Transactions on Machine Intelligence 6(3) (2023) 175-181

ISSN Online: 2821-1693

Transactions on Machine Intelligence

Journal Homepage: https://tmachineintelligence.ir/index.php/TMI

Using Reinforcement Learning to Find the Shortest Path between two Locations on the Public Roadways

H. R. Naji[1], F. Amirteimoury[2,*]

[1] Associate Professor, Department of Computer Engineering and Information Technology Graduate University of Advance Technology Kerman, Iran
[2] Department of Computer Engineering and Information Technology Islamic Azad University of Kerman, Kerman, Iran

ARTICLE INFO

Article History:
Received 3 April 2023
Received in revised form 7 May 2023
Accepted 26 July 2023
Available 28 September 2023

Keywords:
Reinforcement Learning (RL), Transportation, Single-Agent reinforcement learning, Q-Learning

ABSTRACT

With the rapid increase in urban populations, the proliferation of private vehicles, and the worsening state of air quality, conventional urban transportation planning methods are struggling to meet modern demands. These traditional systems often lack the adaptability and intelligence required to respond effectively to dynamic and complex traffic patterns. In response to these challenges, this study proposes a reinforcement learning (RL)-based approach designed to enhance urban transportation efficiency and sustainability. The core objective of the proposed model is to determine optimal routes between origin and destination points by dynamically avoiding congested areas. Unlike static routing algorithms, the RL model continuously learns and adapts to traffic conditions, enabling the selection of routes that minimize travel time and reduce vehicle idling. As a result, the approach significantly contributes to lowering fossil fuel consumption and energy use, while simultaneously addressing the broader environmental concern of urban air pollution. The integration of artificial intelligence in transportation systems through RL not only enhances service quality and traffic flow but also supports the development of smarter, greener cities. This study underscores the transformative potential of RL in revolutionizing traffic management systems and presents a viable framework for future intelligent transportation networks.

1. INTRODUCTION

Nowadays, global warming is escalating at an alarming rate and has become a major concern among authorities and decision-makers. Expanding fossil fuel consumption is one of the primary reasons contributing to climate change and global warming. It is undeniable that fossil fuels play a critical role in modern life, being utilized in a wide range of areas from domestic use to powering industries and the planet. Also, a significant portion of fossil fuel consumption is attributed to public and private transportation, including vehicles [1,2].

The Intergovernmental Panel on Climate Change (IPCC) in their report "Climate Change 2021 – The Physical Science Basis" highlights the significant impact of fossil fuel consumption on climate change. The report emphasizes

* Corresponding Author: famirteimoury@iauk.ac.ir
Department of Computer Engineering and Information Technology Islamic Azad University of Kerman, Kerman, Iran

http://dx.doi.org/10.47176/TMI.2023.175

that the escalating rate of global warming is directly linked to the increasing use of fossil fuels, which release greenhouse gases into the atmosphere [3]. Furthermore, Masson-Delmotte et al. (2022) in their publication "Climate Change and Land" underscore the role of fossil fuel consumption in exacerbating climate change. The authors elaborate on how land-use changes driven by the demand for fossil fuels contribute to environmental degradation and climate change [4]. Tang et al. (2021) present an alternative perspective in their research on electrocatalytic refinery for sustainable production of fuels and chemicals. The study explores the potential for sustainable production of fuels and chemicals as an alternative to traditional fossil fuel consumption. The findings suggest that advancements in electrocatalytic refinery technologies can offer a more sustainable approach to meeting energy and chemical demands [5]. The paper is structured as follows: the next section outlines the framework of our proposed model; the following section explains the methodology applied in this model; and the final section presents simulations and discusses the results.

2. THE FRAMEWORK OF PROPOSED MODEL

To determine optimized routes, we propose a method that utilizes single-agent reinforcement learning and Q-learning. The following steps comprise this method:

- In the first step, the map is imported as the environment of RL process.
- Source and destination points are being selected by the user.
- Then, the training process is done by Q-learning.
- The output which is the shortest route between source and destination points will be determined.

