UNIVERSITY of PENNSYLVANIA

State-Augmented Information Routing in Communication Systems with Graph Neural Networks

Sourajit Das

Dept. of Electrical and Systems Engineering

University of Pennsylvania

Email: dassour@seas.upenn.edu

Web: https://sites.google.com/view/sourajitdas

Co-Authors: Navid NaderiAlizadeh, Alejandro Ribeiro

The Next Generation Communication Networks

- Modern communication networks have evolved massively with the use of intelligent systems and smart devices → 5G, WiFi 6



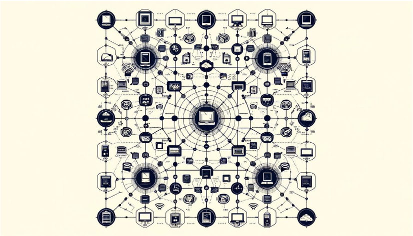


Fig. 1: Generated by OpenAI's ChatGPT DALL ·E, Next Generation Communication Systems

- Autonomous systems can solve complexity and scalability issues in large-scale communication systems: 6G, WiFi 7

- An autonomous communication system is one which makes decisions without human intrusion

AI for Autonomy

- Making decision in communication routing involves solving large-scale constrained network optimization problems.

- Heuristic methods can solve such challenging problems but fail to adapt and generalize beyond a range of operating conditions

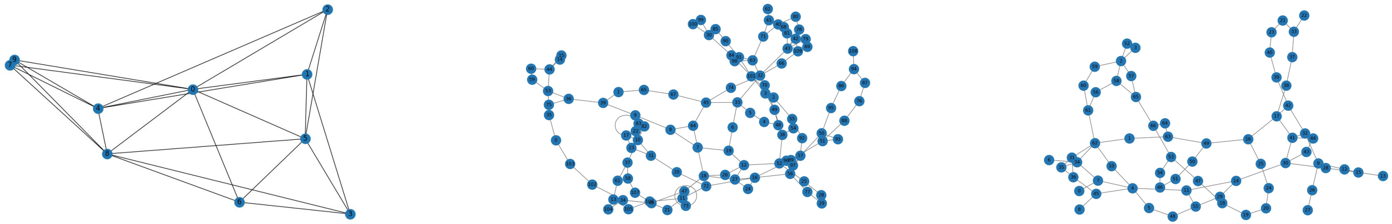


Fig. 2: (a) Random Network (b) Interoute Topology (c) Missouri Topology

- Artificial Intelligence (AI) can learn efficient autonomous communications networks by leveraging on data → Machine learning algorithms

Problem Formulation

- We consider a communication network with *N* nodes and *K* flows represented as a graph $\mathcal{G}=(\mathcal{V},\mathcal{E})$
- The performance function represents the aggregate amount of information generated at all nodes over all flows,

$$\mathbf{f}(a_i^k(t)) = \sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{V}} \log\left(a_i^k(t)\right)$$

- Considering the aggregate utility and necessary constraints, the optimization can be expressed as

$$\max_{\{a_i^k(t), r_{ij}^k(t)\}_{t=0}^{T-1}} \sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{V}} \log\left(\frac{1}{T}\sum_{t=0}^{T-1} a_i^k(t)\right)$$

$$s.t. \quad \sum_{j\in\mathcal{N}_i} r_{ji}^k(t) + a_i^k(t) \leq \sum_{j\in\mathcal{N}_i} r_{ij}^k(t), \ \forall k \in \mathcal{K}, \forall i \in \mathcal{V},$$ ----- Routing Constraint

$$\sum_{k\in K} r_{ij}^k(t) \leq C_{ij}(t), \ \forall (i,j) \in \mathcal{E},$$ ----- Capacity Constraint

$$a_i^k(t) \geq A_i^k(t), \ \forall k \in \mathcal{K}, \forall i \in \mathcal{V}$$ ----- Minimum Constraint

