2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI) | 978-1-7281-9675-6/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICDABI51230.2020.9325657

Reinforcement Learning with the Classical Q-Learning Algorithm for Optimizing Single Intersection Performance

1st M. Rosyidi, 2nd Sahid Bismantoko, 3rd Tri Widodo
Center Of Technology For System And Infrastructure Of Transportation
Agency For The Assessment And Application Of Technology
Tangerang Selatan, Indonesia
(m.rosyidi, sahid.bismantoko, tri.widodo)@bppt.go.id

***Abstract*—Many factors causing the issue of the land transportation like intersections problems . In cities where the volume of vehicles passing through an intersection is so large that sometimes a vehicle can pass through an intersection after two or three red lights. The length of the queue at each leg of the intersection can cause the average delay time to be large. In this research, optimizing the performance of a single intersection is an attempt to minimize the average delay time at a single intersection using the classical Q-Learning algorithm. The data used in this study were obtained from the local government of Pekalongan City. The results of this study indicate that minimizing the average delay time will also affect the traffic light time cycle setting, which causes the queue length at the intersection to decrease as well, in this study the parameter value used for the learning rate is 0.5, and the average delay time decreased by 2.96 seconds.**

***Index Terms*—Q-Learning, Single Intersection, Optimizing, Delay Time**

I. Introduction

In the transportation areas, the intersection is one of the problems with potential cause of congestion, this is due to the large volume of vehicles and the inappropriate traffic light cycle configuration, which can cause queues at each leg of the intersection. The queue length at each leg of the intersection can cause congestion at the intersection, because during rush hour when a vehicle can pass through the intersection after two or three red lights, this will be a problem in transportation management. The current condition for determining the traffic light cycle configuration is done manually, so that if there is an imbalance in the queue at the intersection, changes can not be made quickly to the traffic light cycle configuration, therefore making the optimum traffic light cycle configuration is very helpful for maintaining balance. of the vehicle flow at each leg of the intersection.

Currently, there is a paper that has calculated the time delay by comparing several existing methods [1]. In this research, one of the methods used to estimate the time delay is by using the classic Q-learning algorithm. To facilitate implementation, a case study is used at Sorogenen a single intersection in the city of Pekalongan, Central Java, Indonesia. Sorogenen intersection is an intersection with quite heavy traffic flow in the morning and evening. Data related to traffic at the Sorogenen intersection were obtained from the local government of Pekalongan City, Central Java, Indonesia. The Sorogenen single intersection condition is a suitable object for the implementation of the classic Q Learning algorithm because the required data is complete, making it easier for the implementation of the classic Q-Learning algorithm which aims to minimize the average delay time which has an impact on reducing queue lengths and changes in traffic light cycle configuration. This research aims to optimize the performance of the Sorogenen single intersection, namely by minimizing the average delay time using Reinforcement Learning with the Q-Learning algorithm.

II. Related Work

Research related to optimization of transportation problems especially at intersections has been done a lot, including optimizing the traffic light cycle configuration with the aim of minimizing the average delay time to reduce traffic congestion by using modified Q-Learning and other methods for comparison [1], [2]. These research explain generally about some exploration technique in the Q-Learning approach, but in this research focus using ϵ-greedy approach with comparison with data real from Sorogenen, Pekalongan City. The problem of traffic congestion has also been highlighted in several studies such as the one carried out by controlling traffic signals on transportation networks using reinforcement learning [3]–[5]. To control traffic flow at intersections, a research on the development of traffic signal control at intersections has been carried out to evaluate the density of traffic flow [6], [7], and get the optimization value for delay time [8]. These two research giving ideas how to reduce or solve the problems of intersection congestion.

Research related to traffic signal control algorithms using reinforcement learning for future vehicles (autonomous) where vehicles are related to one another [9]. The idea of Yang et al explain the Q-Learning usage in industrial to solve signal control and strong reference to this research to learn deeper about Q-Learning on transportation area, especially intersection problems.

978-1-7281-9675-6/20/$31.00 ©2020 IEEE

This research provides an overview of the classic Q-Learning algorithm method for the implementation of performance optimization at a single intersection with promising results. This study will improve intersection system from the manual system with a more efficient intersections.

