Graph Reinforcement Learning for Resilient Supply Chain Network Design

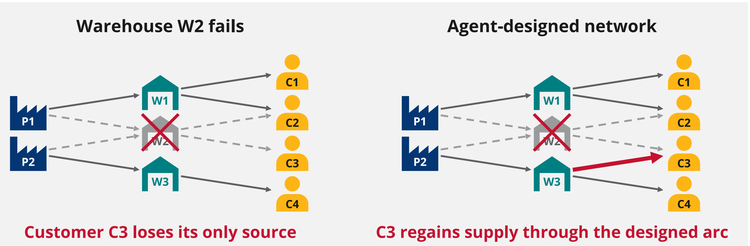
Nicolas Smits, Arsh Bhatia, Carl D. Laird
Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA nsmits@andrew.cmu.edu

Carnegie Mellon University
Chemical Engineering

A design policy trained on networks of at most 19 nodes was applied without modification to networks of up to 70 nodes. Without running the simulation at inference time, the trained policy successfully retained 64% to 80% of the value of greedy simulation-based search and designed a 70-node network in approximately **one CPU-second**, compared with approximately **six CPU-hours** for greedy search.

Motivation

A global chemical manufacturer moves product from plants through warehouses to customers. When a facility or a transport link fails, the demand that it served is delivered late or not at all [1, 2]. Additional transport arcs create alternative supply paths, each at a cost.



Gap in existing methods

1. **Simulation-based search scales poorly.** Greedy search simulates every candidate arc over many failure scenarios. Even with the simple inventory model used here, one 70-node design takes six CPU-hours.
2. **Learned graph construction has not addressed supply chains.** Size generalization has been studied for routing problems such as the TSP [3] and for the robustness of abstract graphs [4], not for networks valued through inventory operations.
3. **Node and arc failure probabilities are seldom known.** A deployable policy must act on routine operational records alone.

Environment: Simulation Step

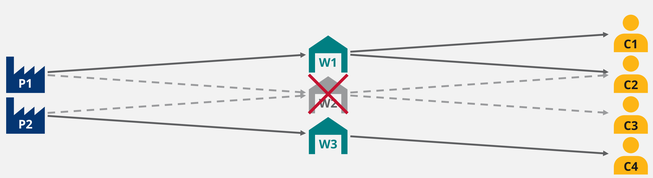
One period t of operations on a fixed network G under a failure scenario ω, repeated for $t = 1, \dots, T$ with $T = 30$.

Node quantities

κ_p production capacity

$I_{w,t}, P_{w,t}, s_w < S_w$ stock, pipeline, reorder point, capacity

$D_{c,t} \sim \text{Pois}(\mu_c), U_{c,t}$ demand, backlog



Arc and failure quantities

$L_{ij}, \nu_{ij}, F_{ij,t}$ lead time, priority, shipment

$\omega_e \sim \text{Bern}(\pi_e)$ failure of component e, fixed for all t

Reorder $R_{w,t} = S_w - I_{w,t-1} - P_{w,t-1}$

(s, S) rule if $I_{w,t-1} + P_{w,t-1} < s_w$

Lead time $L_{ij} = \max\left(1, \text{round}\left(\beta_{ij} d_{ij} / \bar{d}\right)\right), \beta_{ij} \sim \mathcal{N}(3, 0.5^2)$

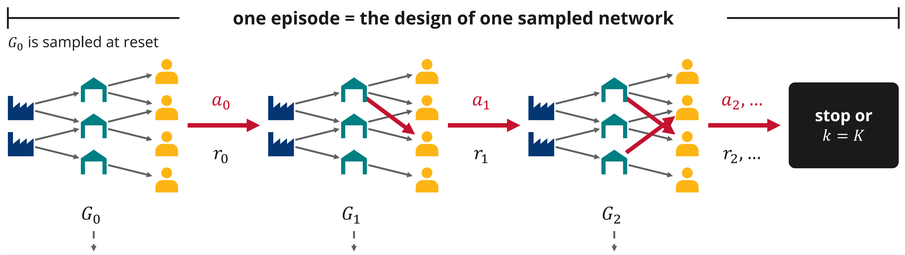
Priority $\nu_{ij} \in \{1,2,3\}$

Backlog $U_{c,t} = \max\left(0, D_{c,t} + U_{c,t-1} - F_{c,t}\right)$

Total $b(G, \omega) = \sum_c U_{c,T}$

Environment: Design Step

One arc a_k added to the network G_k. Every expected backlog $B(G_k)$ requires $N \times T$ simulation steps.



after every design step, $B(G_k) = \frac{1}{N} \sum_\omega b(G_k, \omega)$ over N failure scenarios of T simulation steps

