Beyond Simple Graphs: Neural Multi-Objective Routing on Multigraphs

Filip Rydin[1] Attila Lischka[1] Jiaming Wu[1] Morteza Haghir Chehreghani[1,2] Balázs Kulcsár[1]

Introduction

Current learning-based methods are unable to handle multigraph representations, which arise when multiple competing edges exist between node pairs (e.g., differing in travel time and distance).

Why multigraphs matter: Prior work has shown multigraph routing yields cost savings of up to **10.5%** over simple graph approaches on real instances (Ben Ticha et al., 2017). Yet no learning-based solver existed for this setting.

Main contributions:

- **First** neural solvers specifically designed for multigraph VRPs.
- Two GNN-based models covering the speed–quality trade-off.

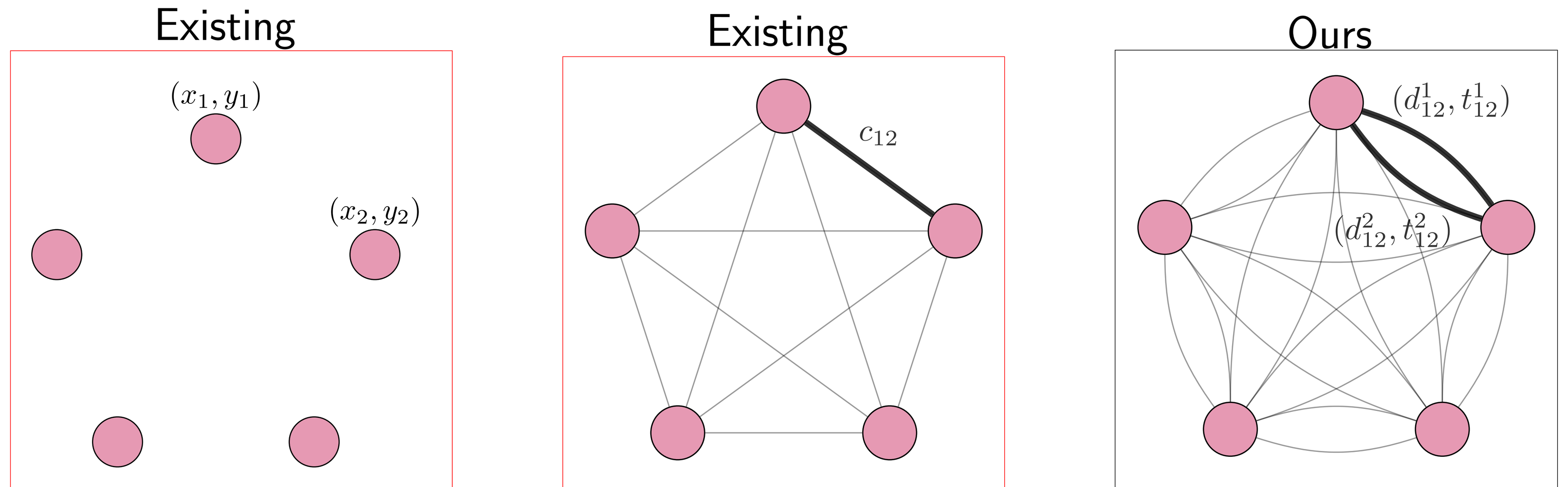

Figure 1a. Euclidean. Figure 1b. Simple graph. Figure 1c. Multigraph.

Problem Formulation

Multi-objective routing on multigraphs: find **Pareto set** of routes π (edge sequences):

$$\min_{\pi \in \Pi} f(\pi), \quad f : \Pi \to \mathbb{R}^m, \; m > 1.$$

Solutions obtained via **decomposition through scalarization**.

Chebyshev scalarization:

$$f_\lambda(\pi) = \max_i \{\lambda_i |f_i(\pi) - z_i^*|\},$$

Unlike linear scalarization, Chebyshev can recover any Pareto-optimal solution.

Key insight (Proposition 1):

- Under linear scalarization and multigraph MOTSP, selecting edge with lowest scalarized cost is optimal.
- Under Chebyshev scalarization, and for more intricate problems, optimal edge selection is not obvious.

