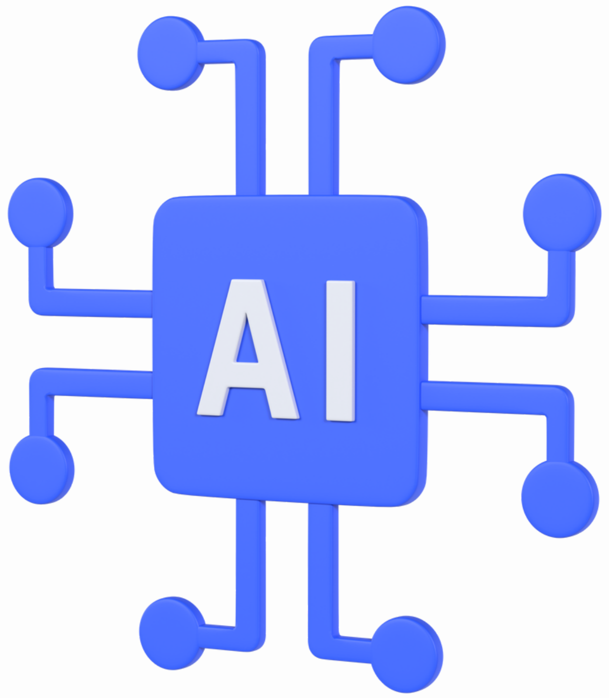
ARTIFICIAL INTELLIGENCE

J COMPONENT - REVIEW 3



AI AGENT TO HELP CAB DRIVERS MAXIMISE PROFIT WITHIN A MONTH

TEAM MEMBERS :

SHYLESH SHANKAR V -20BCE1314

AKSHAY KUMARAN V-20BCE1433

LOKESHWAR M-20BCE1825

INTRODUCTION

We are developing an AI agent to help cab driver's maximize the profit. Our agent is trying to aim at maximizing profit based on the requests it picks up, so we are using Reinforced Learning (RL). Q-learning is a model-free reinforcement learning algorithm to learn the value of an action in a particular state and creates an exact matrix for the working agent which it can “refer to” to maximize its reward in the long run. It can handle problems with stochastic transitions and rewards without requiring adaptations.

PROBLEM STATEMENT

As we saw after this pandemic situation, the rise in cab drivers is huge. Everyone needs a secure and cheap ride to travel. Competition between the cab service providers is high, and as a result, the cab drivers are having difficulty making decisions. The post-pandemic economy has an impact on fuel prices. Many drivers complain that although their revenues are gradually increasing, their profits are almost flat. Thus, it is important that drivers choose the right rides, i.e., choose the rides that are likely to maximise the total profit earned by the driver that day.

MOTIVATION

Our motivation is to provide an AI agent to make the right decision in choosing the right ride, which makes profit for the cab drivers. For example, say a driver gets three ride requests at 5 p.m. The first one is a long distance ride, guaranteeing a high fare, but it will take him to a location where it is unlikely to get him another ride for the next few hours. The second one ends in a better location, but it requires him to take a slight detour to pick up the customer, adding to fuel costs. Perhaps the best choice is to choose the third one, which, although medium-distance, will likely get him another ride subsequently and avoid most of the traffic. Considering this factor, our AI agent will help cab drivers pick up the best request at any time they are free.

ABSTRACT

The efficiency of taxi services in big cities is Urban transport and benefits as well as the convenience of people's mobility Taxi driver. To balance the demand and supply of taxis, Spatiotemporal knowledge obtained from historical trajectories Both passengers are advised to find an available taxi A taxi driver estimates the location of the next passenger. However, Taxi Lane is a long sequence where he is one step. Optimization cannot guarantee global optimization. With the aim of generating long-term returns, new methods have been proposed based on. Reinforcement learning to optimize taxi driving strategies for global profit maximization.