Q-learning learn from previous experiences or state ; focus in traffic signal delay time. Learning from past action will assist the algorithm to make better result and decisions for the future action to make adaption into the dynamic change of traffic signal delay time, especially in Pekalongan city.

In this study, especially the case of single intersection using the classical Q-Learning approach has the advantage of being simpler and of speed in getting the minimum value of the average delay time.

III. Reinforcement Learning With Q-Learning

Reinforcement Learning is a Machine Learning paradigm, by which a controller's policy can be optimized through trial-and-error interactions with an environment. Reinforcement learning making optimal decision by learn from past experiences. The process of reinforcement learning consist :

1) Observation of the environment
2) Deciding how to act using some strategy
3) Acting accordingly
4) Receiving a reward or penalty
5) Learning from the experiences and refining our strategy
6) Iterate until an optimal strategy is found

There are 2 main types of Reinforcement Learning algorithms. They are model-based and model-free. Algorithm with model free-based, estimate the optimal policy without transition and reward function of the environment. Model-based algorithm reconsider the transition and reward function to estimate optimal policy.

A schematic illustration is shown in Figure 1. By using a predefined reward function from the environment as feedback, the controller can learn which policies are effective and ideally converge to the most optimal policy .

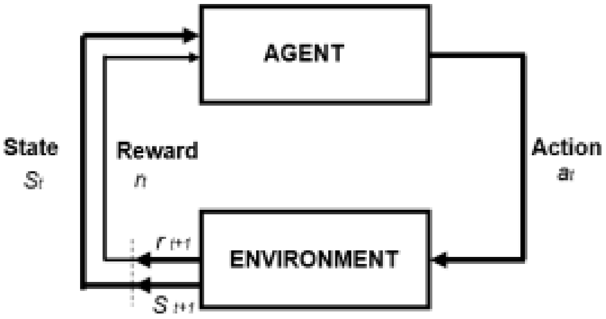

Fig. 1. Agent environment interaction in Reinforcement Learning

A. Q Learning

One of the most basic and popular methods to estimate Q-value functions in a model free fashion is the Q-learning algorithm by Watkins (1989); Watkins and Dayan (1992). The basic idea in Q-learning is to incrementally estimate Q-values for actions, based on feedback (i.e. rewards) and the agent's Q-value function [10].

Q-learning is one of the off-policy of reinforcement learning because its learn from actions outside of the current policy. When q-learning is performed, it's create q-table or matrix that follows the shape of [state, action] and initialize values to zero. After that update and store q-values after an episode. This q-table becomes a reference table for our agent to select the best action based on the q-value.

The update rule use Q-values and a built-in max-operator over the Q-values of the next state in order to update Q_t into Q_{t+1} :

$$Q_{k+1}(S_t, a_t) = Q_k(s_t, a_t) + \alpha * (r_t + \gamma * max Q_k(s_t + 1, a_t) - Q_k(s_t, a_t)) \quad (1)$$

The agent makes a step in the environment from state s_t to $s_t + 1$ using action at while receiving reward r_t. The update takes place on the Q-value of action at in the state s_t from which this action was executed. The following Q-learning step show in algorithm 1 [10]

Algorithm 1 Q-Learning Algorithm

```
Require: discount factor $\gamma$, learning parameter $\alpha$ initialize Q
    arbitrarily (e.g. Q(s, a) = 0, $\forall$s $\in$ S, $\forall$a $\in$ A )
for Each Episode do
    s is initialized as the starting state
    repeat
        choose an action a $in$ A(s) based on an exploration
        strategy
        perform action a
        observe the new state s'
        received reward r
        Q(s, a) := Q(s, a) + $\alpha$ (r + $\gamma$ maxQ(s', a') - Q(s, a))
        s := s'
    until s' is a goal state
end for
```

B. Exploration and Exploitation

An agent interacts with the environment in 1 of 2 ways. The first is to use the q-table as a reference and view all possible actions for a given state. The agent perform the action based on the max value on the current iteration. This is known as exploiting since we use the information we have available to us to make a decision.

