Reinforcement Learning Based Solutions for Scheduling Astronomical Observations: A DDPG Approach

Mark Haddad, Department of Computer Science
Faculty Mentor: Dr. John Korah
RSCA 2025

Introduction

- Efficient utilization of observatory time is crucial, as manual scheduling limits observation efficiency.
- The Astronomer's Traveling Salesman Problem (Astro-TSP) is an NP Hard scheduling problem applied to observatories [3].
- Constraints include time windows for each observation, time dependence, and varying observation priorities.
- These complexities necessitate a tailored solution beyond the classical TSP framework.

Background

- We propose a reinforcement learning-based approach to solve the Astro-TSP efficiently.
- Reinforcement Learning (RL) is a subset of Machine Learning that trains an agent based on experience in an environment, as opposed to labeled training data.
- Our previous work in Q-Learning improves scheduling by updating a table of discretized states and actions.
- To address Q-Learning's limitations in memory management and slow convergence [2], we introduce Deep Deterministic Policy Gradients (DDPG) [1], which supports continuous state-action spaces and enhances memory management by storing weights instead of a large Q-Table.
- Unlike other Actor-Critic methods, DDPG is an off-policy, deterministic method that uses past experiences in batches to directly determine actions, instead of on-policy methods using only recent experience.
- RL methods adapt to the dynamic and combinatorial nature of astronomical scheduling, considering long-term over local optimization.
- We will implement a DDPG agent and compare its performance against Q-Learning, a traditional greedy heuristic, and a random heuristic.

Methodology

- We represent observation scheduling in states and actions.
- Each state and action is generalized via the environment constraints.
- States: $s = \frac{p_a}{\sum p}$, accumulated priority (p_a) over the total priority ($\sum p$) obtainable in the environment.
- Actions: $a = \frac{p}{d}$, priority (p) over distance (d) ratio of each neighboring observation.
- States and actions range from [0, 1], action space is min-max normalized for this range.
- The agent's goal is to visit all observations in an environment maximizing reward, considering all constraints.
- RL agents formulate their reward system considering these constraints.
- Reward: $\frac{p}{d} + \frac{1}{t_r}$, time remaining ($t_r$) considers the time window length minus the travel time and observation length for each visit.

DDPG Training Process for Astro-TSP

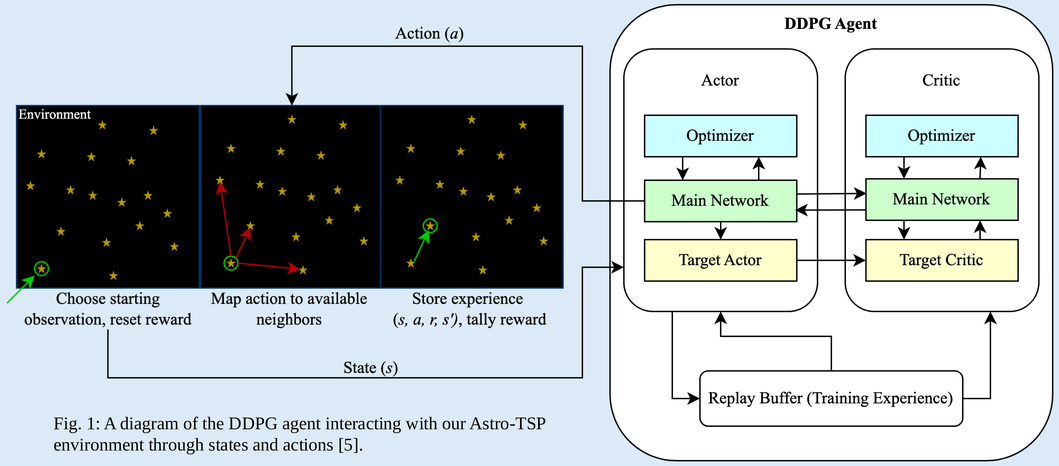

Fig. 1: A diagram of the DDPG agent interacting with our Astro-TSP environment through states and actions [5].

DDPG Components

- Consists of four neural networks (Fig. 1):
  - Actor: Maps states to actions deterministically.
  - Critic: Estimates the Q-value of state-action pairs.
  - Target Actor: A delayed copy of the actor used for soft updates.
  - Target Critic: A delayed copy of the critic for target Q-value estimation.
- Uses a Replay Buffer to store past experiences and randomly sample during training, and soft updates to the target. networks that stabilize the Critic's value estimations.
- Gaussian Noise variable when choosing actions for exploration.

Implementation [1]

- The actor policy gradient ($\nabla_{\theta\mu} J$) determines the action output in the normalized range from 0 to 1.
- Given the expectation, or average, ($\mathbb{E}_{s_t\sim\rho^\beta}$) of the gradient of the critic network ($\nabla_{\theta\mu} Q$) with states and actions chosen according to current policy ($\mu(s_t|\theta\mu)$), we represent the actor policy as:

$$\nabla_{\theta\mu} J \approx \mathbb{E}_{s_t\sim\rho^\beta}\left[\nabla_{\theta\mu} Q(s,a|\theta^Q)|_{s=s_t,a=\mu(s_t|\theta\mu)}\right]$$

- The target Q-value (y_i) is computed through a target actor target critic sequence, and the critic loss (L) is calculated using the main ($Q(s_i,a_i|\theta^Q)$) and target (y_i) critic values.

