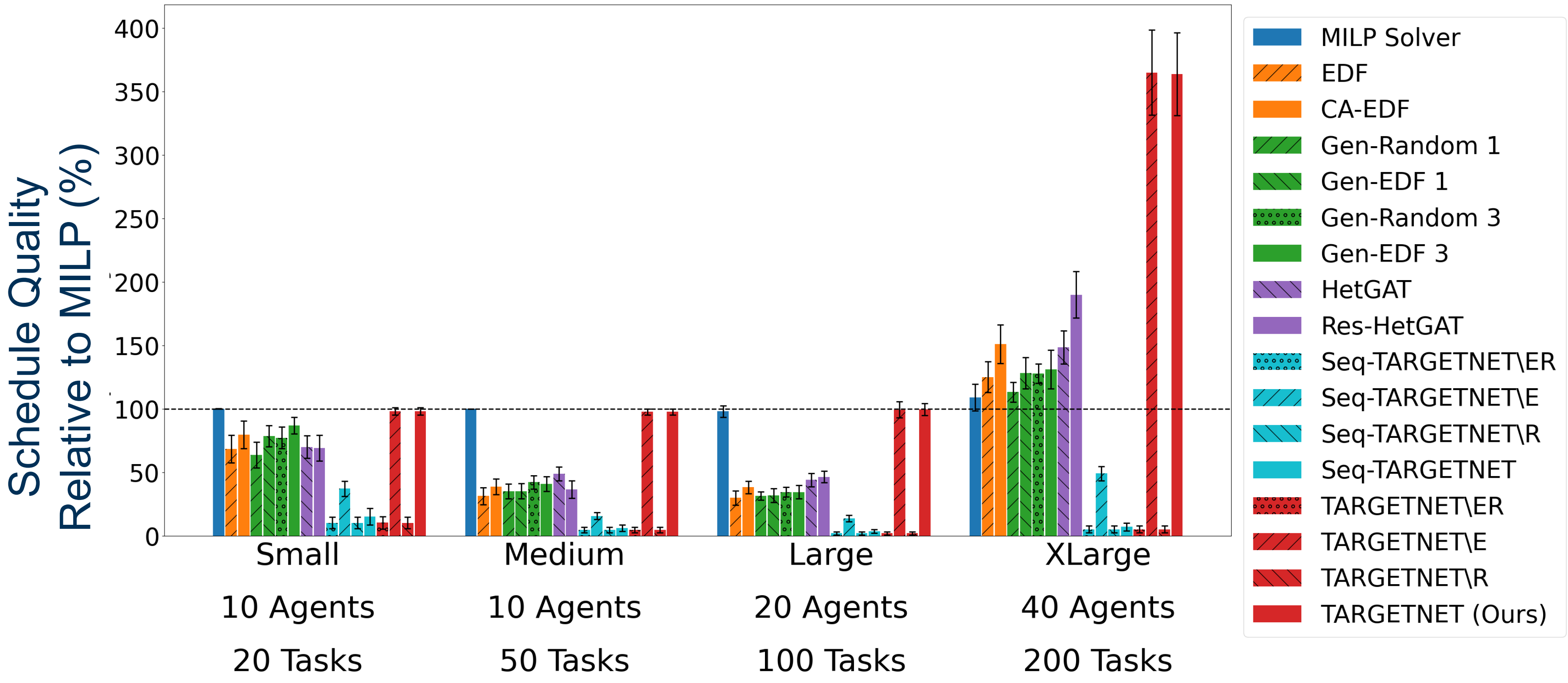


TARGETNET Learns Faster, and Executes Faster



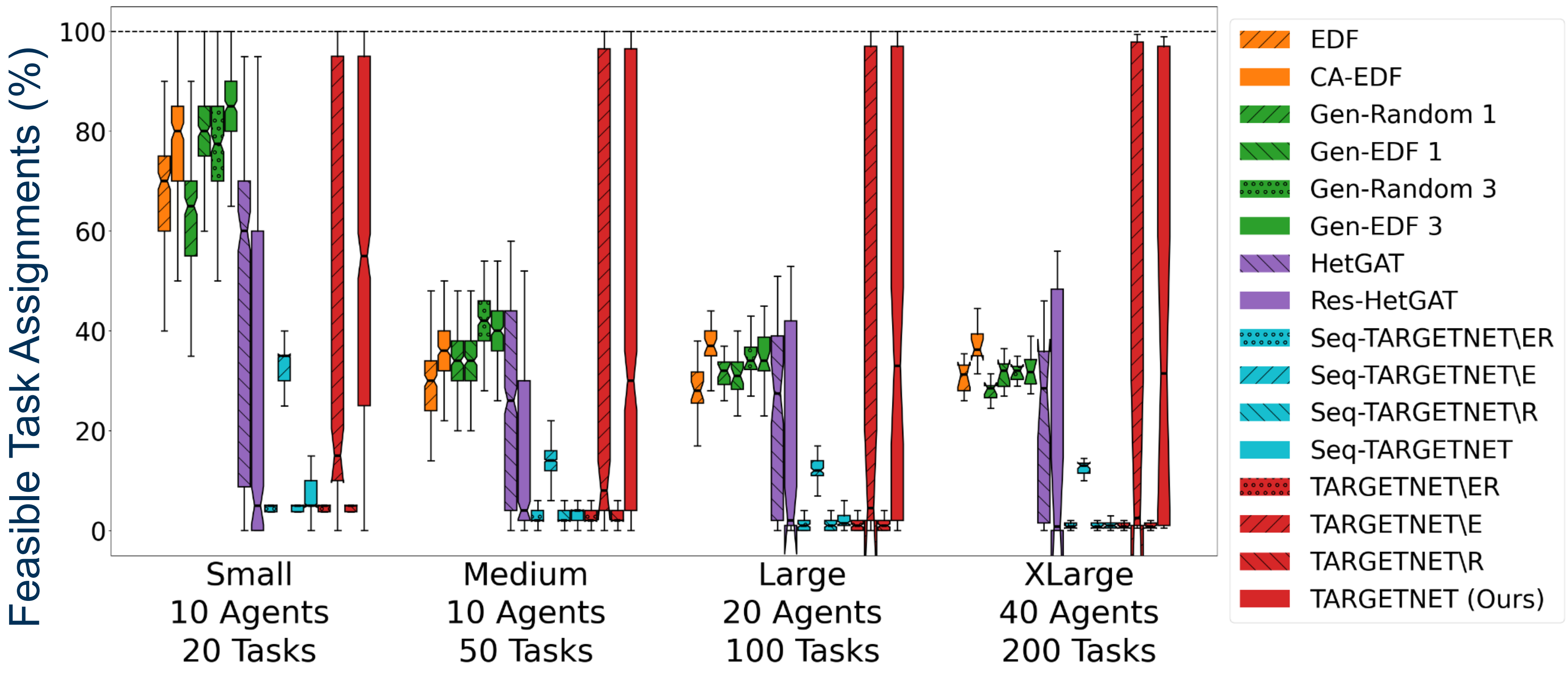
Lower is Better

TARGETNET Generates Schedules of Higher Quality than Baselines



Higher is Better

TARGETNET Produces more feasible schedules than baselines



Higher is Better

Contributions

We present a model that can learn to approximate a scheduling policy that in 1 pass can produce a schedule.

We do this through representing the problem in a Graph Model, before passing it through Heterogenous Graph Transformers with Edge Attention.

- Naval Research Laboratory (NRL) grant numbers N00173-21-1-G009 and N00173-25-1-0050

- Office of Naval Research (ONR) under grant number N00014-23-1-2887