https://doi.org/10.3311/PPtr.15923
Creative Commons Attribution ⓘ

Periodica Polytechnica Transportation Engineering, 48(4), pp. 313–319, 2020

Comparison of Game Theoretical Strategy and Reinforcement Learning in Traffic Light Control

Jian Guo[1*], István Harmati[1]

[1] Department of Control Engineering and Information Technology, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, H-1521 Budapest, P. O. B. 91, Hungary
[*] Corresponding author, e-mail: guojian@iit.bme.hu

Received: 12 March 2020, Accepted: 12 March 2020, Published online: 08 June 2020

Abstract

Many traffic models and control methods have already been utilized in the public transportation system due to the increasing traffic congestion. Thus, an intelligent traffic model is formalized and presented to control multiple traffic light simultaneously and efficiently according to the distribution of vehicles from each incoming link (i.e. sections) in this paper. Compared with constant strategy, two methods are proposed for traffic light control, i.e., game theoretical strategy and reinforcement learning methods. Game theoretical strategy is generated in a game theoretical framework where incoming links are regarded as players and the combination of the status of traffic lights can be regarded as decisions made by these players. The cost function is evaluated and the strategy is produced with Nash equilibrium for passing maximum vehicles in an intersection. The other one is Single-Agent Reinforcement Learning (SARL), specifically with the *Q*-learning algorithm in this case, which is usually used in such a dynamic environment to control traffic flow so the traffic problem could be improved. Generally, the intersection is regarded as the centralized agent and controlling signal status is considered as the actions of the agent. The performance of these two methods is compared after simulated and implemented in a junction.

Keywords

traffic light control, game theory, reinforcement learning

1 Introduction

Traffic lights at the intersections are typically controlled by a constant strategy in the real world, i.e., the time intervals of green or red lights are fixed and periodical, which would aggravate the traffic congestion while the traffic flow is distributed on different incoming links uniformly. A lower cost and more efficient control approach can be developed and achieved instead of building expensive infrastructure in an intersection.

There are plenty of novel models and approaches to controlling traffic light nowadays, which are presented and published in some research papers. E.g., a fuzzy control method to public transportation is designed for the problem, the core of these methods is the fuzzy rule set which depends on traffic situations (Hoyer and Jumar, 1994). Another traffic control system based on Bayesian probability is developed to improve poor traffic management, which is adaptive to the high dynamics of the traffic network (Khamis et al., 2012). Based on the previous historical information, the traffic flow can be analyzed and predicted by a novel traffic management system (Yousef et al., 2019). Besides, it is quite common and popular to integrate some bionic techniques into the traffic model or control methods, such as genetic algorithm (Gora, 2011; Teo et al., 2010; Turky et al., 2009), ant colony technology (D'Acierno et al., 2012; He and Hou, 2012; Jabbarpour et al., 2015) and arteries models (Zhang and Jia, 2011). In game theoretical approach, the traffic management system is constructed as a gameplay problem where incoming links are commonly treated as game players and the optimal decisions would be made based on the scale of vehicular flow in one intersection (Alvarez Villalobos et al., 2008; Guo and Harmati, 2019). This method can also be implemented efficiently for multiple intersections (Bui and Jung, 2018; Fan et al., 2014). As machine learning technics grows, Reinforcement Learning (RL) is an excellent way to improve the efficiency in such a dynamic environment of traffic control problem, where the agents representing the incoming links take optimal actions to make sure they gain the maximum reward no matter it is a Single-Agent Reinforcement Learning (SARL) (El-Tantawy and Abdulhai, 2010) or Multi-Agent

Cite this article as: Guo, J., Harmati, I. (2020) "Comparison of Game Theoretical Strategy and Reinforcement Learning in Traffic Light Control", Periodica Polytechnica Transportation Engineering, 48(4), pp. 313–319. https://doi.org/10.3311/PPtr.15923

Reinforcement Learning (MARL) (Bakker et al., 2010; Wiering, 2000). Neural network algorithm can also be combined into RL to form an extension and new approach as Deep Reinforcement Learning (DRL), which is capable of estimating *Q*-value more efficiently in some cases (Liang et al., 2019; van der Pol and Oliehoek, 2016).