3. METHODOLOGY

In this section some important concepts including Reinforcement Learning, Marcov Decision Process, and Q-Learning will describe.

3.1. Reinforcement learning

At present, machine learning (ML) stands as the most influential domain within artificial intelligence (AI). ML is a scientific discipline delving into the investigation and design of algorithms and statistical models that empower machines to learn tasks without explicit programming. This encompasses various learning approaches, including supervised, unsupervised, semi-supervised, and reinforcement learning [6]. The reinforcement learning (RL) paradigm, depicted in Figure 1, comprises two fundamental components: the agent and the environment. The environment encompasses states, with the agent embodying the RL algorithm interacting within this environment [7]. Utilizing a trial-and-error approach, the RL algorithm receives rewards from an interpreter observing the agent's interaction with the environment. This interaction involves three pivotal elements: state, action, and reward. Technical terms and abbreviations are clarified upon their initial usage. The text adheres to a consistent formatting and citation style, ensuring grammatical correctness.

States represent specific arrangements within the environment, actions denote the choices available to agents for modifying the environment, and rewards function as signals defining agent tasks while motivating the selection of one action over others [8].

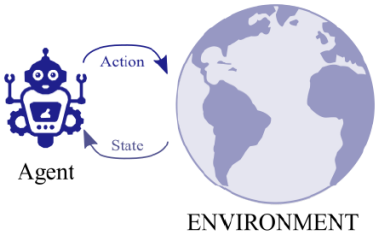

Fig. 1. Reinforcement Learning structure

The iterative learning process involves the agent engaging with the environment by executing actions generated through a policy. This policy comprises stimulus-response rules, mapping each environmental state to a list of actions from which the agent can freely choose. A straightforward implementation of this policy may involve a lookup table. Following action execution, the environment offers feedback in the form of a reward and transitions to the subsequent state. Subsequently, the agent repeats this process, continually refining its actions based on the environment's responses. To develop the necessary skills, the agent explores the policy to maximize the reward [8]. However, relying solely on the reward from a single learning episode is insufficient for determining the optimal policy. Immediate rewards lack insight into the future rewards a specific action may yield, emphasizing the need for a new form of reward, denoted as R_t, which considers long-term gains. It is crucial to highlight that all formulas and equations employed in this method are grounded in references [8, 9-10]. According to [4], formula (1) defines the return value over a finite-length time horizon T, with rt representing the reward the agent receives at time-step t.

$$\mathrm{R_t} = \mathrm{r_{t+1}} + \gamma \mathrm{r_{t+2}} + \gamma^2 \mathrm{r_{t+3}} + \cdots + \gamma^{T-t-1} \mathrm{r_T} = \sum_{i=0}^{T-t-1} \gamma^i \, \mathrm{r_{t+i+1}} \tag{1}$$

Formula (2) is sometimes used to evaluate the return value for time horizons that have no defined endpoint:

$$\mathrm{R_i} = \sum_{i=0}^{\infty} \gamma^i \, \mathrm{r_{i+1}} \tag{2}$$

Where γ is a discounted factor such that 0≤γ<1. To assess the quality of a specific state or state-action pair, two value functions can be defined. Specifically, the state's value function is computed as equation (3) under a policy p.

$$\mathrm{V}\pi(\mathrm{s}) = \mathrm{E}[R|s_t = s, \pi] \tag{3}$$

and the value faction of the state–action pair is calculated as (4):

$$Q_n(s, a) = E[R_t|s_t = s, a_t = a, \pi] \tag{4}$$

A value function can be represented as the connection between two sequential states, s_t and s_{t+1}, determining the Bellman Equations (5) and (6).

$$\mathrm{V_\pi(s)} = \sum_{\mathrm{a}} \pi(\mathrm{s_t}, \mathrm{a}) \sum_{\mathrm{s_{t+1}}} \mathrm{p}(\mathrm{s_{t+1}}|\mathrm{s_t}, \mathrm{a}) \left(\mathrm{W_{s_t \to s_{t+1}|a}} + \gamma \mathrm{V}\pi(\mathrm{s_{t+1}}) \right) \tag{5}$$

$$Q_\pi(s, a) = \sum_{s_{t+1}} p(s_{t+1}|s_t, a) \left(W_{s_t \to s_{t+1}|a} + \gamma \sum\nolimits_{\acute{a}} \pi(s_{t+1}, \acute{a}) Q_\pi(s_{t+1}, \acute{a}) \right) \tag{6}$$

Approximating solutions for these equations often involve dynamic programming, which requires comprehensive information about the problem's dynamics. An alternative and effective approach is to employ model-free reinforcement learning (RL) algorithms [8].

