Distributed Q-learning with Gittins Prioritization

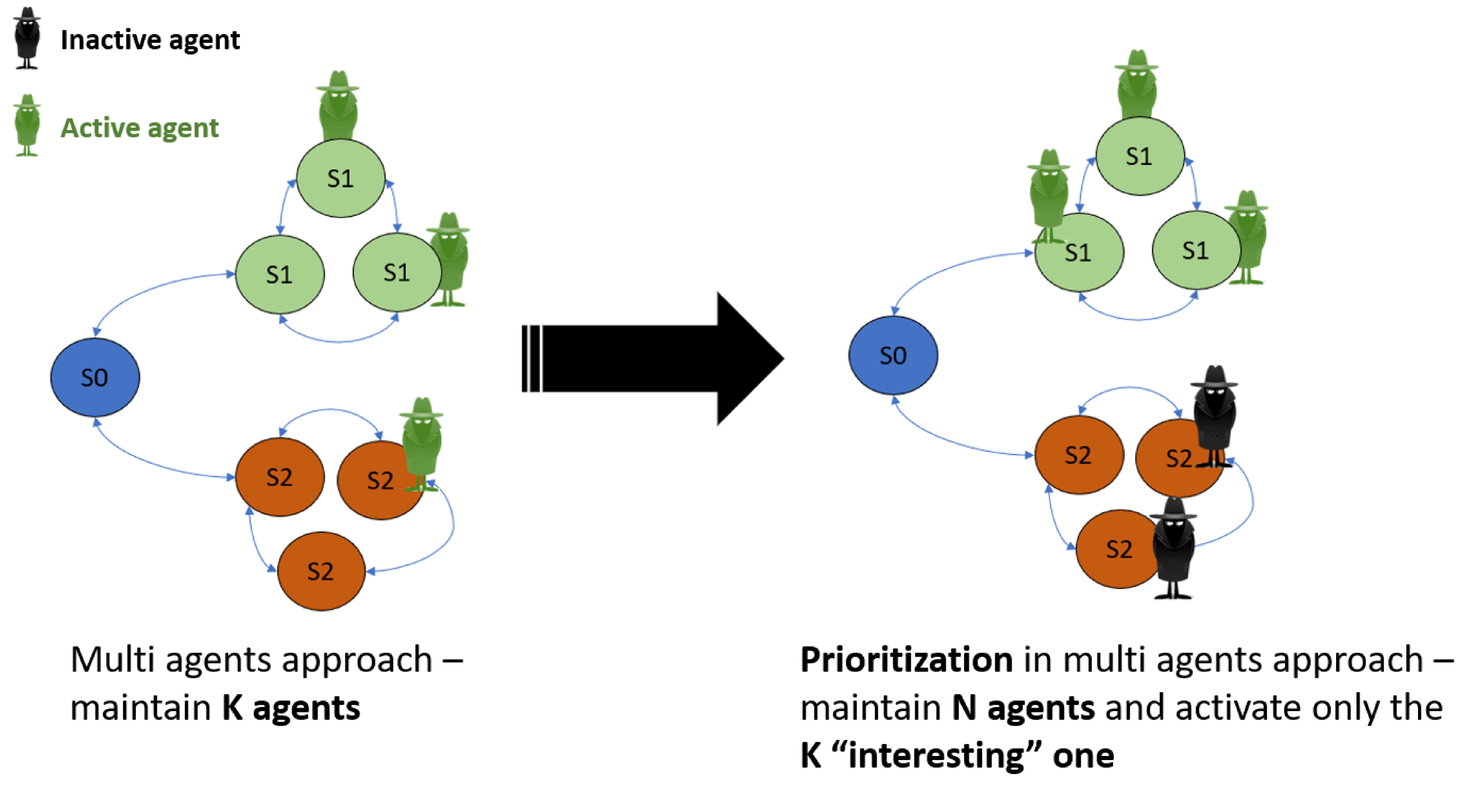
Jhonathan Osin, Naama Pearl, Tom Zahavy, Chen Tessler, Shie Mannor

yoni.osin@gmail.com, naama.pearl@gmail.com, tomzahavy@campus.technion.ac.il, chen.tessler@campus.technion.ac.il

1. Introduction

Main Contribution

- Distributed RL framework:
  - N agents share the same policy
  - At each step, $K < N$ agents are selected to act
  - A prioritization mechanism selects between the agents
- Sampling resources are used more efficiently!



