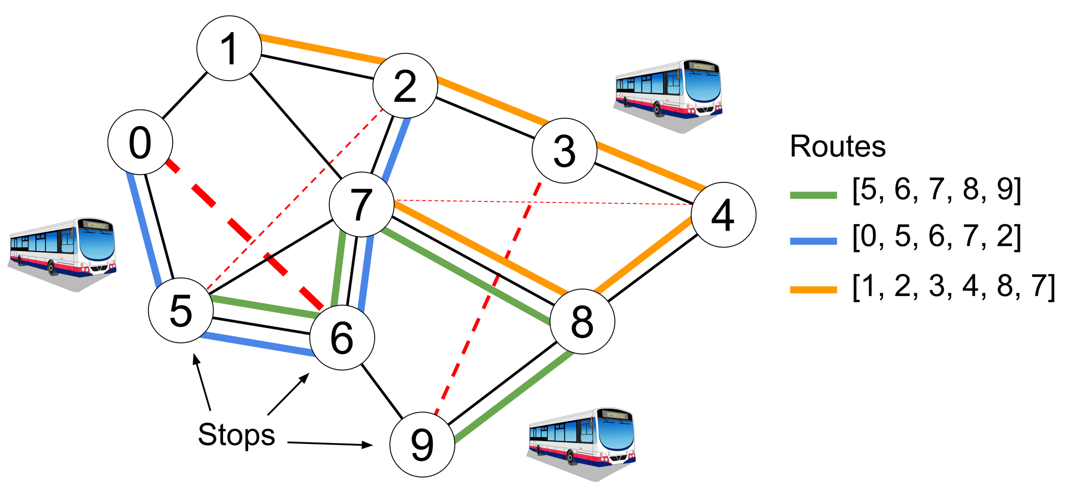
A **Graph Neural Net** can choose mutations in an **evolutionary algorithm** to design **better routes** for **self-driving buses.**

A Neural-Evolutionary Algorithm for Autonomous Transit Network Design

Andrew Holliday, Gregory Dudek

Transit Network Design



We are given a city graph $\mathcal{C}$ composed of:

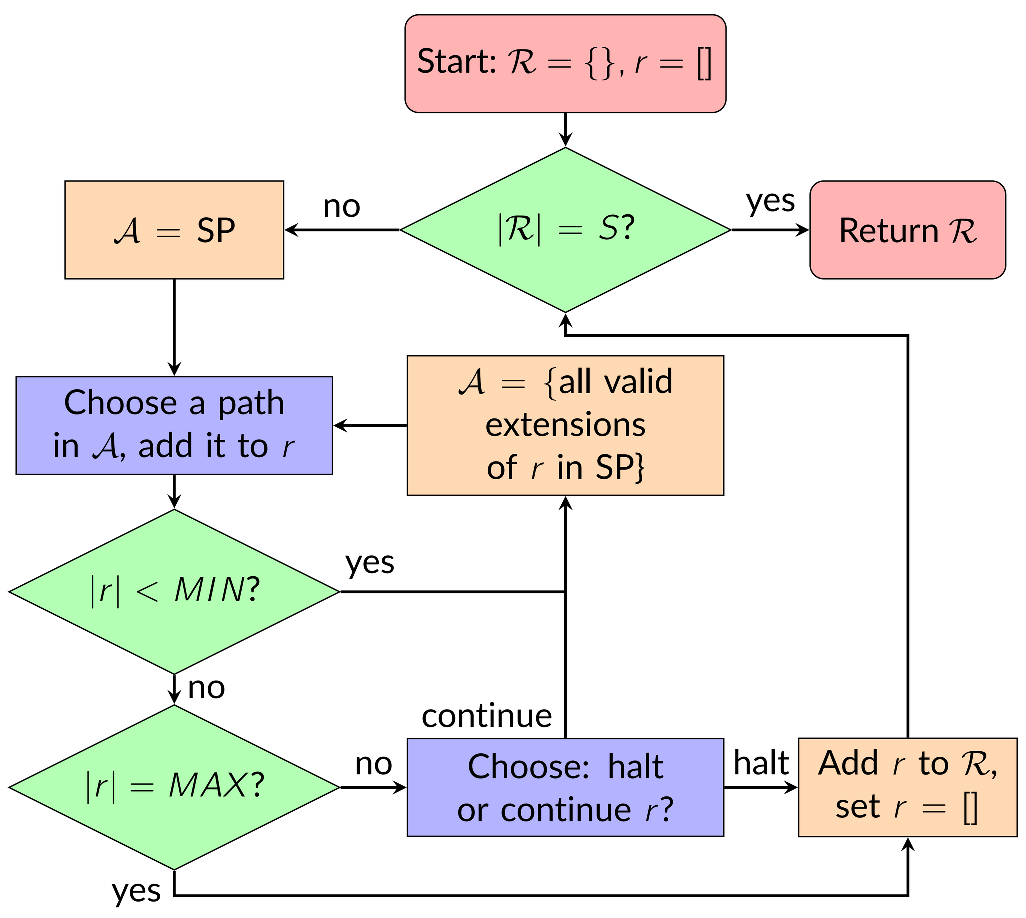
- Transit stop nodes $\mathcal{N}$,
- A set of road edges $\mathcal{E}_s$,
- An $|\mathcal{N}| \times |\mathcal{N}|$ demand matrix D