3.2. Marcov Decision Process

In single-agent RL, the environment is typically modeled as a Markov decision process (MDP). Formally, a Markov decision process is defined by a tuple (S, A, P, R, g), where:

- A is the action space
- S is the state space
- P : S × A →δ(S)is the transition probability from state $s \in$ S to $s' \in S$ given the action $a \in A$
- The reward function R:S×A×S→R represents the reward received by the agent when transitioning from the state–action pair (s, a) to the state s'.
- The discount factor, $\gamma \in [0,1]$, is a parameter that compensates for the effect of instantaneous and future rewards [8,10].

3.3. Q-Learning

Q-learning stands as a model-free, value-based approach to reinforcement learning, drawing inspiration from dynamic programming [8]. The primary goal of Q-learning is to formulate a policy that guides the agent in deciding which action to take in each given circumstance [11]. The Q-learning agent is entrusted with the task of deducing the optimal policy, aiming to maximize the total discounted reward [8], as calculated using equation (7).

$$V^*(s) = V^{\pi^*}(s) = max\left\{R_s(a) + \gamma \sum_{\acute{s}} p_{s\acute{s}}[a] V^{\pi^*}(\acute{s})\right\} \tag{7}$$

The estimation of the state-action pair value function (Q-values) involves determining where the average reward, Rs(a), is received by an agent in state s upon selecting action a. The transition probability from state s to $\acute{s}$ given action a is denoted by $p_{ss'}[a]$. To establish estimates of all the value functions at time-step n, a matrix $Qn(s, a)$ with dimensions S×A is created. The Q-values are then improved in each iteration n of the learning process using equation (8).

$$Q_n(s,a) = \begin{cases} (1-\alpha_n)Q_{n-1}(s,a) + \alpha_n[r_n * \gamma V_{n-1}(s_{n+1})] & if\ s = s_n\ and\ a = a_n \\ Q_{n-a}(s,a) & otherwise \end{cases} \tag{8}$$

Where α_n is the learning rate, r_n is the instant reward received, also

$$V_{n-1}(s_{n+1}) = \max\left(Q_{n-1}(s_{n+1}, \acute{a})\right) \tag{9}$$

Under the assumption of bounded rewards $|r_n| \leq R$ and learning rate $0 \leq \alpha_n < 1$ such that

$$\sum_{i=1}^{\infty} \alpha_{n^i(s,a)} = \infty, \sum_{i=1}^{\infty} \left[\alpha_{n^i(s,a)}\right]^2 < \infty, \forall s, a \tag{10}$$

The estimates $Q_n(s, a)$ will converge to the optimal Q-value $Q^*(s, a)$ with probability 1.

4. SIMULATION AND DISCUSSION OF RESULTS

To determine the shortest path between source and destination points, this paper first imports a transportation map matrix as an environment for the algorithm. In this case, a 5*5 matrix is used. Each edge in the matrix represents a pathway between two points. Following this, the user selects both source and destination points. Fig. 2 depicts the 5*5 matrix as the environment with the chosen source and destination points. Also, there are three areas marked as forbidden due to traffic congestion which the agent must avoid.