LITERATURE SURVEY 1

Paper 1

Reinforcement Learning for Optimizing Driving Policies on Cruising Taxis Services

The reinforcement learning (RL) framework was established in this paper to improve driving practices for cruising taxi services. First, we developed the drivers' behaviors as the Markov decision process (MDP) advanced, taking long-term effects into account. Different objects were aimed to increase taxis services’ efficiency, such as minimization of waiting time, minimization of the distance between the drivers and passengers, maximization of the expected profits, and the probability of finding potential passengers.

LITERATURE SURVEY 2

Paper 2
A Deep Reinforcement Learning for List-wise Recommendations

In recent years, recommender systems have grown in popularity and have been applied to a range of fields, including music, literature, movies, search queries, and social tags. The majority of recommender systems currently in use are created to maximise the short-term rewards of recommendations, i.e., to get users to order the suggested items, while completely ignoring the possibility that these suggestions will result in future benefits that are more likely or profitable (long-term).

PROPOSED ARCHITECTURE

In our project we used a Deep Q neural network as the brain of our agent than simple Q – network. Since deep learning models have more learning capacity than other models. Thus, it can show improvement in performance even with dynamic environment with changing state and action space.

1. The Markov Decision Process we will be defining the state space , action space , reward function and terminal state for an episode, requests generator, next state transition.
2. Policy learning and dynamic programming
3. Agent designing using reinforcement learning, DQN.

IT'S FUNCTIONALITY

- We use Markov Decision Process (MDP) to describe the environment. The Markov Decision Process (MDP) is typically used to model sequential decision-making problems, which is the necessary process of RL

- States we are focusing on are Locations, Time of the day and Day of the week. We send the state space as a vector to our learning model.

- The action space contains the possible combinations of travelling between any two locations.

- For the reward function we calculate the remaining amount available by subtracting the price of fuel used between the pickup and end points.

- We use Poisson distribution for generating probable number of requests that can come if the cab driver is at a particular location. Based on the number we randomly pick up that many number of samples from action space and add it to possible action space.

RESULTS

AGENT TESTING

TEST 1

```
agent = CabDriver()
agent.test_run()
```

```
CURRENT STATE: (4, 3, 6)
REQUESTS: [(4, 1), (1, 0), (4, 3), (3, 1), (2, 0), (1, 3), (2, 4), (0, 0)]
REWARDS: [8.0, 14.0, 8.0, 6.0, -1.0, 6.0, -9.0, -5]
NEW POSSIBLE STATES: [[1, 5, 6], [0, 11, 6], [3, 5, 6], [1, 9, 6], [0, 14, 6], [3, 9, 6], [4, 12, 6], [4, 4, 6]]
MAXIMUM REWARD: 14.0
ACTION : (1, 0)
NEW STATE: [0, 11, 6]
NN INPUT LAYER: [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
```