The goal of this research is to find a more efficient and effective method to control traffic light so that the maximum number of vehicles could pass the intersection in specific cycle time. A constant strategy is introduced firstly, i.e., the time intervals of green or red lights are fixed and periodical. Then game theoretical strategy with Nash equilibrium is proposed to optimize the traffic flow. The players represent the incoming links and make decisions based on the cost function. The third method – SARL is also presented and implemented in the experiment for the comparison of the previous two approaches. In this method, the centralized agent representing the intersection observes the state of the environment, get the cumulative reward, and take actions optimally to make the vehicles pass the intersection as many as possible. The final result indicates that the game theoretical strategy and RL method can both improve the efficiency of traffic management compared with the constant strategy.

2 Traffic model formulation

In Fig. 1, four incoming links with four individual directional paths are in this common intersection. Each directional path has traffic lights individually, which is green and red in Eq. (1). Red light is represented by 0, and green is 1, yellow light is not considered to simplify the control process. Fig. 1 and Eq. (1) are shown as follows:

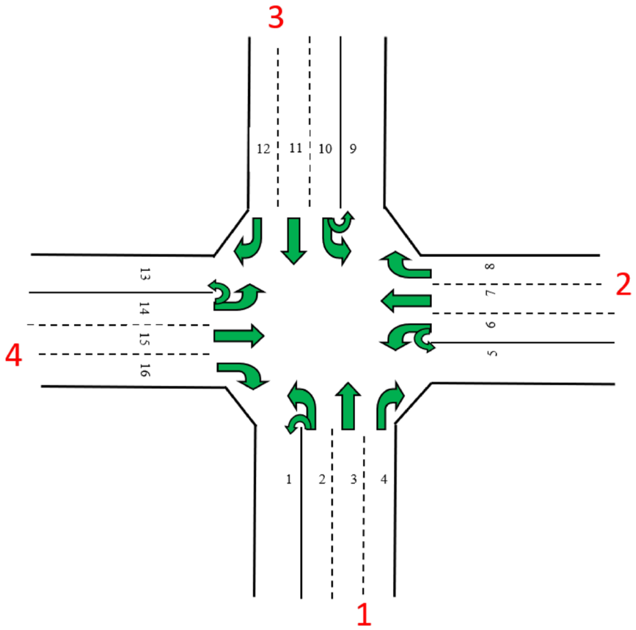

Fig. 1 General structure of a single intersection

$$g = \begin{cases} 0 & \text{red} \\ 1 & \text{green} \end{cases}. \tag{1}$$

The initial incoming link where the vehicles departure is defined as *w*, *w* = 1,…,4 and the target incoming link where the vehicles arrive is defined as *z*, *z* = 1,…,4. Based on that, the moving direction of the vehicle flow can be represented as $w - z$, i.e., the vehicles are going to the target incoming link *z* from incoming link *w*. Waiting vehicles at the traffic light is defined as queue length with a notation L_{wz}, and incoming stream S_w is the incoming vehicles outside the queues with each second. Turning rate t_{wz} determines how much ratio of the vehicles will be selected from the incoming stream for going different target incoming links. The speed of the leaving traffic flow is $F_{L,wz}$, and t_e is defined to remove all the waiting vehicles in the queue and incoming vehicles from outside the queues.

Thus, Eq. (2) can be created with t_e, i.e., the incoming vehicles and waiting vehicles should be equal to the leaving vehicles in t_e at the *k* time slice (Eq. (2)):

$$L_{wz}(k) + S_w t_{wz} t_e(k) = F_{L,wz} t_e(k). \tag{2}$$

Solving t_e from Eq. (2) (Eq. (3)):

$$t_e(k) = \frac{L_{wz}(k)}{F_{L,wz} - S_w t_{wz}}. \tag{3}$$

Some cases for updating queues can be discussed based on Eq. (3).