Routing in Communication Networks

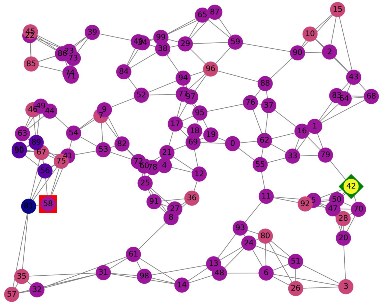


Fig. 3: Example communication network

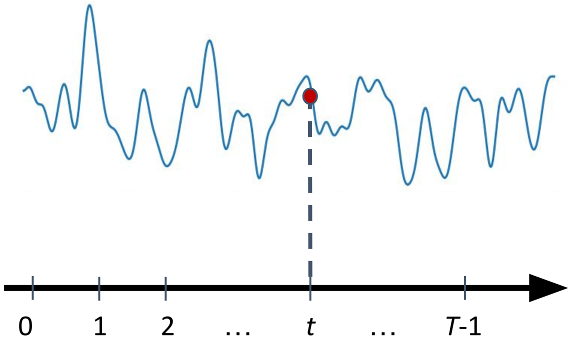
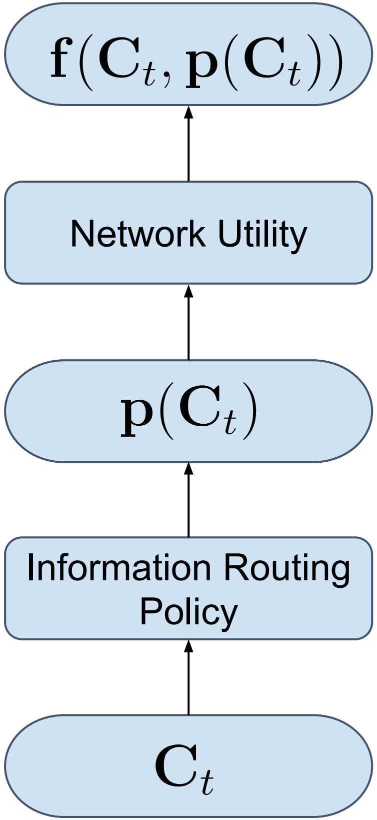


Fig. 4: Network state with respect to time

$$\max_{\{\mathbf{p}(\mathbf{C}_t)\}_{t=0}^{T-1}} \quad \mathcal{U}\left(\frac{1}{T}\sum_{t=0}^{T-1}\mathbf{f}(\mathbf{C}_t, \mathbf{p}(\mathbf{C}_t))\right)$$

$$s.t. \quad \mathbf{g}\left(\frac{1}{T}\sum_{t=0}^{T-1}\mathbf{f}(\mathbf{C}_t, \mathbf{p}(\mathbf{C}_t))\right) \geq 0$$



Routing Policy Parameterization

- At any time *t*, conventional formulation requires routing decisions to be recalculated for the given network state $\mathbf{C}_t$.
  ⇒ Learning and deploying such a policy becomes infeasible to deploy in real-time.

- Solution is to parameterize the resource allocation policy, replacing $\mathbf{p}(\mathbf{C}_t)$ with $\mathbf{p}(\mathbf{C}_t;\boldsymbol{\phi})$.

- Parameterization helps in generalization and scaling to large scale networks, learn and solve problems offline.

Unparameterized Formulation

$$\max_{\{\mathbf{p}(\mathbf{C}_t)\}_{t=0}^{T-1}} \quad \mathcal{U}\left(\frac{1}{T}\sum_{t=0}^{T-1}\mathbf{f}(\mathbf{C}_t,\mathbf{p}(\mathbf{C}_t))\right)$$

$$s.t. \quad \mathbf{g}\left(\frac{1}{T}\sum_{t=0}^{T-1}\mathbf{f}(\mathbf{C}_t,\mathbf{p}(\mathbf{C}_t))\right)\geq 0$$

→

Parameterized Formulation

$$P^* = \max_{\boldsymbol{\phi}} \quad \mathcal{U}\left(\frac{1}{T}\sum_{t=0}^{T-1}\mathbf{f}(\mathbf{C}_t,\mathbf{p}(\mathbf{C}_t;\boldsymbol{\phi}))\right)$$

$$s.t. \quad \mathbf{g}\left(\frac{1}{T}\sum_{t=0}^{T-1}\mathbf{f}(\mathbf{C}_t,\mathbf{p}(\mathbf{C}_t;\boldsymbol{\phi}))\right)\geq 0$$