The second way to take action is to act randomly. This is called exploring. The main idea is selected an action at random in this process. Acting randomly is important because it allows the agent to explore and discover new states that otherwise may not be selected during the exploitation process. You can balance exploration/exploitation using epsilon ($epsilon$) and setting the value of how often you want to explore vs exploit.

TABLE I
SINGLE INTERSECTION SOROGENEN DATA

No	Description	Time
1	Green Light North	18 Seconds
2	Green Light West	31 Seconds
3	Average Delay Time	16.87 Seconds

Here's some rough code that will depend on how the state and action space are setup.

In this research, exploration and exploitation to determine the next action using ϵ–greedy, at the initial stage of the exploration process in determining the direction of the action carried out randomly, the value of the parameter ϵ is between 0< ϵ <1, this exploration is carried out while maintaining balance with exploitation, then the exploitation process is carried out where previous learning has got experience so that the previously obtained values can become experience for taking the next action. The description of exploration and exploitation using ϵ–greedy is as shown in Figure 2.

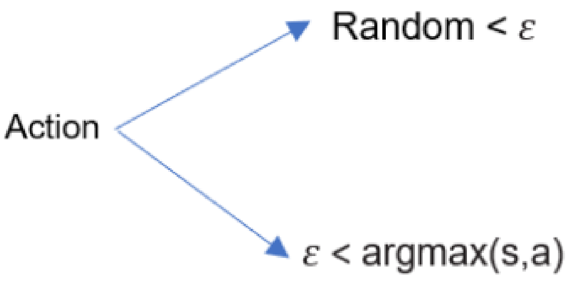

Fig. 2. ϵ-greedy

C. Single Intersection

The single intersection in this research is one of the intersections used as a case study. Single intersection "Sorogenen" is a simple intersection in which only two traffic light cycle configurations are used in this research, namely north to south and east to west like Figure 3, there are only two traffic light cycle configuration settings, from north to south or vice versa and from west to east or vice versa. The current condition of the intersection shows that in the morning and evening there is an increase in vehicle volume in both directions and the traffic light cycle configuration settings are done manually so that an imbalance in the queue of vehicles at the intersection can occur.

Compile data at Sorogenen Intersection such as table I.

Table 1 shows the average value of the delay time and the setting value of the traffic light time cycle where the data is obtained based on traffic flow observations and measurement of road infrastructure dimensions. Observation and measurement data are then performed manually calculations to obtain the average value of delay time and traffic light time cycle. This process will be very troublesome if there is a change in traffic flow, so it is necessary to observe the traffic flow again at the intersection to make adjustments to the traffic light time cycle settings. To speed up the calculation of the minimum value of the average time delay and determine the appropriate traffic light time cycle, this study was conducted using a Q-Learning approach.

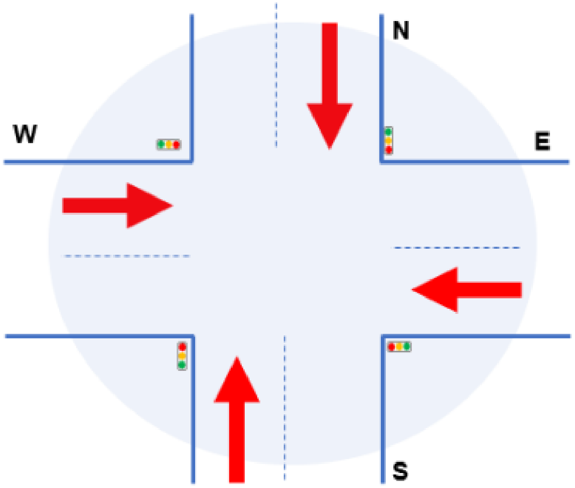

Fig. 3. Sorogenen single intersection

IV. IMPLEMENTATION

Implementation of RL at a single intersection, it is necessary to define the terms in a single intersection with the RL, Q is the Average Delay Time, the state is the traffic light time setting. Action shows that in one condition there are two actions that occur at one time, so it is defined as follows [1]:

1) Action 1
   - The Traffic Light North Time setting is updated
   - Traffic Light West Time setting +3
2) Action 2
   - The Traffic Light North Time setting is updated
   - Traffic Light West Time setting -3
3) Action 3
   - Traffic Light North Time setting + 3
   - The Traffic Light West Time setting is updated
4) Action 4
   - Traffic Light North Time setting - 3
   - The Traffic Light West Time setting is updated