A **route** as a **non-repeating path** in $\mathcal{C}$. Our goal is to find a **set of routes**, $\mathcal{R}$, that connects all nodes while minimizing a cost function:

$$C(\mathcal{C}, \mathcal{R}) = \alpha C_p(\mathcal{C}, \mathcal{R}) + (1 - \alpha) C_o(\mathcal{C}, \mathcal{R}) \quad (1)$$

- C_p: mean trip time over all passengers
- C_o: total length of all routes
- $\alpha \in [0, 1]$

We formulate transit network **construction** as a Markov Decision Process (MDP). The below flowchart gives the MDP structure. Blue nodes are agent choices.

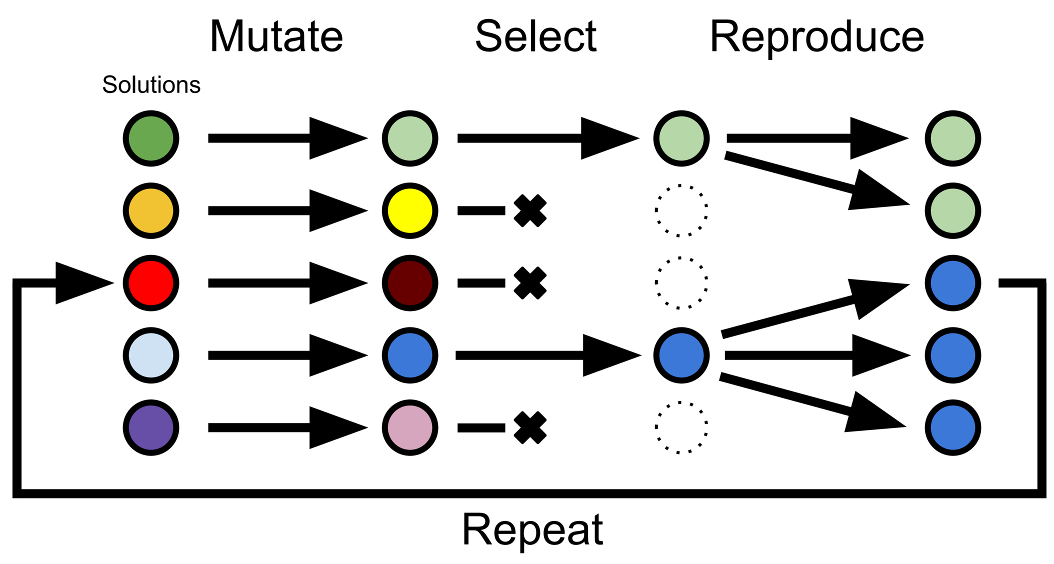


Neural Policy

We train a policy for this MDP via REINFORCE with baseline [3] on a dataset of synthetic cities, with reward $R = -C(\mathcal{R})$. The policy π_θ is a graph attention net with two "head" MLPs for alternating actions.

Evolutionary Algorithm

The evolutionary algorithm of [2] (**EA**) is a transit network **improvement** algorithm. It takes an initial network $\mathcal{R}$ and iteratively applies random "mutations", then stochastically filters the mutated networks $\mathcal{R}_b$ based on $C(\mathcal{C}, \mathcal{R}_b)$.