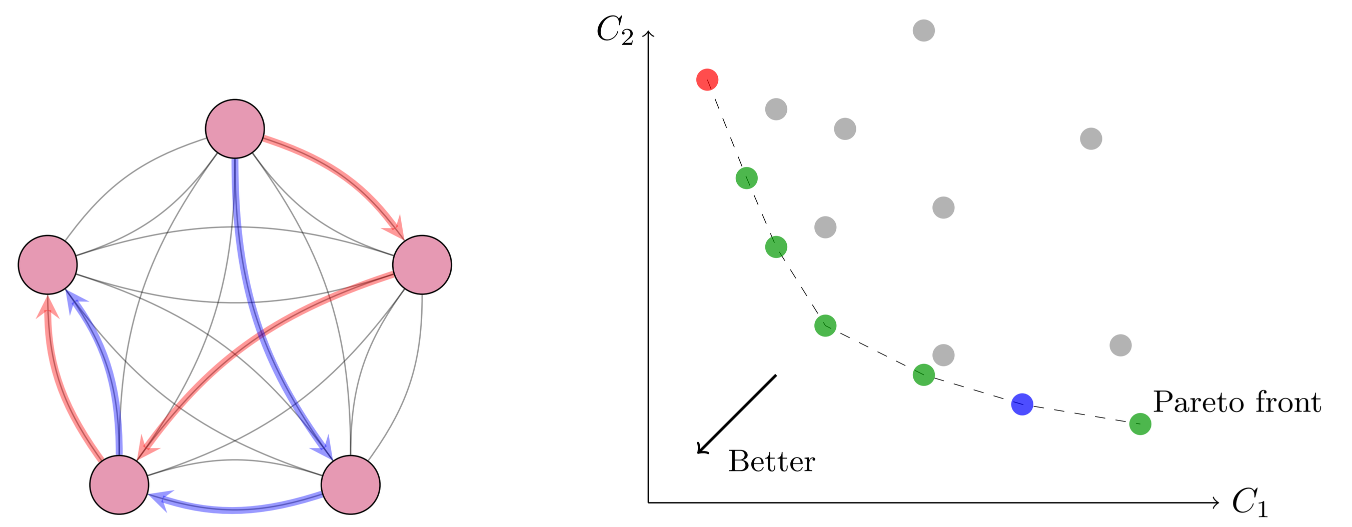

Figure 2. Different solutions in the Pareto set correspond to different node permutations and edge selections.

Proposed Method 1: Edge-Based GMS (GMS-EB)

Key idea: Autoregressively select *edges*, so both the next node *and* which parallel edge to use are decided jointly.

Components:

- **Encoder:** Graph Edge Attention Network (GREAT) layers producing edge embeddings.
- **Decoder:** Edge-based Multi-Pointer (MP) decoder: scores each outgoing edge.
- **Hypernetwork:** MLP generates preference-conditioned decoder weights $\theta_{\text{dec}}(\lambda) = \text{MLP}(\lambda)$, encoder is preference-agnostic.

Training with MO REINFORCE:

$$\nabla J(\theta) \approx \frac{1}{BK} \sum_{i,j} \left(R_\lambda(\pi_{ij}) - b_\lambda(s_i)\right) \nabla_\theta \log p_{\theta(\lambda)}(\pi_{ij} | s_i)$$

Complexity: Representation size increases from $\mathcal{O}(N)$ to $\mathcal{O}(MN^2)$, yielding increased space and time complexity.

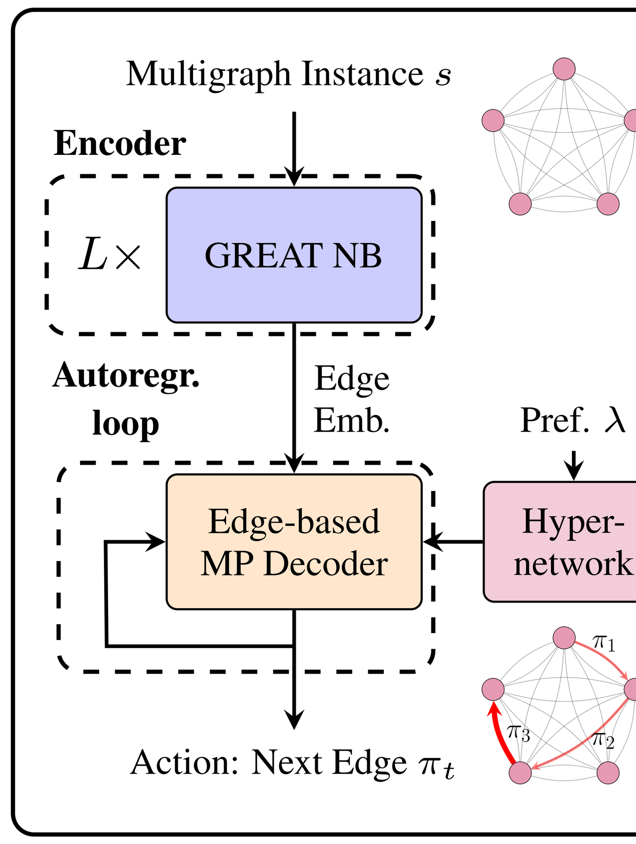

Figure 3: GMS-EB architecture.