Reset G_0 drawn from the training sizes

State $s_k = [X, E, \mathcal{A}_k, \rho_k]$

Action $a_k \in \mathcal{A}_k \cup \{\text{stop}\}$

Transition $G_{k+1} = G_k + a_k$

Termination stop, or $k = K = 5$

Discount $\gamma = 1$

Reward of one design step

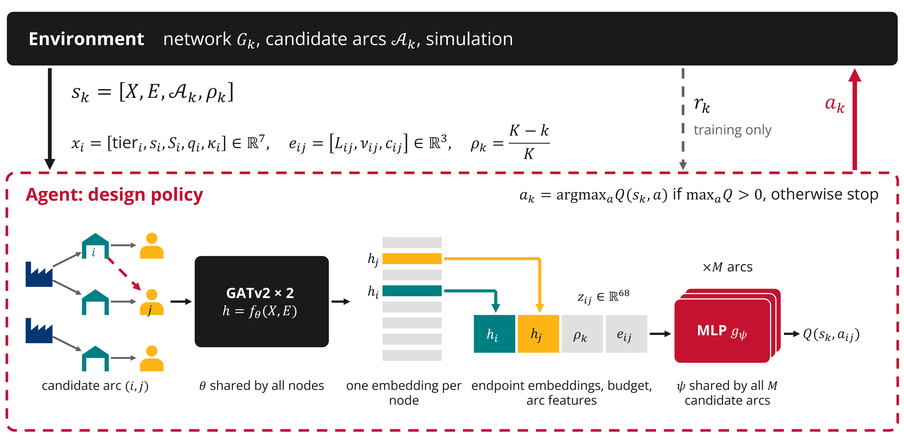
$$r_k = \frac{B(G_k) - B(G_{k+1})}{B(G_0)} - \lambda\, c(a_k)$$

Return of the episode = design value

$$\sum_{k=0}^{K-1} r_k = \frac{B(G_0) - B(G_K)}{B(G_0)} - \lambda \sum_k c(a_k)$$

Free of scale by construction. The service gain is a fraction of the baseline backlog $B(G_0)$, and the arc cost $c(a) = \frac{c_a}{\bar{c}} \cdot \frac{c_{ref}}{C}$ is relative to the mean candidate arc and to the number of customers C, with $\lambda = 0.05$.

Design Policy



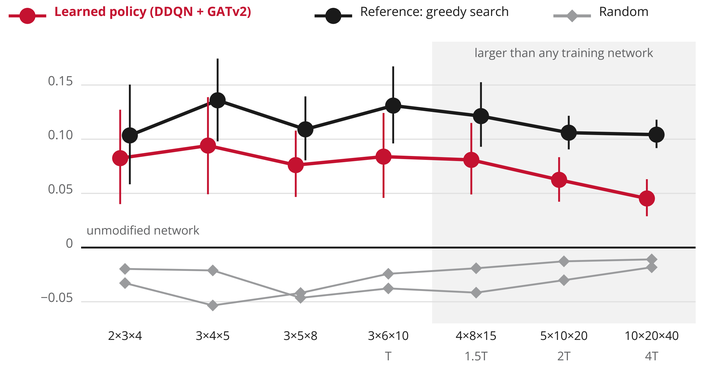
28,481 parameters $|\theta| + |\psi|$ at every network size, from 9 to 70 nodes

Preliminary Results

Five training seeds and twenty held-out networks per size.



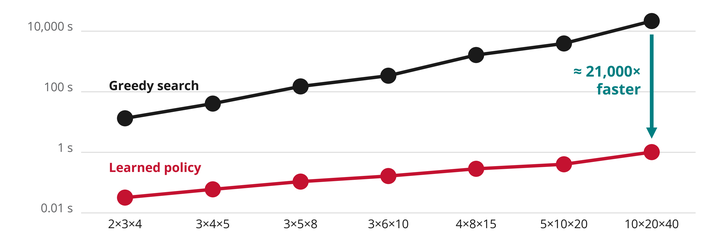
Held-out design value (mean and 95% interval)



Transfer. The policy selects arcs without running the simulation. It nevertheless exceeded the unmodified network and both random baselines at every size, with every lower interval bound above zero, and retained most of the value of greedy search up to 2T.

Share of the reference. 64% to 80% at the training sizes [and 43% at 4T].

CPU time to design one network (median)



Limitations and next steps

- With denser sourcing than in training, the policy added arcs of little value and fell below the unmodified network.
- Current work aims to make the policy robust to changes in network connectivity.
- Validation value was still rising at 3,000 episodes, which indicates that the policies were not fully trained. Longer training and a full hyperparameter search are planned.

[1] Snyder et al., IIE Trans. 48 (2016). [2] Simchi-Levi et al., Interfaces 45 (2015). [3] Joshi et al., Constraints 27 (2022). [4] Darvariu et al., Proc. R. Soc. A 477 (2021). Related work: Li et al. (Think4SCND), IEEE Access 12 (2024); Kotecha and del Rio Chanona, arXiv:2410.18631 (2025); Hubbs et al. (OR-Gym), arXiv:2008.06319 (2020); Burtea and Tsay, Comput. Chem. Eng. 181 (2024); Rolf et al., Int. J. Prod. Res. 61 (2023); Gao and Schweidtmann, Curr. Opin. Chem. Eng. 44 (2024); Brown et al., Mater. Des. 218 (2022); Sutton and Barto, Reinforcement Learning, MIT Press (2018). Methods: GATv2, Brody et al., ICLR (2022); Double DQN, van Hasselt et al., AAAI (2016).

Preliminary results, not for citation.