Case 1: $0 < t_e(k) < T_s$. Where time slice T_s is a part of cycle time T_c and g_{wz} can represent the status of a traffic light as in Eq. (1). All the waiting vehicles on the path will be removed in the period $T_s - t_e(k)$ and the incoming vehicles can also be removed in the , if the traffic light is green ($g_{wz} = 1$). Otherwise, any vehicles cannot leave the intersection if the traffic light is red ($g_{wz} = 0$). The updated queue length $L_{wz}(k + 1)$ can be expressed (Eq. (4)):

$$\begin{aligned} L_{wz}(k+1) &= L_{wz}(k) - g_{wz}(k) F_{L,wz} t_e(k) \\ &- g_{wz}(k) S_w t_{wz}\left(T_s - t_e(k)\right) + S_w t_{wz} T_s. \end{aligned} \tag{4}$$

Case 2: $t_e(k) \le 0$ or $t_e(k) \ge T_s$. Removing all the waiting vehicles and incoming vehicles in the period T_s is nor possible even while the traffic light is green. Thus, it is simpler to consider the updated queues in the period T_s (Eq. (5)):

$$L_{wz}(k+1) = L_{wz}(k) - g_{wz}(k) F_{L,wz} T_s + S_w t_{wz} T_s. \tag{5}$$

Case 3: $t_e(k) = 0$. The initial queue length $L_{wz}(k) = 0$ results in $t_e(k) = 0$ from Eq. (3). This situation mainly depends on whether the speed of leaving vehicle stream $F_{L,wz}$ is larger or not, compared with the speed of incoming vehicle stream $S_w t_{wz}$. The difference between them can be defined as FS_{wz}. If the sign of FS_{wz} is positive, then the queue length will keep empty. However, if it is negative, then the queue length will be increased even though a small part of the vehicle stream is leaving the intersection with the speed $F_{L,wz}$. Otherwise, if it is 0, then the queue will remain 0 during this period. The queues can be updated in period T_s (Eq. (6)):

$$L_{wz}(k+1) = \begin{cases} -g_{wz}(k) S_w t_{wz} T_s + S_w t_{wz} T_s, & FS_{wz} > 0 \\ L_{wz}(k) = 0, & FS_{wz} = 0. \\ -g_{wz}(k) F_{L,wz} T_s + S_w t_{wz} T_s, & FS_{wz} < 0 \end{cases} \quad (6)$$

3 Traffic control methods

3.1 Constant strategy

As aforementioned before, the constant strategy is that the time interval of green light or red light is fixed and periodical for the incoming links. In general, the simple traffic light controllers are what are known as electro-mechanical signal controllers, which are utilized worldwide in reality nowadays, unlike computerized signal controllers. These controllers use dial timers that have fixed, signalized intersection time plans. The plans can be scheduled according to the scale of the traffic flow through the intersection in history. This traffic light can only store only a one-time plan while it is working, which is not so efficient to control the traffic flow when the scale of traffic flow changes on some occasions such as holidays, accidents, and bad weather. In this case, the specific actions for the signal controllers are generated, and it will be explained in later research.

3.2 Game theoretical strategy

In game theory, the traffic management problem is constructed as a gameplay problem where incoming links are regarded as the players. As is known that there are 4 players in this case from Fig. 1, and the decisions of these players represent the status of traffic control light (red and green) for each directional path. The notation indicates the status of a traffic light for the directional path. Thus, the decision vector for each player can be combined by the status of traffic light of each path belonging to its incoming link (Eq. (7)):

$$d_w = \begin{pmatrix} g_{w1} & g_{w2} & g_{w3} & g_{w4} \end{pmatrix}, \quad (7)$$

where d_w is a decision vector with 16 possible values since it combines 4 bits binary code g_{wz}.

The cost function can also be defined for the goal of passing maximum vehicles, the same as minimum queues, which corresponds to the smallest cost value in the cost function. However, it has the greatest cost value if the traffic light is red since the corresponding path will gather maximum vehicles without any passing vehicles. Thus, it can be described as (Eq. (8)):

$$J_{wz}(g_{wz}) = \begin{cases} \dfrac{L_{wz}}{C_{wz}} & g_{wz} = 1 \\ J_{\max} & g_{wz} = 0 \end{cases}, \quad (8)$$

where C_{wz} is the capacities defined as the largest queue length and $J_{\max}$ is a great constant which is far larger than general cost value.