Proposed Method 2: Dual-Head GMS (GMS-DH)

Key idea: Non-autoregressive (NAR) **pruning**, autoregressive (AR) **node-based routing**.

Components:

- **Shared encoder:** preference-agnostic GREAT NB layers (run *once* per instance).
- **Selection head (NAR):** multi-pointer scoring, pick one edge per node pair: selection $\mathcal{E}$.
- **Final GREAT layer + L_2 Transformer layers:** produce expressive node embeddings conditioned on $\mathcal{E}$.
- **Routing head (AR):** preference-conditioned node-based MP decoder.
- **Two hypernetworks:** one per decoder head.

Joint training: Selection head maximizes

$$J_1(\tilde{\theta}) = \mathbb{E}_{\lambda \sim \Lambda, s \sim \mathcal{S}, \mathcal{E} \sim q_{\tilde{\theta}}}[R_\lambda(\tilde{\pi}(\mathcal{E}))].$$

Approximate with head 2:

$$R_\lambda(\tilde{\pi}(\mathcal{E})) \approx \max_{k=1,\ldots K} R_\lambda(\pi_k).$$

Routing head maximizes

$$J_2(\theta) = \mathbb{E}_{\pi \sim p_\theta, \lambda \sim \Lambda, s \sim \mathcal{S}, \mathcal{E} \sim q_{\tilde{\theta}}}[R_\lambda(\pi)].$$

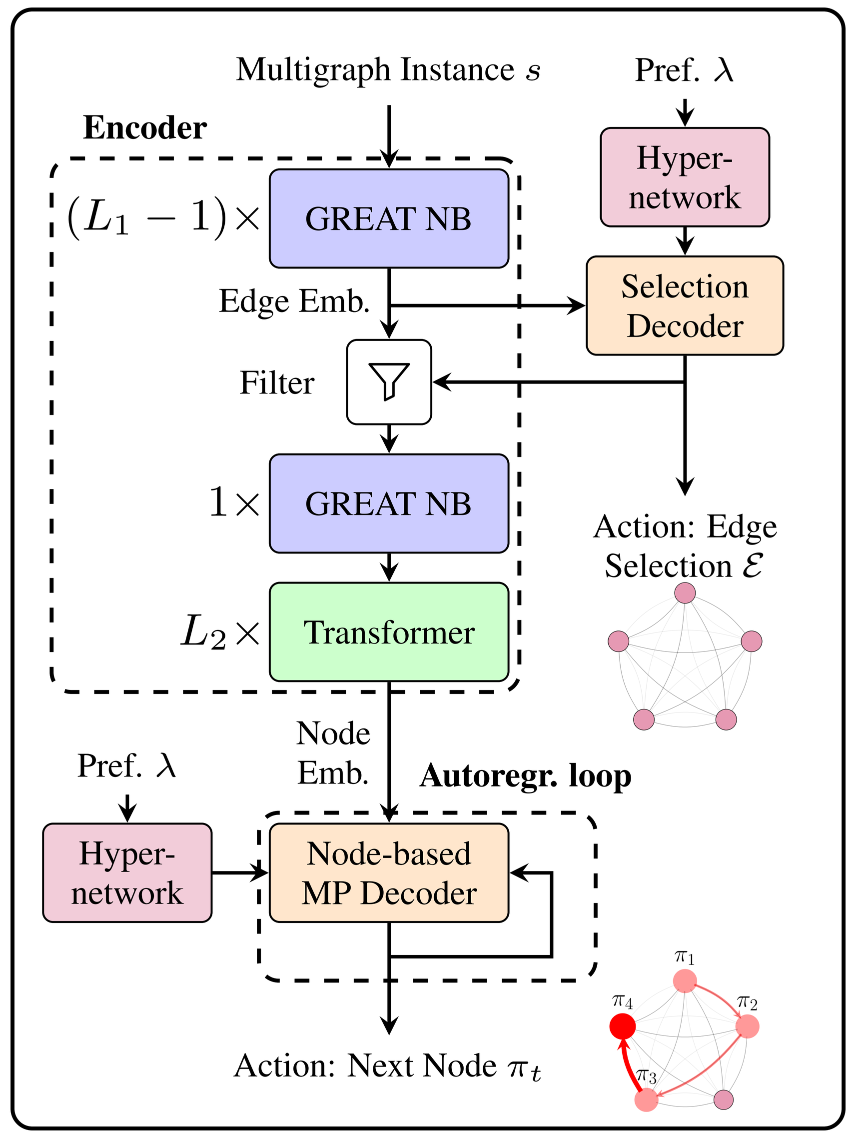

Figure 4: GMS-DH architecture.