Related Work

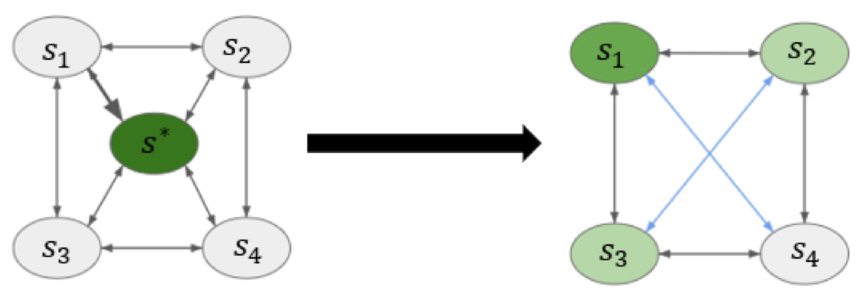
1. **Prioritized Sweeping:** Off policy algorithm which samples state-action pairs according to some index policy - David Andre et al. Generalized prioritized sweeping. InNeurIPS, 1998.
2. **Prioritized experience replay**: Prioritized the reused experiences of online agents Tom Schaul et al. Prioritized experience replay. arXiv, 2015
3. **Prioritizing starting states**: Prioritize the starting state of agents in off-policy algorithms, based on past observation - Tavakoli, Arash, et al. Prioritizing Starting States for Reinforcement Learning. arXiv, 2018

2. Gittins Index

Gittins Index

- **The problem**: given K Markov Reward Process (MRP), in each timestep one process should be activated while others remain frozen. The goal is to **maximize** the accumulated reward.
- The problem of choosing which process to activate seems to have combinatorial complexity.
- Surprisingly, it can be calculated with index policy, **independently** for each process.

$$\nu^{\pi_i}(s) = \sup_{\tau>0} \frac{\mathbb{E}^{\pi_i}[\sum_{t=1}^{\tau} \gamma^t r_i(s_t)|s_0 = s]}{\mathbb{E}\sum_{t=1}^{\tau} \gamma^t}$$



Gittins Index in our framework

- Under a **specific policy**, a Gittins index can be calculated for each state of the MDP.
- Considering each agent as a process, choosing the agent in the state with the highest index will maximize the future accumulative reward.

We show **theoretically** and **empirically** that calculating the Gittins index in the **approximate model** is also the optimal policy of choosing the agents.

Gittins Index in approximate model

- Gittins Index theorem is the optimal policy in a planning problem, when the model is known.
- In our framework, the model is unknown. The learning process aims to aprroximate the problem.
- We prove that also in an approximate model the Gittins index is the optimal policy.

Theorem 1. *The optimal policy for choosing K agents within all N possible operating in an MDP, considering an approximate model of the environment, is to greedily select the best agents based on their Gittins Index.*

Model free Gittins index

- We propose an estimate of the Gittins Index in a model-free environment.
- The empirical value of the index is calculated in 2 phases:
  1. An estimation based on weighted average of discounted accumulated reward is calculated for every trajectory length in $\tau \in [0, T]$, where $T \propto \frac{1}{1-\gamma}$, is the effective horizon of the problem.
  2. The final index is the maximal estimation from those calculated in step 1.

$$\hat{\nu}^{\pi}(s) = \max_{\tau\in[0,T]} \frac{1}{m} \frac{\sum_{t=1}^{\tau}[\gamma^t r_i(s_t)|s_0 = s]}{\sum_{t=1}^{\tau} \gamma^t}$$

3. Method

Framework

- N **agents** interact with a **single unknown MDP**.
- At each timestep a **subset of** k **agents** are prioritized to advance. Other remain frozen.
- A **global policy**, is learned using **Q-learning** based on all agents observation.