The cost function of each player is derived from Eqs. (7) and (8) (Eq. (9)):

$$J_w(d_w) = \sum_z J_{wz}. \quad (9)$$

A rational optimal strategy called Nash equilibrium in game theory can be generated to balance the cost of each player. The levels of the players are the same in this non-cooperative game, and the players cannot improve the interest any more with other decisions once the Nash equilibrium is reached among the players (Başar and Olsder, 1998). The Nash equilibrium solution can be got (Eq. (10)):

$$\left(d_1^*, d_2^*, d_3^*, d_4^*\right) = \arg\min_{d_i} J_w\left(d_1, d_2, d_3, d_4\right). \quad (10)$$

In such a high quantity of decision-combination, it is quite common to get not only one solution of Nash equilibrium. However, only one Nash equilibrium solution can be selected for calculating, and the minimum average value of four players should be chosen (Eq. (11)):

$$\left(d_1^*, d_2^*, d_3^*, d_4^*\right) = \arg\min_{d_i} \sum_{i=1}^{4} J_i / 4. \quad (11)$$

3.3 Reinforcement Learning

Reinforcement Learning (RL) is a learning method in which agents learn a policy $\pi(s) = a$ that mapping to an action from the current state of environment s. The agents have to find an optimal policy π^* and the corresponding action will be taken to maximize the cumulative reward $r(s, a)$. The SARL with Q-learning algorithm is implemented in such a dynamic environment since it is an online method, and the optimal actions are updated in a repeated process. The Q-function is the core of Q-learning, which reflects the relation between state and action. Due to the uncertainty of its parameters in a dynamic environment, the Q-function can start with an arbitrary Q_0 and be updated at iteration step t as follows (Eq. (12)):

$$Q_{t+1}\left(s_t, a_t\right) = (1-\alpha) Q_t\left(s_t, a_t\right) \\ +\alpha\left[r_t + \gamma \max_a Q_t\left(s_{t+1}, a_t\right)\right], \tag{12}$$

where $\alpha \in [0,1]$ is the learning rate, which determines to what extent newly acquired information overrides old information. In general, the agent learns nothing if $\alpha = 0$, while a factor of 1 makes the agent consider only the most recent information. The discount factor $\gamma \in [0,1]$ determines the importance of future rewards. If $\gamma = 0$, then the agent only considers the current reward which is short-sighted, while a factor approaching 1 will make it strive for a long-term high reward. As can be seen from Eq. (12), the Q-values corresponding to the pair of current state and action is updated according to the previous Q-values and new feedback reward. Meanwhile, ε-greedy algorithm can be applied to find the optimal action while the Q-values converge the maximum point after all state-action pairs (s_t, a_t) are visited as many as possible and also balance the exploration of a new state with random action. That is described in as follows (Eq. (13)):

$$a_t = \begin{cases} \arg\max Q\left(s_t, a_t\right) & 1-\varepsilon \\ \text{random action} & \varepsilon \end{cases}, \tag{13}$$

where $\varepsilon \in [0,1]$ is the exploration probability, which determines how fast the agent explores the external environment. The agent exploits the optimal action based on the highest Q-values with the probability of $1 - \varepsilon$. Otherwise, the agent chooses a random action to explore the external environment with probability ε.

State: defined as s_t, which reflects how the traffic situation in the environment at the time step t, in specific, how many vehicles are in the queues or pass the intersection is considered in this research. The reward function and actions of the agents are determined by the definition of the state space $S\left\langle s_1, s_2 \ldots s_n\right\rangle$.

Action: defined as a_t, action space $A\left\langle a_1, a_2 \ldots a_n\right\rangle$, which determines how much reward the agent can get, and the state of the environment ate the time step t + 1. In this case, controlling traffic lights is regarded as the actions of agents, which is similar to the decision vector as in Eq. (7) in game theoretical strategy part.

