Deep Reinforcement Learning and Application to the Intrusion Response Case: TaxiAssignment

Luca Maria Lauricella 519997

July 2023

1 Introduction

In this project, our objective is to create a new environment for the taxi-v3 problem. The enhancement involves introducing a fuel level concept with 5 refueling stations, comprising 4 randomly placed stations and one located near the taxi's starting point, enabling refueling during its operations. Each movement executed by the taxi will consume a certain amount of fuel, and the episode will conclude when the fuel level reaches zero.

2 State Initialization

The first step entails instantiating the fuel stations position and setting the tank capacity to a maximum of 10 fuel levels. Consequently, the number of states will significantly increase compared to the base case, in which there are 500 states calculated by multiplying together 5 columns, 5 rows, 5 passenger positions, and 4 possible destinations. Additionally, the number of available actions will be updated to 7.

3 Rewards and action mask updating

Next, we will construct the state matrix and define the rewards associated with specific actions. Positive or negative rewards will be assigned based on the chosen action, and the episode will terminate if the fuel level depletes and becomes less than or equal to 0. We will proceed by updating the decode and encode functions, introducing new parameters, and implementing the action mask function. The action mask function will be utilized to determine when a particular action can be executed (mask[i]=1). Furthermore, we will handle a scenario where the taxi is positioned at a refueling station. In this case, the corresponding action mask will be set to 1, indicating that the taxi can refuel at that specific location.

4 Training

The training times of the two algorithms differ significantly. In the new extended version, with over 100,000 states, it requires many more episodes for the algorithm to modify the internal values of the matrix and assign probability values to all possible actions. It's evident that the number of states heavily impacts the algorithm's convergence, and the base case is undoubtedly more simple and with fewer negative rewards.

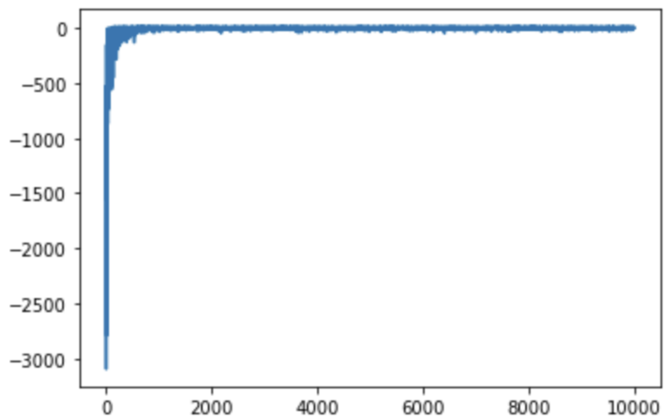

Figure 1: Reward Graph of Base Case

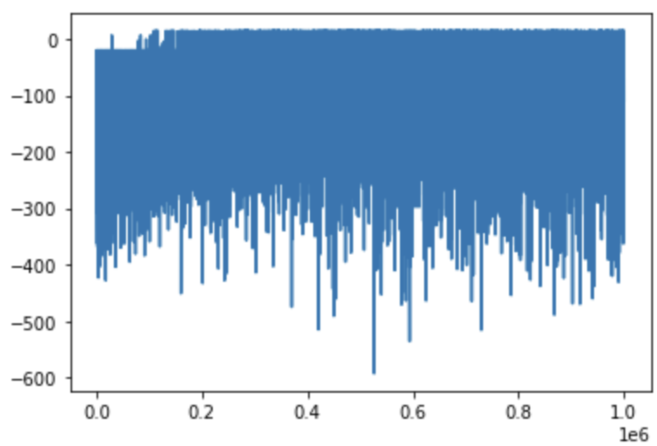

Figure 2: Reward Graph of Extended Case

To speed up the training phase and enhance the algorithm's accuracy, one can achieve this by reducing the number of states. For instance, this can be done by fixing all possible positions of the refueling stations or reducing the number of fuel levels.

If it's not possible to reduce the input dimension or optimize the agent's parameters and algorithm hyperparameters, an alternative approach could be to consider parallel processing on GPU or CPU to speed up the training process.