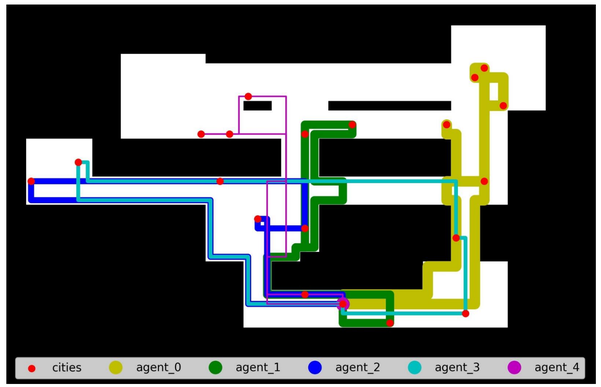
Multiple Travelling Salesmen Problem (mTSP) on Realistic Maps via Decentralised Attention-Based Reinforcement Learning

Travelling Salesmen: Derek Tan, Ma Yixiao, Zhou Lyu, Low Jia Liang

National University of Singapore

INTRODUCTION



MinMax multiple travelling salesmen problem (mTSP) is a NP-hard problem where multiple agents collaborate to visit all the cities, while minimizing the maximum tour length taken by any agent.

Applications: Robotics search and rescue, urban planning, logistics and transportation, telecommunications planning

Limitation: Most mTSP formulations assume an empty world and do not model environmental features that constraints path planning, such as roads or obstacles. Our work focuses on non-euclidean mTSP on an occupancy grid map commonly used in robotic navigation, which extends well to constrained graphs such as road networks.

Attention-based Deep Reinforcement Learning (DRL):
RL allows agents to make non-myopic decisions by maximizing long-term cumulative rewards. Self-attention allows agents to accurately predict implications of current decisions, by modelling inter-dependencies between the agents and cities.

Problem formulation

The multiple Traveling Salesmen Problem (mTSP) is formulated on a **probabilistic roadmap graph *G = (V, E)* generated in the free-space of an occupancy grid map**, where V = {1, ..., n} represents a collection of n nodes and E is the set of edges that connect these nodes. We use the A* distances between cities as the distance to be travelled by the agents as they move between cities.

$$L(\pi^i) = \sum_{j=1}^{n_i-1} c_{\pi^i_j \pi^i_{j+1}} \qquad \text{minimize} \max_{i \in \{1,\dots,m\}} L(\pi^i)$$

MTSP – As an RL problem

We formulate the mTSP as a sequential decision-making problem where agents cooperatively tour all cities while minimising cost.

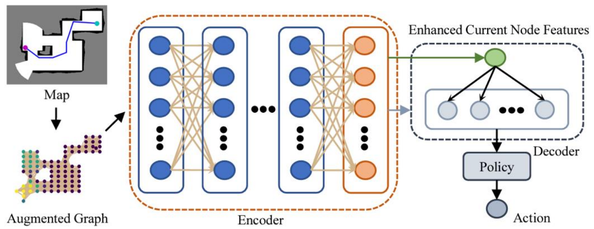
Sequential Decision Making: Agents makes decisions asynchronously based on individual observations as they arrive at their intended city.

State: Map of the environment, position of the cities and agents, and visitation history of which cities have been visited.
(Assume a fully observable world).

Action: Selection of an unvisited city or the depot. Agents can take an action when they have arrived at their target city.

Reward: Negative max tour length of all the agents.

MODEL



Each agent's model is determined by an attention-based neural network that consists of encoder and a decoder.

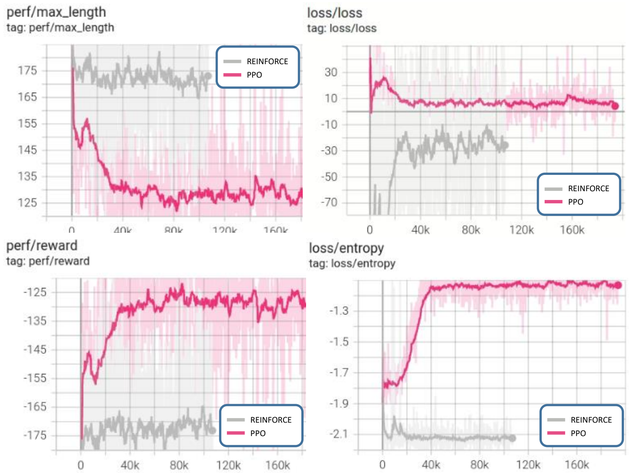
Attention Layers: The inputs to the attention layers consist of a query vector and a key value vector. The output, is determined by taking the weighted sum of the value vector, where the weights are determined based on the similarity between the query and key vectors.