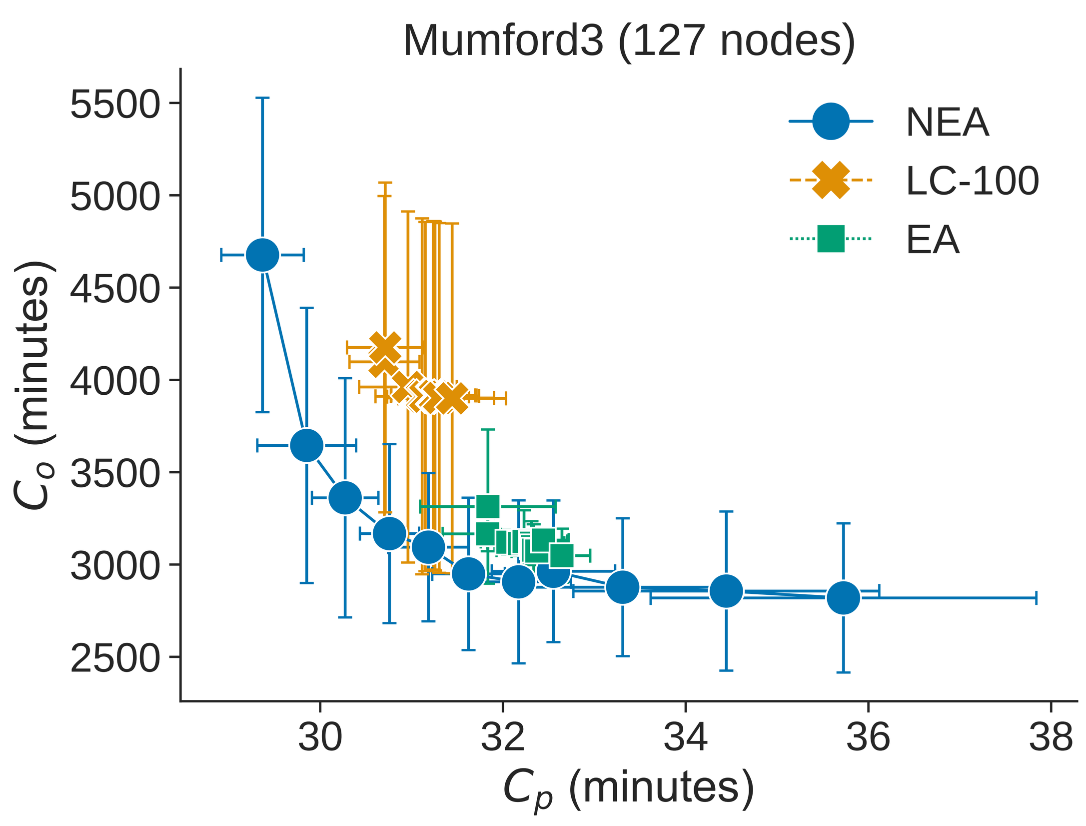
Schematic of an evolutionary algorithm

This algorithm has two "mutator" operators. One deletes a random route r from $\mathcal{R}$ and replaces it with the shortest path between one terminal of r and a random other node. The other chooses a random route and either adds or deletes a random node at one end.

We change the first mutator: instead of a random shortest path, we replace r with a new route r' sampled from $\pi_\theta(\mathcal{R} \setminus \{r\})$. We call this modified algorithm the **neural evolutionary algorithm (NEA)**.

Experiments

We compare NEA to EA and to π_θ alone. For π_θ alone, we sample 100 transit networks and take the lowest-cost network. We call this procedure **LC-100**. We evaluate on the widely-used Mumford benchmark cities [1]. For benchmark cities with $\geq$ 70 nodes, NEA consistently dominates EA and LC-100 across values of α.



This figure shows our results for α values from 0.0 to 1.0 on the largest Mumford city, Mumford3, which has 127 nodes.

References

[1] Christine L Mumford. "New heuristic and evolutionary operators for the multi-objective urban transit routing problem". In: *2013 IEEE congress on evolutionary computation*. IEEE. 2013, pp. 939–946.

[2] Miloš Nikolić and Dušan Teodorović. "Transit network design by bee colony optimization". In: *Expert Systems with Applications* 40.15 (2013), pp. 5945–5955.

[3] Ronald J Williams. "Simple statistical gradient-following algorithms for connectionist reinforcement learning". In: *Machine learning* 8.3 (1992), pp. 229–256.

McGill

ICRA2024 YOKOHAMA | JAPAN

Scan QR code to get the full paper