- We utilize a Graph Neural Network (GNN) architecture to parameterize the network routing policy

Learning in the Dual Domain

A. Dual-Descent using Lagrangian Dual

- We consider a set of non-negative dual variables $\boldsymbol{\mu}$ corresponding to the constraint g.
- The Lagrangian can now be written as,

$$\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\mu}) = \mathcal{U}\left(\frac{1}{T}\sum_{t=0}^{T-1}\mathbf{f}(\mathbf{C}_t, \mathbf{p}(\mathbf{C}_t; \boldsymbol{\phi}))\right) + \boldsymbol{\mu}^T\mathbf{g}\left(\frac{1}{T}\sum_{t=0}^{T-1}\mathbf{f}(\mathbf{C}_t, \mathbf{p}(\mathbf{C}_t; \boldsymbol{\phi}))\right)$$

- Now the objective is to maximize the Lagrangian over ϕ while minimizing over $\boldsymbol{\mu}$,

$$\mathcal{L}^* = \min_{\boldsymbol{\mu}\geq 0}\max_{\boldsymbol{\phi}\in\boldsymbol{\Phi}} \quad \mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\mu})$$

Cons: Primal-Dual suffers from slow convergence

B. Method of Multipliers (MoM) using Augmented Lagrangian Dual

- Method of Multipliers handles more general framework suitable for real-time scenarios.
- We introduce an auxiliary variable z to convert inequality constraints to equality constraints.
- The augmented Lagrangian can now be written as,

$$\mathcal{L}_a(\boldsymbol{\phi}, \boldsymbol{\mu}, \boldsymbol{z}) = \mathcal{U}\left(\frac{1}{T}\sum_{t=0}^{T-1}\mathbf{f}(\mathbf{C}_t, \mathbf{p}(\mathbf{C}_t;\boldsymbol{\phi}))\right) + \boldsymbol{\mu}^T\left\{\mathbf{g}\left(\frac{1}{T}\sum_{t=0}^{T-1}\mathbf{f}(\mathbf{C}_t, \mathbf{p}(\mathbf{C}_t;\boldsymbol{\phi}))\right) - \boldsymbol{z}\right\} + \frac{\rho}{2}\left\|\boldsymbol{\mu}^T\left\{\mathbf{g}\left(\frac{1}{T}\sum_{t=0}^{T-1}\mathbf{f}(\mathbf{C}_t, \mathbf{p}(\mathbf{C}_t;\boldsymbol{\phi}))\right) - \boldsymbol{z}\right\}\right\|^2$$

- 3 step optimization including the penalty method

$$\mathcal{L}_a(\boldsymbol{\phi}, \boldsymbol{\mu}) = \max_{\boldsymbol{z}} \mathcal{L}_a(\boldsymbol{\phi}, \boldsymbol{\mu}, \boldsymbol{z}) \longrightarrow \mathcal{L}_a^*(\boldsymbol{\phi}, \boldsymbol{\mu}) = \max_{\boldsymbol{\phi}} \mathcal{L}_a(\boldsymbol{\phi}, \boldsymbol{\mu}) \longrightarrow \mathcal{L}_a^*(\boldsymbol{\phi}, \boldsymbol{\mu}) = \min_{\boldsymbol{\mu}} \mathcal{L}_a(\boldsymbol{\phi}, \boldsymbol{\mu})$$

Iterative MoM Update

- The primal model parameters ϕ and the dual variable $\boldsymbol{\mu}$ are updated iteratively
- Let m be the iteration index and T_0 be iteration duration between consecutive updates