$$q_i = W^Q h_i^q,\ k_i = W^K h_i^{k,v},\ v_i = W^V h_i^{k,v},$$
$$u_{ij} = \frac{q_i^T \cdot k_j}{\sqrt{d}},\ w_{ij} = \frac{e^{u_{ij}}}{\sum_{j=1}^{n} e^{u_{ij}}},\ h_i' = \sum_{j=1}^{n} w_{ij} v_j,$$

Encoder: Stacks 3x attention layers to extract important features representing long-term relationships between nodes.

Decoder: Inputs encoder features, with the agent's current position and unvisited cities, into a policy to choose the next city to visit.

Training



We benchmark two policy-based RL training algorithms to assess their performance and convergence properties given the mTSP setup.

REINFORCE: Basic policy-based optimization algorithm to select actions that maximizes expected long-term returns. It considers the advantage (Reward – Baseline) and log-prob of action taken.

$$L = -\mathbf{E}_{p_\theta(\pi^i)}[(R(\pi) - b(\pi))\nabla \log p_\theta(\pi^i)]$$

Proximal Policy Optimization (PPO): Addresses training instabilities due to high variance induced by the REINFORCE algorithm. Introduces a trust region that clips the advantage (Reward – Baseline) to constraint the maximum update step size. This prevents the new policy from diverging too much from the old policy.

$$L^{CPI}(\theta) = \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t\right] = \hat{\mathbb{E}}_t\left[r_t(\theta)\hat{A}_t\right]$$

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right]$$

Shared Experience Replay Buffer: Results from asynchronous training simulations are collected and stored in a common replay buffer.

Parameter Sharing: Global policy weights are updated every fixed training interval, and propagated back to all training simulations

Hardware : 4x A5000 GPU, Ryzen Threadripper 3970x CPU (32 cores)

Training: 5 agents, 20 cities (50k episodes, 10h, 5600 training maps)

Experiments

(a) simple (b) medium (c) complex

Test Set: 100x simple, 100x medium, 100x complex maps
Setup: 3 random seeds (900 episodes in total), map scale 1:10 (64x48)
Rule: All agents start and end at depot, speed = 1pixel/s
Test Results: Map-constraint DAN works better than vanilla DAN

Environment	Vanilla DAN	Our Work	Improvement
Simple	112 (±90)	94 (±61)	16.1%
Medium	153 (±108)	125 (±69)	18.3%
Complex	177 (±120)	141 (±83)	20.3%

Conclusions

Attention-based DRL has demonstrated excellent results via long-term planning for mTSP problems on obstacle maps. Future steps include:

Varying Number of Agents & Cities: Investigate how number of agents and cities impact performance and scalability.

Model-Based RL: Current model-free RL is highly sample inefficient.

Alternate Map Representation: Directly input occupancy grid maps as an 2D array and encode using CNNs.

Partially Observable Environment: Limit knowledge of other agents' positions and visitation history based on proximity.

Acknowledgements

We would like to extend our gratitude to:

Prof. You Yang: Inspiring lectures on SOTA DL algorithms

Fuzhao, Xiangyu (TAs): For their patience and mentorship

MARMOT Lab @ NUS: Author of DAN[1] and ARiADNE[2]

[1] Cao, Y., Sun, Z., & Sartoretti, G. (2021). *DAN: Decentralized Attention-based Neural Network to Solve the MinMax Multiple Traveling Salesman Problem* (arXiv:2109.04205). arXiv. http://arxiv.org/abs/2109.04205
[2] Cao, Y., Hou, T., Wang, Y., Yi, X., & Sartoretti, G. (2023). *ARiADNE: A Reinforcement learning approach using Attention-based Deep Networks for Exploration* (arXiv:2301.11575). arXiv. http://arxiv.org/abs/2301.11575