Reward: defined as r_t, which indicated how much positive benefits the agent can receive to control traffic flow more efficiently after they observe the state of the environment and take the selected actions. In general, the reward function is reverse to the cost function in game theoretical strategy, and it can be such as queue length, cumulative delay or throughput (i.e., the number of vehicles that go through the intersection). The reverse of the cost function of the player in Eq. (9) can be the reward function of the agent, which would find the maximum reward instead of minimum cost.

Configuration of input parameters such as initial Q-values Q_0, state s_0 and the number of calculation iterations is the first step in the RL procedure. Then the actions will be selected based on ε-greedy algorithm as in Eq. (13). After implementing these actions from optimal policy, the new state will be updated and the reward will be received by the agent. This process will be repeated until finish the iterations. A Single Agent Reinforcement Learning (SARL) algorithm is shown in Algorithm 1.

4 Implementation

4.1 Parameters

The parameters are shown Table 1 and Table 2. Table 1 shows the collision rate parameters $Y(w^1,z^1,w^2,z^2)$, (w^1,z^1) represents the first bent track where the vehicles are going to the target incoming link z from the initial incoming link w, similar to (w^2,z^2) which represents the second bent track. The value in Table 1 represents how much percent of the speed of the vehicle stream is reduced while two vehicles flow pass through the intersection from different directions at the same time. If the value is 0, then these two vehicle flow cannot be allowed to pass the intersection at the same time. Similarly, if it is 1 and they can both pass without any collision or interference. That can be simply expressed as:

$$F_{L,w^1z^1} = F_{L,w^1z^1,0} \prod\nolimits_{w^2z^2}\left(w^1, z^1, w^2, z^2\right), \tag{14}$$

Algorithm 1 Q-learning with a single agent

```
Result: The optimal action of the current state.
Initialization: initialize $Q$-function $Q_0$, initialize state
        $s_0$, number of iterations $X$;
While Iteration counter $c < X$ do
    Choose a random number $m \in [0,1]$
    If $m \in [\varepsilon,1]$ then
        $a_t = \arg\max Q(s_t, a_t)$;
    Else
        $a_t$ = random action
    End
    Implement the corresponding action $a_t$;
    Get the new state $s_{t+1}$ and the reward $r_t$;
    Update $Q$-function;
    $c = c + 1$;
End
```

Table 1 Initial value of collision rate $Y(w^1,z^1,w^2,z^2)$

$(w^1,z^1/w^2,z^2)$	(1,1)	(1,2)	(1,3)	(1,4)	(2,1)	(2,2)	(2,3)	(2,4)	(3,1)	(3,2)	(3,3)	(3,4)	(4,1)	(4,2)	(4,3)	(4,4)
(1,1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(1,2)	1	1	1	1	1	1	0	0	1	0	1	1	1	0	0	1
(1,3)	1	1	1	1	1	1	0	0	1	1	0.9	0	0	0	0.8	1
(1,4)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(2,1)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
(2,2)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(2,3)	1	0	0	1	1	1	1	1	0	1	0.9	0	1	0	0.8	1
(2,4)	1	0	0	1	1	1	1	1	0	1	1	0	0	1	1	1
(3,1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(3,2)	1	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1
(3,3)	1	1	0.5	1	1	0.5	1	1	1	1	1	1	1	1	0.9	1
(3,4)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(4,1)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(4,2)	1	0	0	1	1	1	0	1	0	0	1	0	1	1	1	1
(4,3)	1	1	1	0.5	1	1	0.5	1	1	1	0.9	1	1	1	1	1
(4,4)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 2 Initial value of turning rate t_{wz}, queues L_{wz}, capacities C_{wz} and incoming stream S_w

w/z	1	2	3	4	S_w
1	0, 0, 25	0.5, 33, 100	0.5, 83, 250	0, 0, 100	2
2	0.25, 33, 100	0, 0, 25	0.25, 33, 100	0.5, 83, 250	3
3	0, 0, 250	0.9, 33, 100	0.1, 8, 25	0, 0, 100	3
4	0, 0, 100	0.9, 83, 250	0, 1, 33, 100	0, 0, 25	2

Table 3 Comparison of decimal decisions in one cycle with three strategies

k_i/S_j	S_1	S_2	S_3
k_1	[6 0 0 0]	[0 13 6 4]	[0 13 6 4]
k_2	[0 13 0 4]	[0 13 6 4]	[0 13 6 4]
k_3	[0 0 6 0]	[6 1 4 4]	[6 1 4 0]
k_4	[0 9 0 6]	[0 13 6 4]	[0 13 6 4]

where $F_{L,w^1z^1,0}$ is the speed of the vehicle stream that would leave the intersection without disturbing by any other vehicles on other bent tracks.