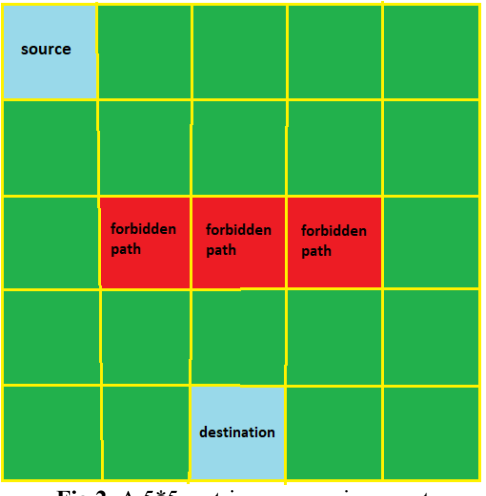

Fig 2. A 5*5 matrix as an environment

Then, the Q-learning algorithm is implemented in this environment. Algorithm 1 contains the pseudo code of the main algorithm, which has been applied in the environment shown in Figure 2. Algorithm 1 outlines how Q-learning discovers the optimal route. The agent initiates movement by taking an action, choosing from four options at each position: up, down, left, or right. The optimal choice is determined either randomly or by selecting the action with the highest Q-table value based on the agent's position. The Q-table is initially populated with random values but is updated as the agent takes actions. In reinforcement learning and Q-learning, forbidden paths result in a maximum penalty of -100, and reaching the destination yields a reward of +100. The agent receives penalties based on its distance from the destination, with greater distances incurring greater penalties. The path with the highest rewards is considered the shortest route from source to destination points. Table 1 displays crucial parameter values for Algorithm 1.

Figure 3 illustrates the rewards received by the agent during each learning episode in the environment (Figure 2). Initially, incorrect choices led to negative rewards, but with trial and error, the agent improved its choices, resulting in higher rewards. The maximum reward indicates convergence, and Table 2 shows the average reward every 50 episodes, demonstrating convergence over time. After 500 episodes, the algorithm achieves the maximum reward.

Algorithm 1

```
MATRIX_SIZE = 5
NNUMBER_OF_ACTION = 4
MAX_PENALTY = -100
MIN_PENALTY = -1
MAX_REWARD = 100
initialize q_table [agent's previous position, action] with
random samples from uniform distribution between (-5,0)
for episode< Episode:
   local_reward = 0
   for i < STEPS:
      trajectory[i] = agent's position (x,y)
      tradeoff = random float in the half-open interval [0.0, 1.0)
      if tradeoff > epsilon:
         action = according to agent positin select the action
                  which has max value in q-table
      else:
         action = random integer between 1 and 4
         #1: Up, 2: Down, 3: Left, 4: Right
      move the agent according to the selected action
      calculate reward according to agent's new position
      calculate max_future_q according to max value
      of q-table[agnet's position, all actions]
       if reward == MAX_REWARD:
         new_q = reward
      else:
         current_q = q_table[agent's previous position, action]
         temp1=(1 - LEARNING_RATE) * current_q
         temp2= LEARNING_RATE * (reward + DISCOUNT * max_future_q)
         new_q = temp1+ temp2
      q_table[agent's new position, action] = new_q
      local_reward += reward
      if reward == MAX_REWARD or reward == MAX_PENALTY:
         exit for loop
   end
   epsilon *= EPS_DECAY
   best_reward = local_reward
   best_trajectory = trajectory
end
```

Table 1. the parameters of Q-Learning algorithm

LEARNING_RATE	DISCOUNT(DISCOUNT FACTOR)	EPS_DECAY(EPSILON DECAY)	EPSILON
1.03	0.237	0.0723	0.584

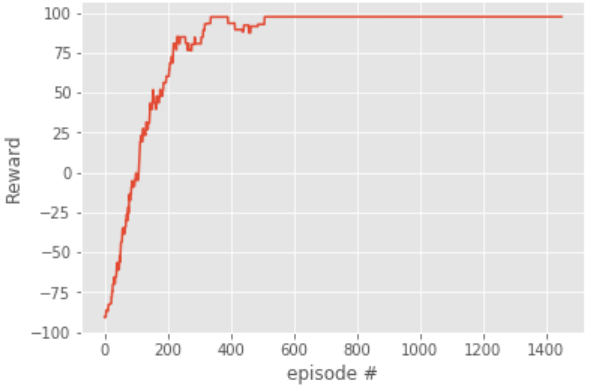

Fig 3. the received reward by the agent per episode of learning process

Table 2. the mean of received reward every 50 episodes

# episode	the mean of received reward
1	-90.63
50	-52.15
100	-0.62
150	39.56
200	60.25
250	84.90
300	80.73
350	97.38
400	93.46
450	92.22
500	97.52
550	97.52
600	97.52
650	97.52
700	97.52