$$y_i = r_i + \gamma Q'\left(s_{i+1}, \mu'\left(s_{i+1}|\theta^{\mu'}\right)|\theta^{Q'}\right) \qquad L = \frac{1}{N}\sum_i (y_i - Q(s_i,a_i|\theta^Q))^2$$

- The actor policy ($\nabla_{\theta\mu} J$) is updated with its parameters, and the critic Q-values ($Q(s,a|\theta^Q)$ changes with respect to actions (∇_a).

$$\nabla_{\theta\mu} J \approx \frac{1}{N}\sum_i \nabla_a Q(s,a|\theta^Q)|_{s=s_i,a=\mu(s_i)} \nabla_{\theta\mu}\mu(s|\theta^\mu)|_{s_i}$$

- The target network weights are slowly updated at the end of each step.

$$\theta^{\mu\prime} \leftarrow \tau\theta^\mu + (1-\tau)\,\theta^{\mu\prime} \qquad \theta^{Q\prime} \leftarrow \tau\theta^Q + (1-\tau)\,\theta^{Q\prime}$$

Dataset

- Coordinates of each observation are tracked using Right Ascension (RA) and Declination (DECL).
- Our dataset records multiple instances of observation time, the air mass at each instance, and exposure time affected by the air mass for each observation entry.
- The time windows are formed by ranging the observation time, and our observation length is interpolated to the exposure time of the given time in that time window.
- Our dataset consists of 170 observations generated across 6 different nights of telescope time. Each night in our dataset records between 24 and 37 observations.

Results

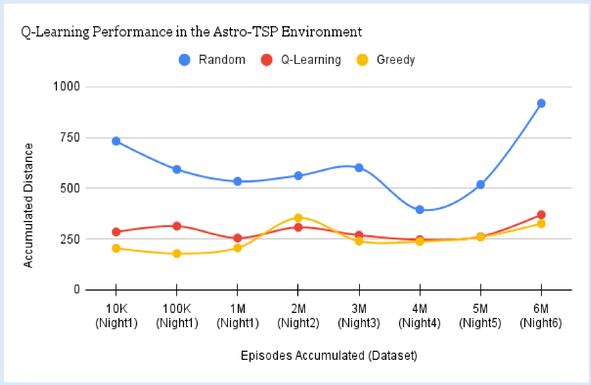

Fig. 2: Graph of training performance of Q-Learning that considers priority and time windows

- Previous Q-Learning results (Fig. 2) shows equivalent scheduling performance to a Greedy algorithm, with both algorithms significantly better than a random algorithm.
- Results for DDPG are still ongoing and in progress. As we start to see results in the future after the full implementation, we anticipate to see a faster reward accumulation and convergence compared to Q-Learning and the other heuristics.

Significance of Expected Results

- It is hypothesized that DDPG will outperform Q-Learning in our Astro-TSP environment for the following reasons:
  - Although our environment is a discrete coordinate map, we have represented our state and action spaces in coordinate-agnostic ratios based on the priority and distance constraints.
  - This represents our action space with a continuous value, comparable to a smooth volume knob rather than several buttons to change the state.
  - Q-Learning discretizes our action space into millions of action options.
  - Instead of storing values for every possible action, DDPG directly learns the best action that we can conversely map to an action option in our environment.
  - In a continuous environment, this methodology promotes faster convergence, computation, improved memory management, and overall better environment representation with a more stable learning process.
  - Through this process we hope to support the claim that DDPG handles continuous action spaces much better than Q-Learning [1][2][4], thus is an improved solution for Astro-TSP.

Conclusions

- We hypothesize that the move from a discrete to a continuous action space representation would improve the scheduling performance for the Astro-TSP.
- Results for DDPG's training and testing performance are in progress, and we aim to compare DDPG with Q-Learning, and a previous Genetic Algorithm implementation [3].
- Both RL models will utilize the same Environment implementation, and each model can be tested interchangeably.

References

[1] T. Lillicrap, J. Hunt, et al. (2015). Continuous control with deep reinforcement learning. 2016 ICLR Conference. https://doi.org/10.48550/arXiv.1509.02971
[2] Shripad V. Deshpande, Harikrishnan R, Babul Salam KSM Kader Ibrahim, Mahesh Datta Sai Ponnuru, Mobile robot path planning using deep deterministic policy gradient with differential gaming (DDPG-DG) exploration, Cognitive Robotics, Volume 4, 2024, Pages 156-173, ISSN 2667-2413, https://doi.org/10.1016/j.cogr.2024.08.002
[3] M. Humphries, "Astro-TSP: Traveling Salesman Problem Based Solutions for Scheduling Astronomical Based Observations" California State Polytechnic University, Pomona, 2024, http://hdl.handle.net/20.500.12680/f7623m070
[4] Rahul, V. S., & Chakraborty, D. (2023). Exploring reinforcement learning techniques for discrete and continuous control tasks in the MuJoCo environment. https://doi.org/10.48550/arXiv.2307.11166
[5] Gong, H., Wang, P., Ni, C., & Cheng, N. (2022). Efficient Path Planning for Mobile Robot Based on Deep Deterministic Policy Gradient. *Sensors*, *22*(9), 3579. https://doi.org/10.3390/s22093579
[6] Tabor, P. "Everything You Need to Know About Deep Deterministic Policy Gradients (DDPG) | TensorFlow 2 Tutorial," YouTube, 2020. [Online]. Available: https://www.youtube.com/watch?v=4jh32CvwKYw