The initial value of turning rate t_{wz}, queue length L_{wz}, and capacities C_{wz} are shown in Table 2, as well as the incoming stream S_w in the last column. Turning rate t_{wz} determines how much ratio of the incoming stream S_w will be split into a different path. E.g., the values of turning rate t_{wz} are (0, 0.5, 0.5, 0) corresponding to row $w = 2$ and $S_w = 2$, thus, the split vehicle stream for path 1, 2, 3, 4 of incoming link 2 is (0, 1, 1, 0) respectively. In specific, the value is (0.5, 33, 100) when it comes to the first row and the second column, which means the initial queue is 33, and capacity is 100 for this bent track (2-3).

4.2 Results

After implemented in Matlab, the performance of the constant strategy, game theoretical strategy, and SARL algorithm are compared.

As can be seen in Table 3, the optimal decimal decisions or actions in each incoming link of the intersection at each time slot k_i, $i = 1,\ldots,4$ for these methods, S_1, S_2 and S_3 represent constant strategy, game theoretical strategy, and RL algorithm respectively. At the time slot k_1, the decisions are [6 0 0 0] corresponding to S_1, which stands for the status of a traffic light for each incoming link respectively. Specifically, 6 represents the controlling of traffic light of the incoming link $w = 1$, the corresponding binary code of 6 is [0 1 1 0], so the traffic lights are [red green green red] respectively. Thus, all the status of traffic light can be controlled by the generated code in these methods.

Fig. 2 shows the Q-values from Q-function of all the permissible actions in a long-term iteration of SARL algorithm, which corresponds to the current state during the different time slots in the whole cycle time. The sub-figure for k_1 slot can be considered as an example to be introduced. Each curve represents the Q-values of a permissible action which are exploited and explored based on Eq. (13) and there are 112 permissible actions according to the collision rate in Table 1 corresponding to 112 curves in this sub-figure. This can be obvious that all the curves trend to be convergent as the iterations increase (i.e. 50000 iterations in one time slot), and the optimal action corresponding to the maximum Q-values will be selected in the end.