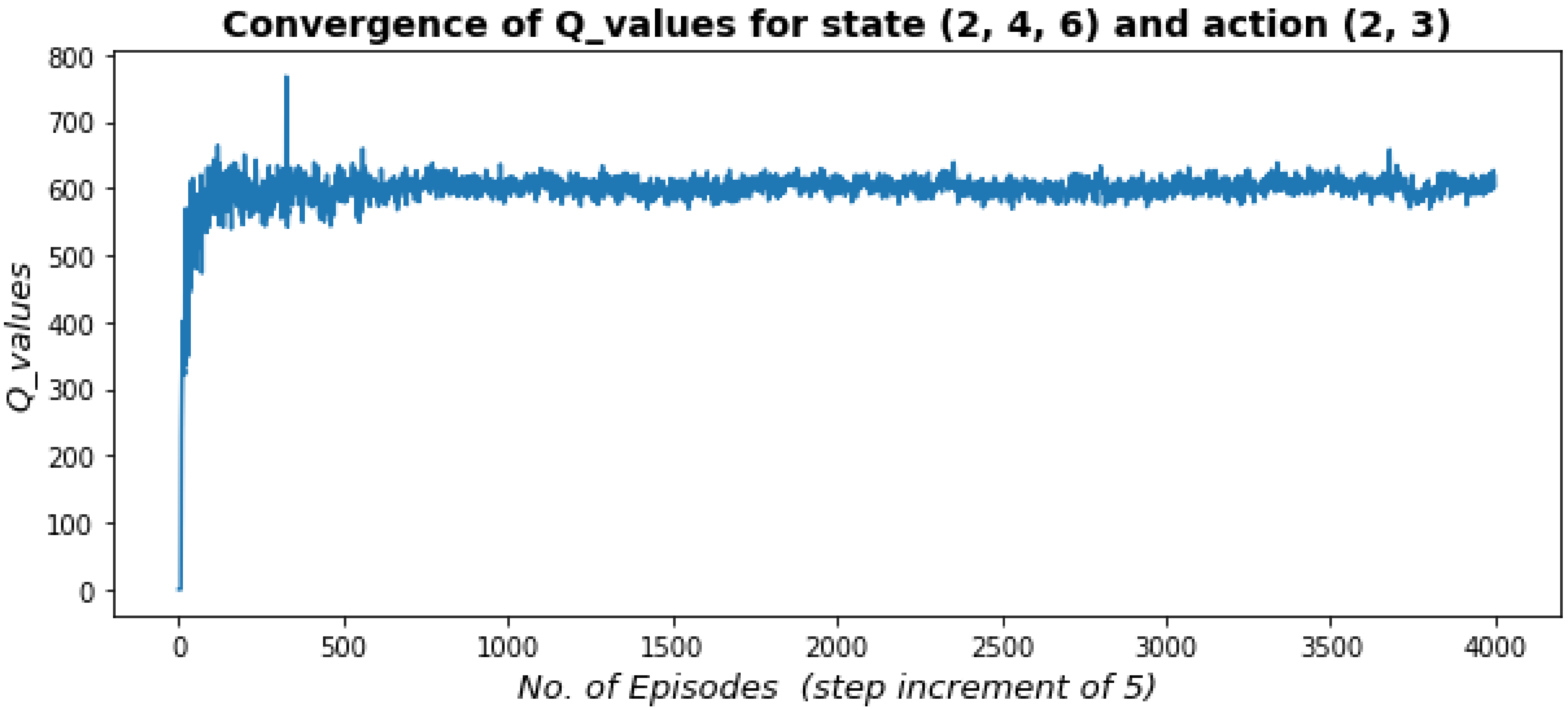
TEST 2

```
agent = CabDriver()
agent.test_run()
```

```
CURRENT STATE: (2, 10, 2)
REQUESTS: [(1, 3), (4, 2), (2, 1), (1, 4), (4, 0), (3, 0), (4, 1), (0, 0)]
REWARDS: [-16.0, -12.0, 16.0, -20.0, -16.0, -7.0, -20.0, -5]
NEW POSSIBLE STATES: [[3, 15, 2], [2, 16, 2], [1, 14, 2], [4, 14, 2], [0, 15, 2], [0, 15, 2], [1, 14, 2], [2, 11, 2]]
MAXIMUM REWARD: 16.0
ACTION : (2, 1)
NEW STATE: [1, 14, 2]
NN INPUT LAYER: [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0]
```