$$\boldsymbol{\phi}_m = \arg\max_{\boldsymbol{z},\boldsymbol{\phi}\in\boldsymbol{\Phi}} \left[\mathcal{U}\left(\frac{1}{T_0}\sum_{t=mT_0}^{(m+1)T-1} \mathbf{f}(\mathbf{C}_t, \mathbf{p}(\mathbf{C}_t;\boldsymbol{\phi}))\right) + \boldsymbol{\mu}^T\left\{\mathbf{g}\left(\frac{1}{T_0}\sum_{t=mT_0}^{(m+1)T_0-1} \mathbf{f}(\mathbf{C}_t, \mathbf{p}(\mathbf{C}_t;\boldsymbol{\phi}))\right) - \boldsymbol{z}\right\} + \frac{\rho}{2}\left\|\boldsymbol{\mu}^T\left\{\mathbf{g}\left(\frac{1}{T_0}\sum_{t=mT_0}^{(m+1)T_0-1} \mathbf{f}(\mathbf{C}_t, \mathbf{p}(\mathbf{C}_t;\boldsymbol{\phi}))\right) - \boldsymbol{z}\right\}\right\|^2\right]$$

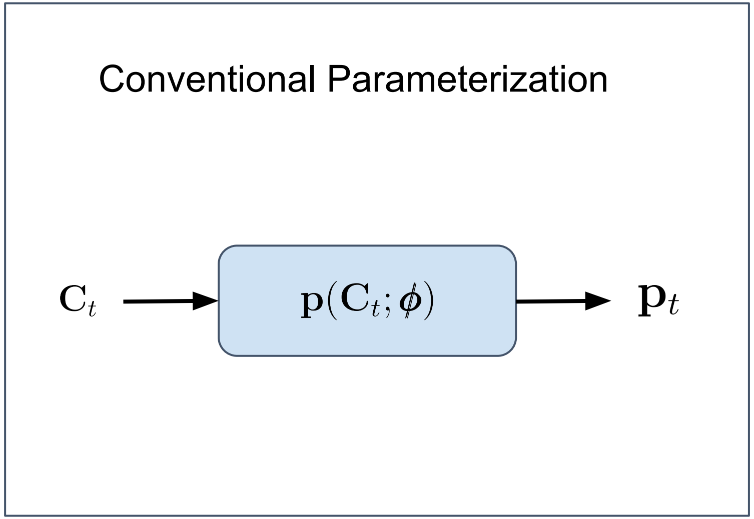
$m \rightarrow m+1$

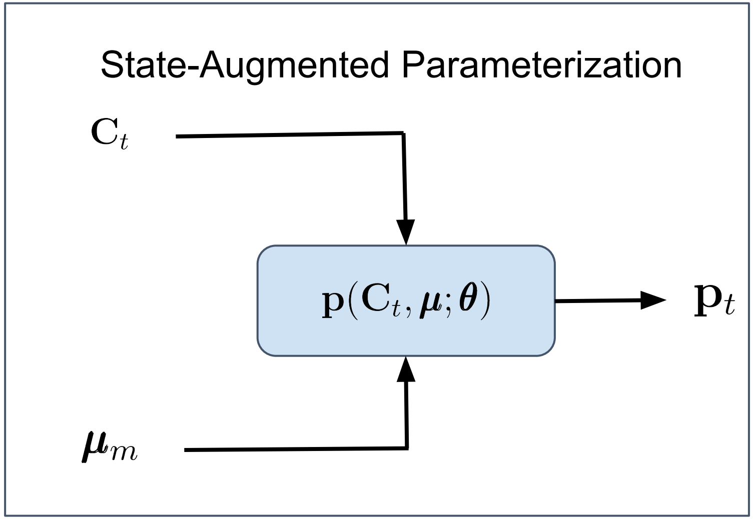
$$\boldsymbol{\mu}^{m+1} = \left[\boldsymbol{\mu}^m - \rho^m \mathbf{g}\left(\frac{1}{T_0}\sum_{t=mT_0}^{(m+1)T_0-1} \mathbf{f}(\mathbf{C}_t, \mathbf{p}(\mathbf{C}_t;\boldsymbol{\phi}_m))\right)\right]^+$$

- Constraint slacks are the gradient of the augmented Lagrangian with respect to the dual variables.