5. CONCOLUTION

In this groundbreaking paper, we introduce a cutting-edge model that leverages Q-learning to ascertain the shortest route between two given points. Q-learning, a model-free, value-based reinforcement learning technique, engages in a trial and error process, garnering rewards or penalties for each action taken. The proposed methodology implements this model across a spectrum of scenarios encompassing distinct starting and ending points. Through comprehensive simulations, our findings underscore a remarkable convergence rate in pinpointing optimal routes. This underscores the effectiveness of the Q-learning model in effectively addressing the shortest route problem across diverse and complex scenarios, positioning it as a versatile and powerful solution in route optimization.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] Othman, B., De Nunzio, G., Di Domenico, D., & Canudas-de-Wit, C. (2019). Ecological traffic management: A review of the modeling and control strategies for improving environmental sustainability of road transportation. Annual Reviews in Control, 48, 292–311. doi:10.1016/j.arcontrol.2019.09.003

[2] Guo, D., Wang, J., Zhao, J. B., Sun, F., Gao, S., Li, C. D., … Li, C. C. (2019). A vehicle path planning method based on a dynamic traffic network that considers fuel consumption and emissions. The Science of the Total Environment, 663, 935–943. doi:10.1016/j.scitotenv.2019.01.222

[3] Change, Intergovernmental Panel on Climate. (2023). Climate Change 2021 – The Physical Science Basis. http://doi.org/10.1017/9781009157896

[4] Masson-Delmotte, V.., Zhai, Panmao., Pörtner, H.., Roberts, Debra., Connors, Sarah., Diemen, R. V.., Ferrat, M.., Haughey, Eamon., Neogi, S.., Pathak, Minal., Petzold, Jan., Vyas, Purvi., Huntley, Elizabeth., Kissick, Katie., Belkacemi, Malek., & Malley, Juliette. (2022). Climate Change and Land. . http://doi.org/10.1017/9781009157988

[5] Tang, Cheng., Zheng, Yao., Jaroniec, M.., & Qiao, S.. (2021). Electrocatalytic Refinery for Sustainable Production of Fuels and Chemicals.. Angewandte Chemie . http://doi.org/10.1002/anie.202101522.

[6] Nian, R., Liu, J., & Huang, B. (2020). A review On reinforcement learning: Introduction and applications in industrial process control. Computers & Chemical Engineering, 139(106886), 106886. doi:10.1016/j.compchemeng.2020.106886

[7] Lema, G. G., Weldemichael, K. S., & Weldemariam, L. E. (2021). Performance evaluation of cooperative mobile communication security using reinforcement learning. Heliyon, 7(5), e07108. doi:10.1016/j.heliyon.2021.e07108

[8] Canese, L., Cardarilli, G. C., Di Nunzio, L., Fazzolari, R., Giardino, D., Re, M., & Spanò, S. (2021). Multi-agent reinforcement learning: A review of challenges and applications. Applied Sciences (Basel, Switzerland), 11(11), 4948. doi:10.3390/app11114948

[9] Guo, J., & Harmati, I. (2020). Evaluating semi-cooperative Nash/Stackelberg Q-learning for traffic routes plan in a single intersection. Control Engineering Practice, 102(104525), 104525. doi:10.1016/j.conengprac.2020.104525

[10] Nian, R., Liu, J., & Huang, B. (2020). A review On reinforcement learning: Introduction and applications in industrial process control. Computers & Chemical Engineering, 139(106886), 106886. doi:10.1016/j.compchemeng.2020.106886

[11] Hossain, M. S., Nwakanma, C. I., Lee, J. M., & Kim, D.-S. (2020). Edge computational task offloading scheme using reinforcement learning for IIoT scenario. ICT Express, 6(4), 291–299. doi:10.1016/j.icte.2020.06.002