In this research, based on the test results, the rewards used are as follows :

$$Reward(r) = Q_{best} - Qold \tag{2}$$

V. RESULT

Table II shows the results of the average value of delay time and the setting value of the traffic light time cycle in manual calculations (existing conditions) compared to calculations using RL with the Q-Learning approach. The calculation process manually is carried out based on traffic flow observation data and road dimension measurement data. Then from these data, the calculation of the average value of the delay time and the setting value of the traffic light time cycle is carried out. In manual calculations it takes a long time to complete. This will become a problem when there is a change in data traffic flow so it is necessary to recalculate it to get the average delay time and the traffic light time cycle setting values. RL with the Q Learning approach gave promising results, using the python program and the initial setting for Learning Rate (α) = 0.5 and Discount rate (γ) = 0.01, the results obtained showed an average delay time value of 13.91 seconds while the setting value of the traffic light time cycle (Green Light North) is 12 seconds and the traffic light time cycle (Green Light West) is 18 seconds.

TABLE II
COMPARISON RESULT OF EXISTING COND AND Q-LEARNING

No	Description	Exsisting Condition (time)	Q-Learning (time)
1	Green Light North	18 Seconds	12 Seconds
2	Green Light West	31 Seconds	18 Seconds
3	Average Delay Time	16.87 Seconds	13.91 Seconds

The results of calculations with the Q Learning approach when compared with manual calculations, the average value of delay time in the existing conditions decreased by 2.96 seconds with a traffic light time cycle (Green Light North) setting value of 12 seconds and a traffic light time cycle (Green Light West) of 18 seconds. The significant decrease in the average value of delay time will have an impact on reducing the queue length at the intersection. In the case of a single intersection, the calculation of the average value of delay time with the Q Learning approach helps to make it easier to get the optimal value of the average delay time.

Figure 4 shows that the change in the Q value does not change after the episode 100. Thus, after more than 10 iteration, the matrix value for every action are become converge

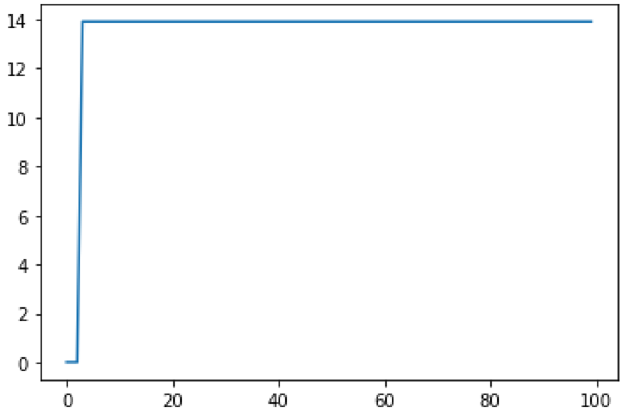

Fig. 4. Episode vs Q

Figure 5 show all of the converge of action matrix after 10000 iteration, it scale between 13.5 to 14.5 to take the plot little closer, so the plot dynamically can be represent. Figure 6 is blue line graph from figure 5. The result of all action Q-value converge with differences each cell near zero value.