The State-Augmented Algorithm

- We use both the network state $\mathbf{C}_t$ and dual variables $\boldsymbol{\mu}$ as input to the network routing policy
- We use a separate parameterization $\boldsymbol{\theta}$ for the state-augmented policy as $\mathbf{p}(\mathbf{C}_t, \boldsymbol{\mu}; \boldsymbol{\theta})$





State-Augmented Primal and Dual Update

- The revised augmented Lagrangian with the new parameterization can now be expressed as

$$\mathcal{L}_a(\boldsymbol{\theta}) = \mathcal{U}\left(\frac{1}{T_0}\sum_{t=kT_0}^{T-1}\mathbf{f}(\mathbf{C}_t, \mathbf{p}(\mathbf{C}_t, \boldsymbol{\mu}; \boldsymbol{\theta}))\right) + \boldsymbol{\mu}^T\left\{\mathbf{g}\left(\frac{1}{T_0}\sum_{t=kT_0}^{T_0-1}\mathbf{f}(\mathbf{C}_t, \mathbf{p}(\mathbf{C}_t, \boldsymbol{\mu}; \boldsymbol{\theta}))\right) - \boldsymbol{z}\right\} + \frac{\rho}{2}\left\|\boldsymbol{\mu}^T\left\{\mathbf{g}\left(\frac{1}{T_0}\sum_{t=kT_0}^{T_0-1}\mathbf{f}(\mathbf{C}_t, \mathbf{p}(\mathbf{C}_t, \boldsymbol{\mu}; \boldsymbol{\theta}))\right) - \boldsymbol{z}\right\}\right\|^2$$

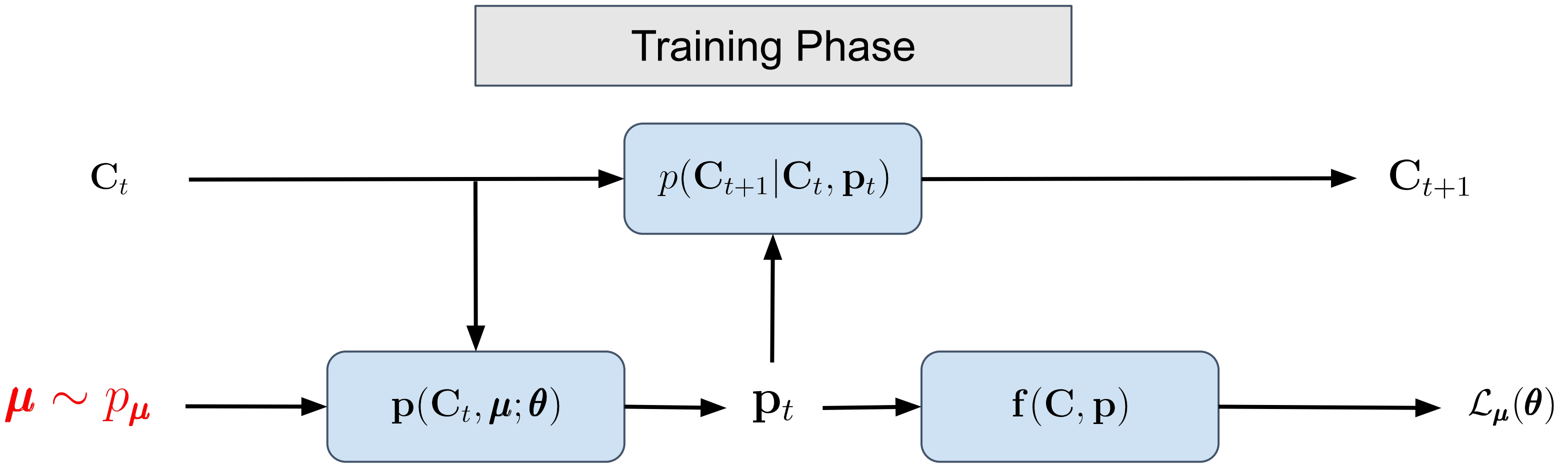
- The optimal state-augmented policy parameters are obtained during the training and saved as

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \mathbb{E}_{\boldsymbol{\mu}\sim p_{\boldsymbol{\mu}}}\left[\mathcal{L}_{\boldsymbol{\mu}}(\boldsymbol{\theta})\right]$$