Complexity: Representation size $\mathcal{O}(N)$ maintained in AR decoding, yielding substantially faster inference and training with small performance drop.

Results

Table: Multigraph bi-objective TSP and CVRP with 100 nodes.

	MGMOTSP						MGMOCVRP					
	FLEX2-100			FIX2-100			FLEX2-100			FIX2-100		
	HV	Gap	Time	HV	Gap	Time	HV	Gap	Time	HV	Gap	Time
LKH / HGS	0.94	0.00%	(31m)	0.95	0.00%	(30m)	0.89	0.00%	(75h)	0.91	0.00%	(74h)
OR Tools	0.91	3.46%	(1.8h)	0.93	2.25%	(2.0h)	0.84	5.97%	(1.9h)	0.88	3.78%	(1.9h)
MOEA/D / NSGA-II	0.68	28.46%	(99h)	0.72	24.69%	(91h)	0.56	36.54%	(219h)	0.61	33.28%	(224h)
MBM	0.91	3.72%	(44s)	0.93	2.41%	(44s)	0.81	9.38%	(54s)	0.87	5.00%	(54s)
MBM (aug)	0.91	3.28%	(5.9m)	0.93	2.10%	(5.9m)	0.82	8.30%	(6.9m)	0.87	4.32%	(6.9m)
GMS-EB	0.93	1.81%	(9.1m)	0.94	1.33%	(9.2m)	0.85	4.08%	(10m)	0.89	2.81%	(9.9m)
GMS-EB (aug)	**0.93**	**1.47%**	(1.2h)	**0.94**	**1.09%**	(1.2h)	**0.86**	**3.66%**	(1.4h)	**0.89**	**2.38%**	(1.4h)
GMS-DH	0.92	2.11%	(49s)	0.93	2.13%	(52s)	0.85	4.94%	(1.0m)	0.88	3.58%	(1.0m)
GMS-DH (aug)	0.93	1.61%	(6.6m)	0.93	1.92%	(6.6m)	0.85	3.89%	(8.2m)	0.88	3.05%	(8.1m)

Metric: bodyized Hypervolume (HV), averaged over 200 test instances. Runtime is total time for all 200 instances.

MBM: Neural baseline based on the MatNet architecture. Utilizes linear-scalarization pre-pruning before the encoder.

More Intricate Problem: MGMOTSPTW

Table: Multigraph bi-objective TSP with time-windows. Objectives = (1) #violated time-windows, (2) total distance. Edge pruning is non-trivial.

	FLEX2-50			FIX2-50		
	HV	Gap	Time	HV	Gap	Time
MOEA/D	0.60	32.79%	(34h)	0.60	35.73%	(36h)
MBM	0.80	11.09%	(14s)	0.64	31.18%	(14s)
MBM (aug)	0.81	10.10%	(1.8m)	0.66	28.58%	(1.9m)
GMS-EB	0.89	0.97%	(60s)	0.93	0.55%	(59s)
GMS-EB (aug)	**0.90**	**0.00%**	(7.9m)	**0.93**	**0.00%**	(7.9m)
GMS-DH	0.85	5.10%	(13s)	0.91	1.74%	(12s)
GMS-DH (aug)	0.87	2.69%	(1.6m)	0.92	0.78%	(1.7m)
GMS-DH Simple	0.83	7.01%	(13s)	0.85	8.93%	(12s)

GMS-DH Simple: Ablation replacing the learned selection-decoder with a simple linear-scalarization pruning heuristic.

Conclusion

- GMS-EB and GMS-DH are the **first learning-based methods** for VRPs on multigraphs, achieving competitive performance across several problems.
- **GMS-EB** offers slightly superior solution quality; **GMS-DH** is significantly faster at inference.

Future work: Alternative decomposition methods beyond pruning for improved scalability; single-objective hard-constraint problems; stochastic edge-cost variants inspired by electric vehicle routing.

Links

Code Paper LinkedIn

Contact: filipry@chalmers.se

[1]Chalmers University of Technology [2]University of Gothenburg

ICLR 2026, Rio de Janeiro, Brazil