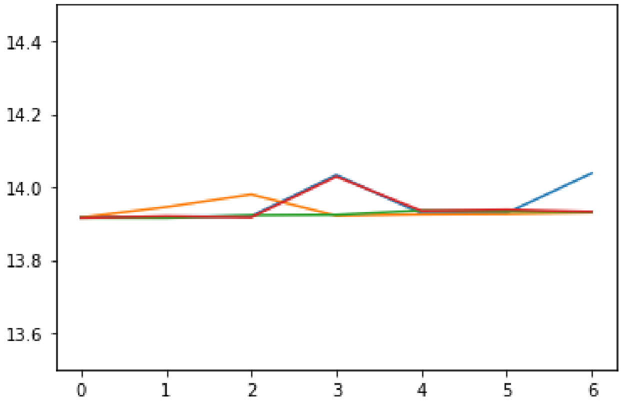

Fig. 5. All Action Converge Value

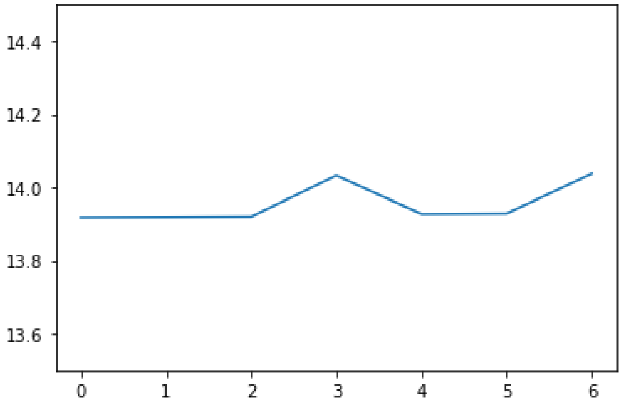

Fig. 6. Blue Line Action Converge Value

VI. CONCLUSION AND FUTURE WORK

A. Conclusion

1) The results of this study indicate that minimizing the average delay time will also affect the traffic light time cycle setting, which causes the queue length at the intersection to decrease as well, in this study the parameter value used for the learning rate is 0.5, while the average delay time value at The existing condition is 16.87 seconds and becomes 13.91 seconds using the Q-Learning approach so that it has a reduction of 2.96 seconds.
2) There needs to be improvements in Q-learning approach, especially in relation to determining the initial state value using discrete values. The use of discrete values considers the calculation process to be shorter, to improve it the action taken needs to be adjusted.
3) In the case of single intersection using the classical Q-Learning approach has the advantage of being simpler

and of speed in getting the minimum value of the average delay time.

4) Reinforcement Learning with the Q-Learning algorithm gives promising results to solve single intersection optimization.

B. Future Work

It is necessary to make comparisons using different exploration and exploitation methods and it is necessary to carry out testing for mutli intersections in an integrated manner.

Acknowledgment

Thanks to PTSPT, BPPT that fully support this research related to the technical support. Gratitude's to the Local Government Kota Pekalongan that provided research environment and intersection data.

References

[1] H. C. Chu, Y. X. Liao, L. H. Chang, and Y. H. Lee, "Traffic light cycle configuration of single intersection based on modified Q-Learning," *Applied Sciences (Switzerland)*, vol. 9, no. 21, 2019.

[2] A. D. Tijsma, M. M. Drugan, and M. A. Wiering, "Comparing exploration strategies for Q-learning in random stochastic mazes," *2016 IEEE Symposium Series on Computational Intelligence, SSCI 2016*, no. February 2018, 2017.

[3] J. Lee, J. Chung, and K. Sohn, "Reinforcement Learning for Joint Control of Traffic Signals in a Transportation Network," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 1375–1387, 2020.

[4] E. Walraven, "Traffic Flow Optimization using Reinforcement Learning," *Belgian/Netherlands Artificial Intelligence Conference*, pp. 215–216, 2014.

[5] D. Johansson and J. V. O. N. Hacht, "Reinforcement Learning Applied to Select Traffic Scheduling Method in Intersections," 2019.

[6] B. Ghazal, K. Elkhatib, K. Chahine, and M. Kherfan, "Smart traffic light control system," *2016 3rd International Conference on Electrical, Electronics, Computer Engineering and their Applications, EECEA 2016*, no. April, pp. 140–145, 2016.

[7] N. S. Jadhao and A. S. Jadhao, "Traffic signal control using reinforcement learning," in *2014 Fourth International Conference on Communication Systems and Network Technologies*, 2014, pp. 1130–1135.

[8] R. Marsetič, D. Šemrov, and M. Žura, "Road Artery Traffic Light Optimization with Use of the Reinforcement Learning," *PROMET - Traffic&Transportation*, vol. 26, no. 2, pp. 101–108, 2014.

[9] M. Yang, Kaidi; Tan, Isabelle; Menendez, "Research CollectionA reinforcement learning based traffic signal control algorithm in a connected vehicle environment," 2017.

[10] M. V. Otterlo, "Reinforcement Learning and Markov Decision Processes A Few Pointers to the Field Function Approximation , Generalization and Abstraction," no. May, pp. 1–4, 2009.