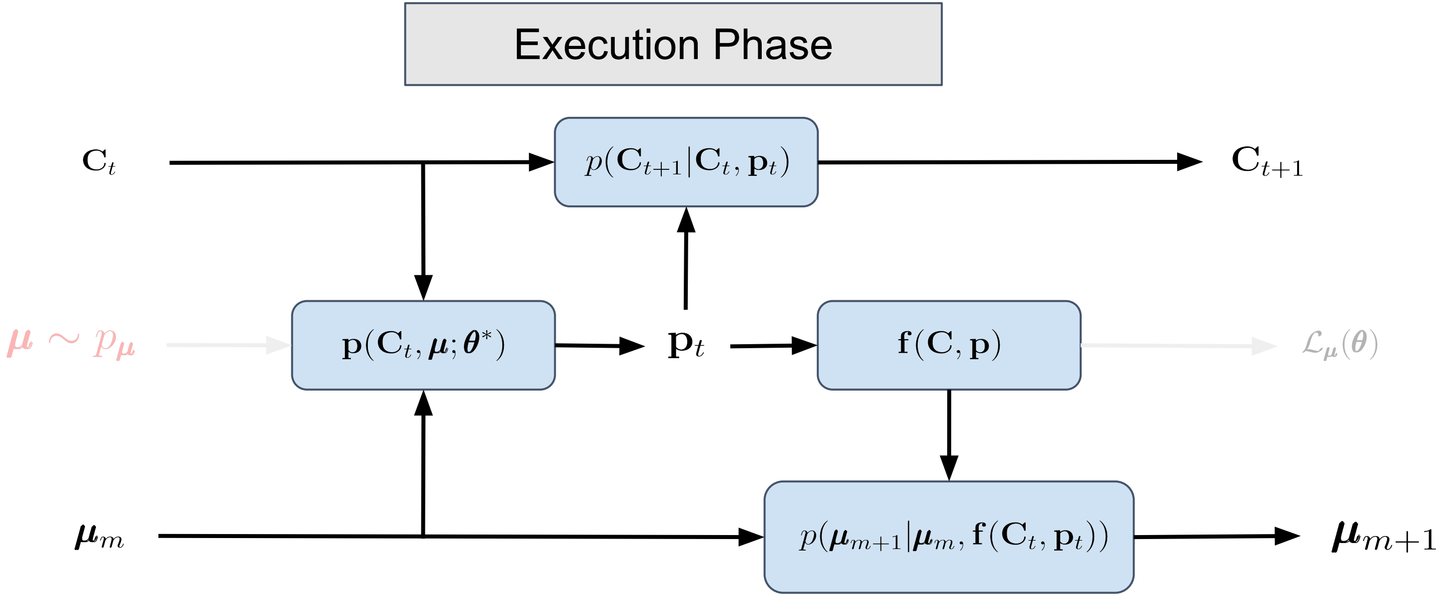
- This waives off the problem of re-optimizing the model parameters for every set of dual variables
- During execution, the dual variables can be updated as

$$\boldsymbol{\mu}^{m+1} = \left[\boldsymbol{\mu}^m - \rho^m \mathbf{g}\left(\frac{1}{T_0}\sum_{t=mT_0}^{(m+1)T_0-1}\mathbf{f}(\mathbf{C}_t, \mathbf{p}(\mathbf{C}_t; \boldsymbol{\theta}^*))\right)\right]^+$$

State-Augmented Training vs. Execution

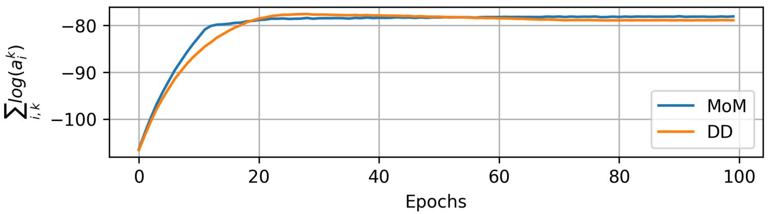


State-Augmented Training vs. Execution

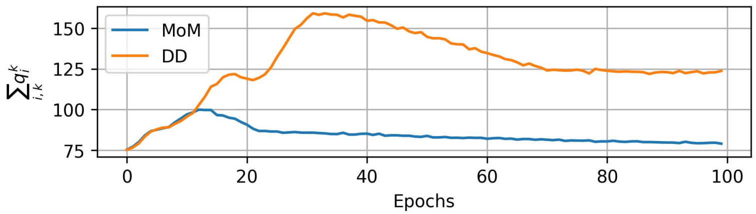


- The State-Augmentation algorithm can guarantee feasibility and near optimality - Navid NaderiAlizadeh *et al*, “State-augmented learnable algorithms for resource management in wireless networks” [IEEE TSP 2022]

