Can reinforcement learning solve a Human Allocation Problem?

Phong Nguyen, Matsuba Hiroya, Tejdeep Hunabad, Dmitrii Zhilenkov, Hung Nguyen and Khang Nguyen

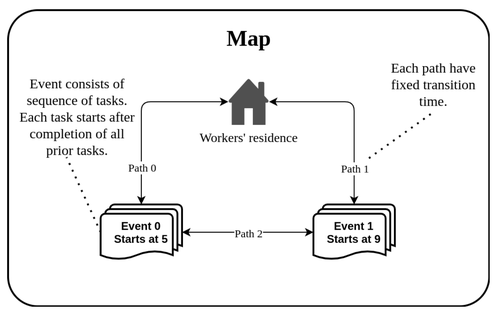
Hitachi Center of Technology Innovation
FPT Software

Abstract

In this paper, we consider the human allocation problem (HAP), which is an example NP-hard class, with the fixed number of workers and tasks, using various RL methods, such as deep contextual bandit, double deep Q-learning network (DDQN), and a combined approach of DDQN with the Monte Carlo Tree Search (MCTS).

Preliminaries

- We consider the HAP with m workers, k events and n tasks.
- Each event consists of fixed amount of sub-tasks t, therefore $n = k \cdot t$, but this assumption could be relaxed for an arbitrary amount of sub-tasks.
- Events have several properties, e.g. starting time, location, task sequence. (i.e. a worker should move between events according to the transition time and they cannot start following sub-task in the sequence while preceding sub-tasks are not completed).
- The assignment is a representation of a map between the set of workers and the set of tasks.
- There is an environment simulator that receives an assignment and estimates total time needed to complete all tasks according to *workers' capabilities* (vectors of allotted time to complete each task)
- The goal is to find an assignment that provides minimal time costs.



General approach

We use deep reinforcement learning frameworks in order to study their performance on HAP, and find the ways how to scale up these methods to real world scenarios. All chosen frameworks are based on the common RL scheme, where the agent play action in a given state and the environment provides a response with the reward and a new state.
Deep models were implemented to approximate a $Q : S \times A \to \mathbb{R}$ function. This function as usually shows the discounted value of action a in state s and guide the agent during the decision process.
The aim was to study the behavior of all chosen methods on the classical discrete optimization problem (HAP) and find the ways to organize a successful training process for HAP environment, that contains non-trivial constraints for classical optimization.

Contextual bandit

In this case the RL-agent receives the state s from the environment and choose the action a_i from the set of actions A, where i is index of action. The context of each action is given as (s, a_i) tuple. The environment takes this action and provides the reward r, and an episode terminates. The training process consists in running episodes, where the agent learns the distribution of rewards for given contexts.

- In case of HAP the state s represents initial environment configuration with workers' capabilities to solve each task, transitions time to move between events and start time of each event.
- The main issue here is that the context a_i represents full assignment. Total amount of all assignments is m^n. Therefore this approach is available only for significantly small amount of workers and tasks because the agent have to predict the value of each action based on its context.
- The question, "Is it possible to avoid exponential number of predictions?", still needs further researches.

DDQN and MCTS

Deep Q-learning is an approach provided an opportunity to use RL ideas in more complex environments. The combination of RL and neural networks gives a way to generalize similar states or actions and use a learned policy even in case of unique (state, action) pair (i.e. the RL-agent can use a policy during the decision process in unexplored states if they are similar to the preceding experience).
Double DQN improves the baseline of DQN and makes a training process more stable.
In general, Monte Carlo Tree Search plays the role of policy evaluation and policy improvement operator.
We implemented DDQN and DDQN with MCTS algorithms to check the opportunity of RL-agent to learn the policy in case of Human Allocation Problem and compared the results.

Results and conclusion

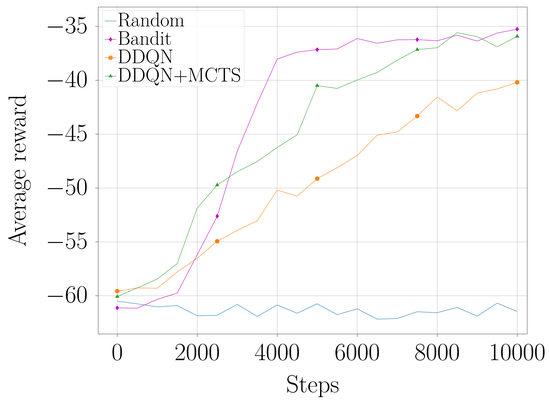


Figure 1:Performance comparison of Contextual Bandit, DDQN, DDQN+MCTS in the configuration of $m = 3$, $n = 2$. Learning curves of three methods based on the best practice results with the simulation of 10000 steps.

Table 1:The lower bound, average, upper bound of each method in various configurations in increasing order of complexity.

Config	Method	Lower Bound	Average	Upper Bound
$m = 3$ $n = 4$	Contextual Bandit	**131.88**	**135.19**	**138.24**
	DDQN	118.46	127.61	130.21
	DDQN+ MCTS	125.23	130.79	133.43
$m = 3$ $n = 9$	DDQN	114.83	126.27	129.43
	DDQN+ MCTS	**121.86**	**128.19**	**130.23**
$m = 5$ $n = 16$	DDQN	104.28	116.96	125.39
	DDQN+ MCTS	**117.54**	**124.58**	**127.59**
$m = 6$ $n = 25$	DDQN	95.92	107.49	118.78
	DDQN+ MCTS	**112.59**	**120.84**	**124.24**
$m = 7$ $n = 36$	DDQN	85.97	108.61	112.18
	DDQN+ MCTS	**106.58**	**118.57**	**122.54**

We use the mean Random Normalized Score (RNS) as the primary metric for comparing between methods. Typically, the random performance is taken as the baseline, and the RNS obtained by an agent on the allocation problem can be measured relative to representative random performance. Each method's mean score is calculated by the best practice results of all experiments.
The experimental data shows a potential utility of usage RL for the case of classical discrete optimization problems, such as HAP, and the following researches are needed to answer the question about implementation of this methods for the real scale scenarios.