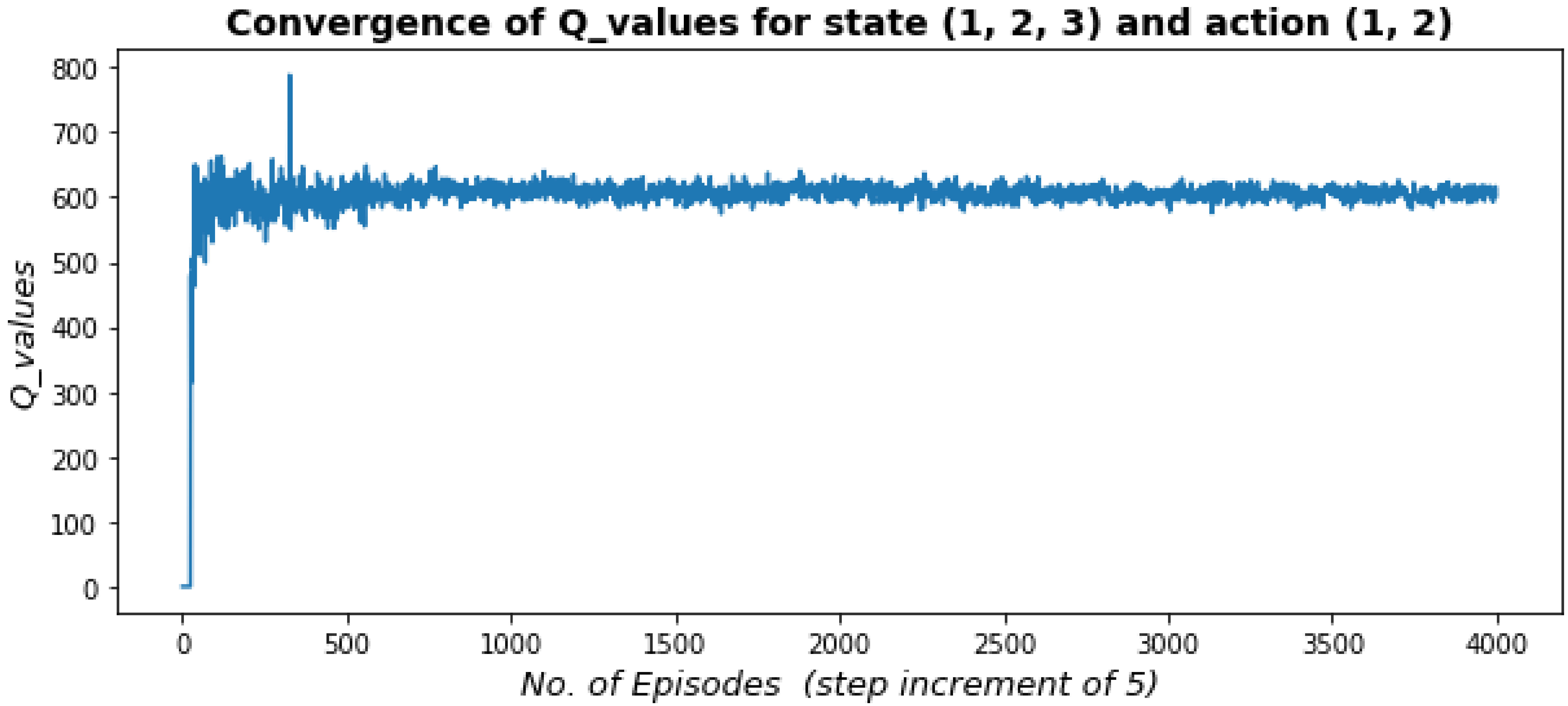
CONVERGENCE GRAPH

PAIR 1

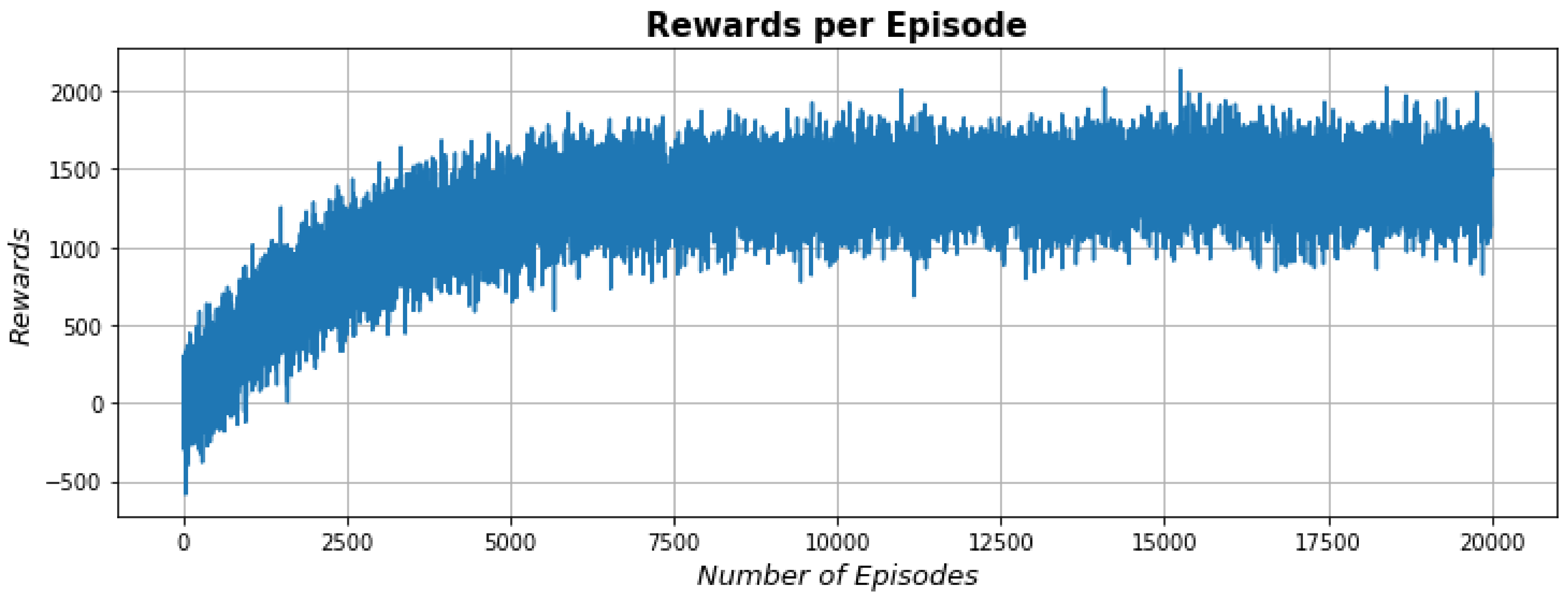


PAIR 2

```
Tracking Convergence for state-action pair 2: State (1,2,3), Action (1,2)
```



REVENUE GRAPH



LIST OF SURVEY PAPERS

1) Liang, E., Wen, K., Lam, W. H., Sumalee, A., & Zhong, R. (2021). An integrated reinforcement learning and centralized programming approach for online taxi dispatching. IEEE Transactions on Neural Networks and Learning Systems.

2) Okuyama, T., Gonsalves, T., & Upadhay, J. (2018, March). Autonomous driving system based on deep q learnig. In 2018 International conference on intelligent autonomous systems (ICoIAS) (pp. 201-205). IEEE.

3) Rong, H., Zhou, X., Yang, C., Shafiq, Z., & Liu, A. (2016, October). The rich and the poor: A Markov decision process approach to optimizing taxi driver revenue efficiency. In Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (pp. 2329-2334).

4) Tong, W., Hussain, A., Bo, W. X., & Maharjan, S. (2019). Artificial intelligence for vehicle-to-everything: A survey. IEEE Access, 7, 10823- 10843.

5) Jin, K., Wang, W., Hua, X., & Zhou, W. (2020). Reinforcement Learning for Optimizing Driving Policies on Cruising Taxis Services. Sustainability, 12(21), 8883.

LIST OF SURVEY PAPERS

6) Gao, Y., Jiang, D., & Xu, Y. (2018). Optimize taxi driving strategies based on reinforcement learning. International Journal of Geographical Information Science, 32(8), 1677-1696.

7) Zou, Z., Yu, X., & Ergan, S. (2020). Towards optimal control of air handling units using deep reinforcement learning and recurrent neural network. Building and Environment, 168, 106535.

8) Iversen, E. B., Morales, J. M., & Madsen, H. (2014). Optimal charging of an electric vehicle using a Markov decision process. Applied Energy, 123, 1-12.

9) Roda, C., Rodríguez, A. C., López-Jaquero, V., Navarro, E., & González, P. (2017). A multi-agent system for acquired brain injury rehabilitation in ambient intelligence environments. Neurocomputing, 231, 11-18.

10) Lu, R., & Hong, S. H. (2019). Incentive-based demand response for smart grid with reinforcement learning and deep neural network. Applied energy, 236, 937-949.

Thank You