Performance Comparison

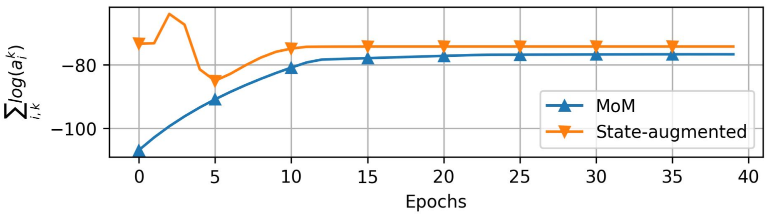


(a) Performance on utility

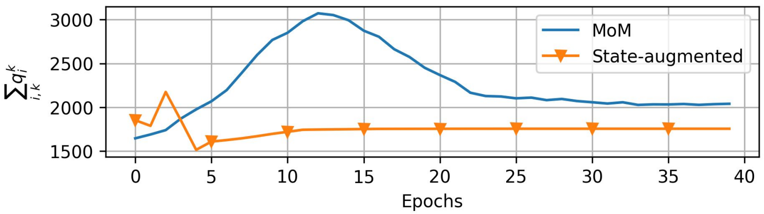


(b) Performance on queue length

Fig. 5: Comparison of two different unparameterized methods of MoM and Dual Descent (DD) for a network with 10 nodes and 5 flows, where the network is run only for a single time step.



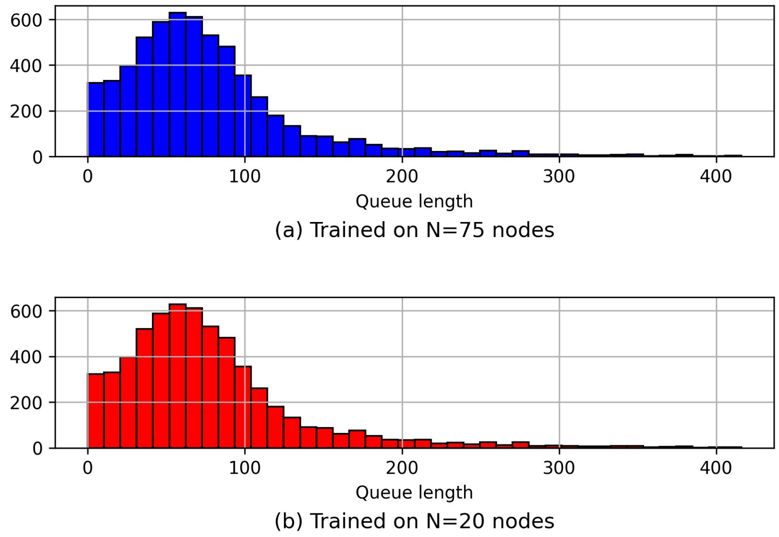
(a) Performance on utility



(b) Performance on queue length

Fig. 6: Comparison of the state-augmented algorithm with MoM for a network with N= 20 nodes and F = 5 flows for T=100.