Prioritization Schemes

During the learning process the **score of each state** is periodically calculated, based on:

1. *reward* - $r(s, \pi(s))$
2. *TD error* - $r(s, \pi(s)) + \gamma \cdot \arg\max_a Q(s', a) - Q(s, \pi(s))$

Above scores yielded **4 prioritization schemes**:

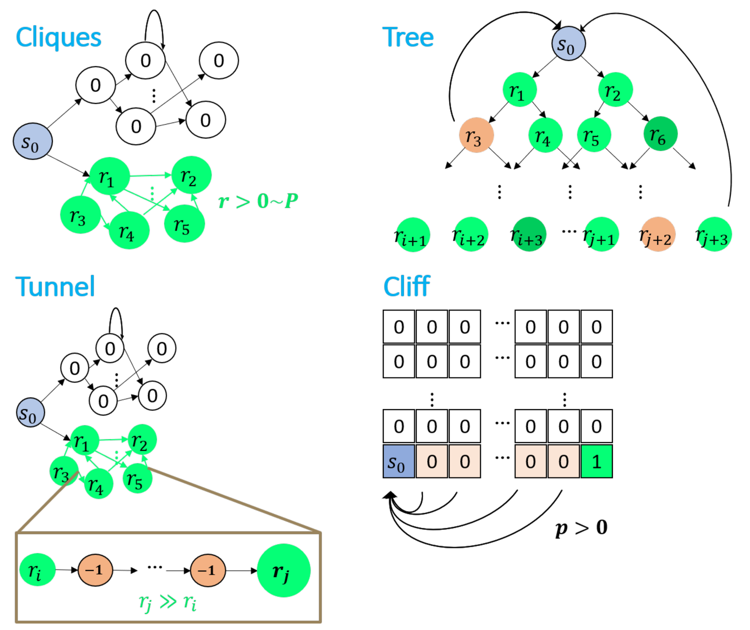
- greedy reward
- greedy TD-error
- Gittins reward
- Gittins TD-error

Performance, compared to **random prioritization** baseline, was evaluated using:

- Online regret
- Periodic offline evaluation of the learned policy

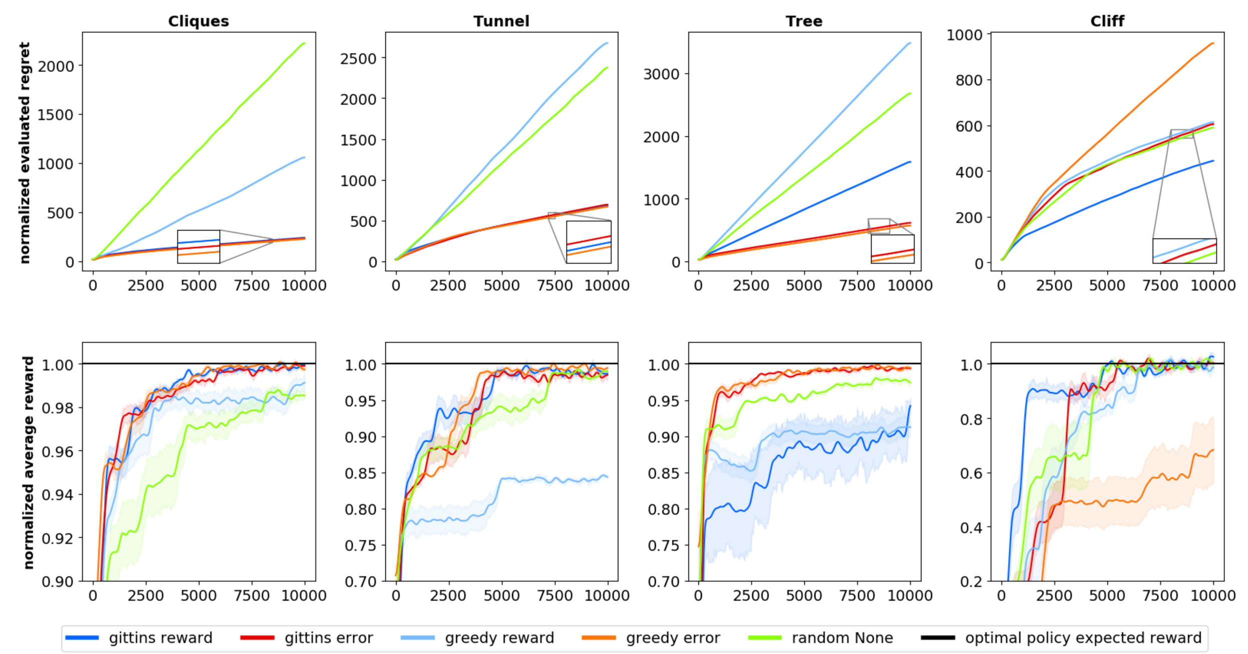
4. Experiments

MDP's



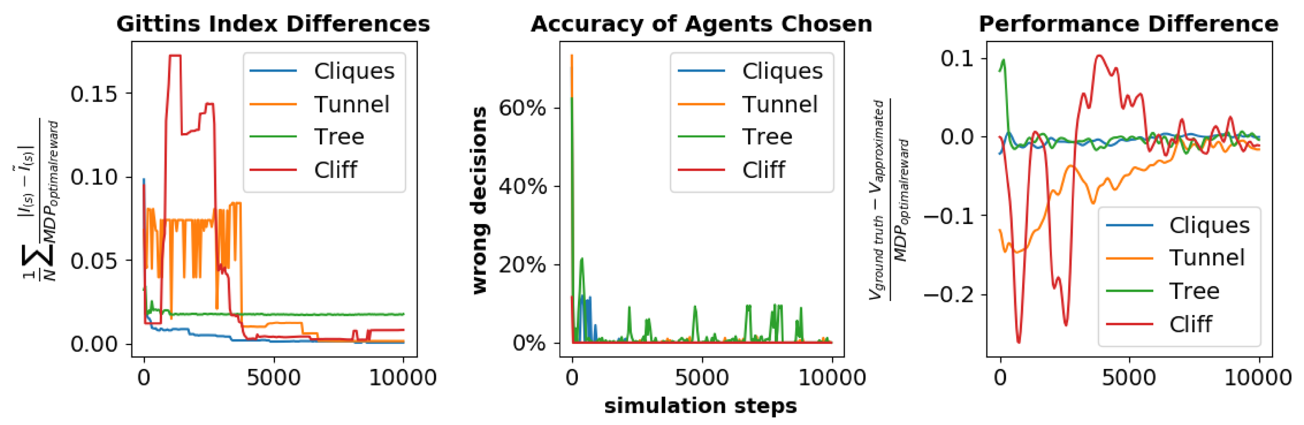
Performance analysing: Regret and average reward

- In most scenarios, **prioritization is better** than random selection
- **Gittins approach based on the TD-error** has the most consistent positive effect across all domains.



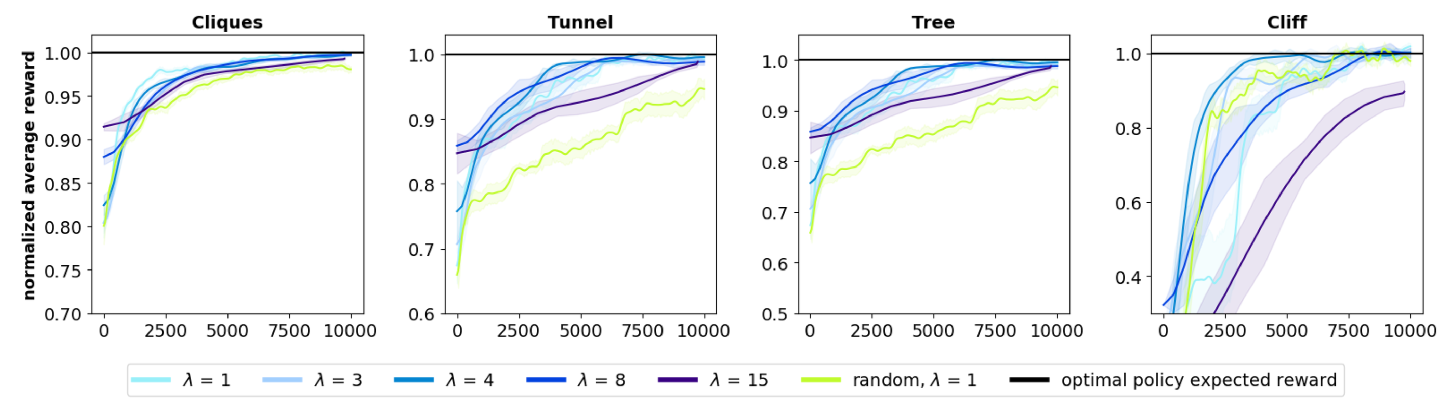
Gittins Indices Accuracy

Gittins Index calculated in the **approximate model** were compared to those in the **real model**:



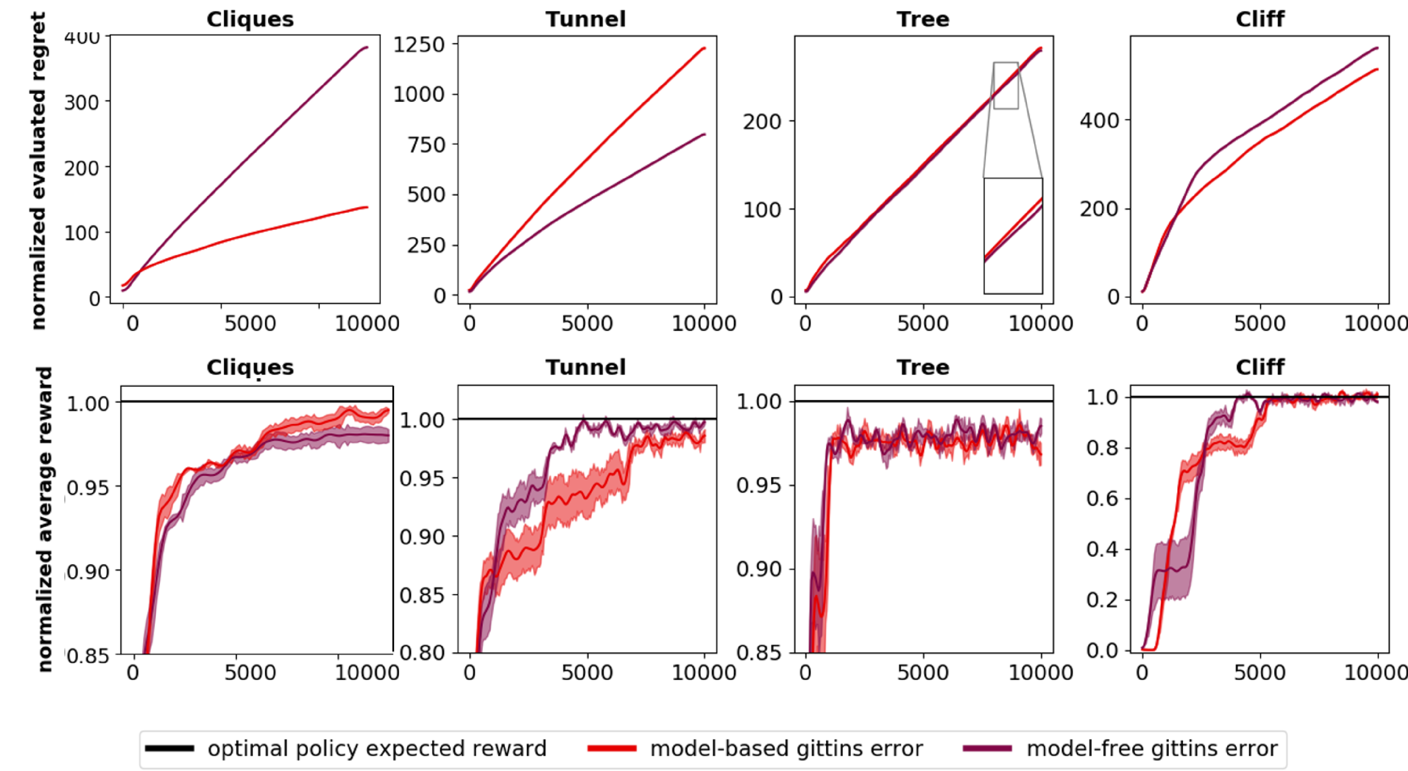
Temporal Extension Implication

- Exploring the implication of selecting the agents prioritized each $\lambda > 1$ timesteps (rather then every timestep).
- Using the temporal extension can improve performance for small values of λ.
- Almost in all the MDPs, adding temporal extension still resulted in improvement over the random baseline.



Model Free Approach

We show empirical results comparing our methods in a model free environment:



5. Summary

1. Prioritization based on the TD-error is the best approach
2. Prioritizing based on the Gittins Index is robust to temporal extensions
3. Using an approximate model to estimate the Gittins Index results in near-optimal performance when compared to using the exact model

6. Work in progress

- Based on prior work, we constructed a DNN which, given a state, approximates it's score (based on the Gittins index)
- We integrated the above DNN into a standard A2C algorithm to investigate the effect of our approach in more complex domains
- We get similar results in several Mujoco and Atari domains.
- Our future plans include adding a network to approximate the Gittins index, testing others domains and promoting agents with off-policy actions for exploration