Scalability & Transferability of GNN based Routing Policies



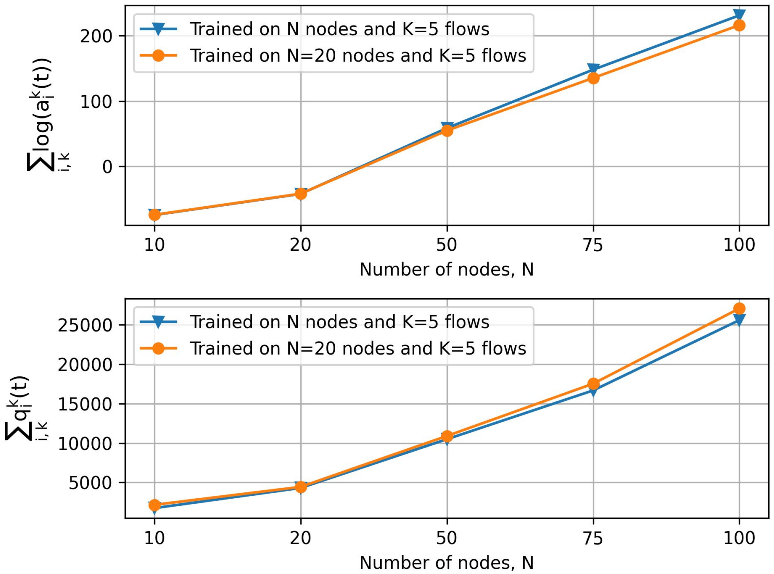


Fig. 7: Transferability of a GNN trained on networks with 20 nodes and tested on networks with 75 nodes (bottom) vs. a GNN that has been both trained and tested on 75-node networks (top).

Fig. 8: Transferability of the proposed state-augmented algorithm on networks with different nodes while they were trained on a network with 20 nodes and 5 flows.

Policy Switching Behavior due to State Augmentation

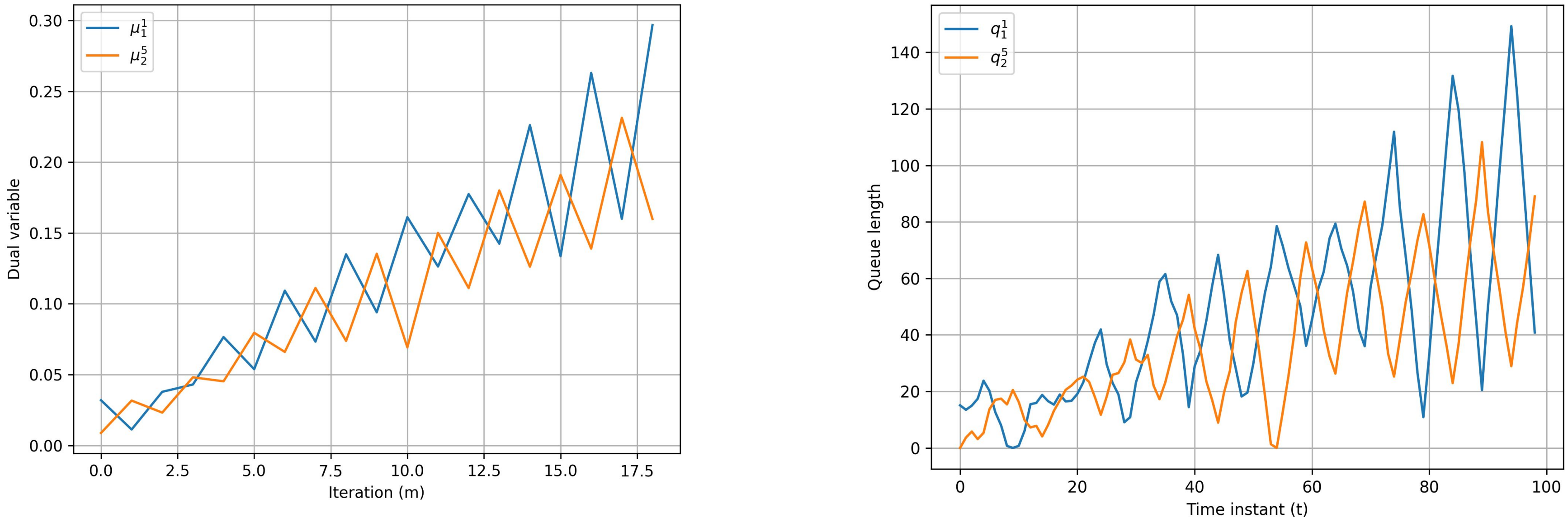


Fig. 9: Behavior of a dual variable and queue length stability for an example node in a network with 50 nodes and 5 flows.

THANK YOU !!!

Sourajit Das
University of Pennsylvania

Navid NaderiAlizadeh
Duke University

Alejandro Ribeiro
University of Pennsylvania

GitHub