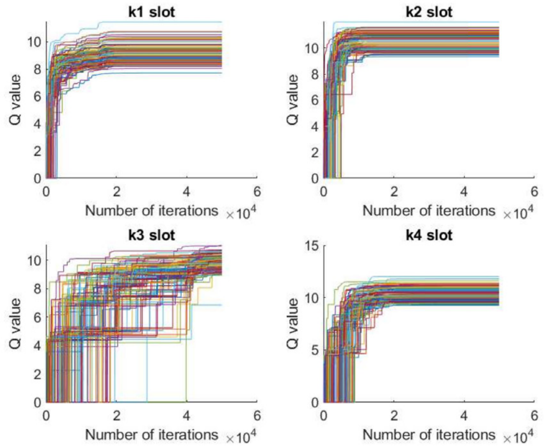

Fig. 2 Q-values of all the permissible actions in a long-term iterations

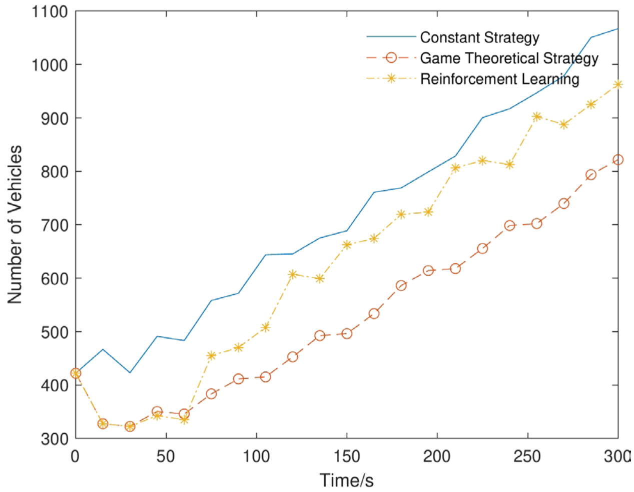

Fig. 3 Vehicles in queues (total) with three strategies in 5 cycles

The change of waiting vehicles in the queues and the passing vehicles in 5 cycles are compared in Fig. 3 and Fig. 4. The *x*-axis for both figures represents 5 cycle time totally, i.e., 300 seconds, and the *y*-axis represents the number of vehicles. In Fig. 3, the length of the queues keeps increasing for all the strategies since the incoming stream comes into the queues. Fig. 4 reflects how the number of the passed vehicles change, which trends to be periodical. In the end, the total passed vehicles in 5 cycles for one intersection are 2343, 2603, 2462 respectively.

5 Conclusion

A conclusion can be drawn that about 11.10 % and 5.08 % improvement are achieved by game theoretical strategy and SARL respectively, compared with constant strategy. Game theoretical strategy provides us with better performance in the traffic signal control, but still, some assumptions and limits are existing. Although SARL does not show the obvious effect in this experiment, it provides the potential performance to the extension method in the future, i.e., a decentralized control strategy called Multi-Agent Reinforcement Learning (MARL).

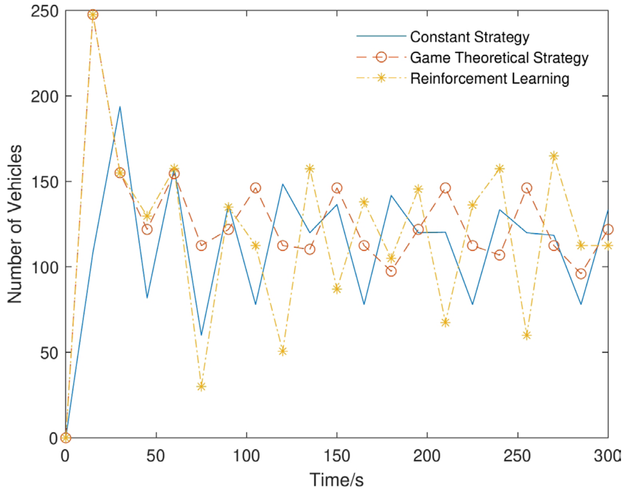

Fig. 4 Passed vehicles (total) with three strategies in 5 cycles

Acknowledgment

The research reported in this paper was supported by the Higher Education Excellence Program in the frame of Artificial Intelligence research area of Budapest University of Technology and Economics (BME FIKP-MI/FM). This project (EFOP-3.6.1-16-2016-00014) is also financed by the Ministry of Human Capacities.

References

Alvarez Villalobos, I., Poznyak, A. S., Tamayo, A. M. (2008) "Urban Traffic Control Problem: a Game Theory Approach", IFAC Proceedings Volumes, 41(2), pp. 7154–7159.
https://doi.org/10.3182/20080706-5-kr-1001.01213

Bakker, B., Whiteson, S., Kester, L., Groen, F. C. A. (2010) "Traffic Light Control by Multiagent Reinforcement Learning Systems", In: Babuška, R., Groen, F. C. A. (eds.) Interactive Collaborative Information Systems, Springer-Verlag, Berlin, Heidelberg, Germany, pp. 475–510.
https://doi.org/10.1007/978-3-642-11688-9_18

Başar, T., Olsder, G. J. (1998) "Dynamic noncooperative game theory", SIAM: Society for Industial and Applied Mathematics, Philadelphia, PA, USA.
https://doi.org/10.1137/1.9781611971132

Bui, K. H. N., Jung, J. J. (2018) "Cooperative game-theoretic approach to traffic flow optimization for multiple intersections", Computers & Electrical Engineering, 71, pp. 1012–1024.
https://doi.org/10.1016/j.compeleceng.2017.10.016

D'Acierno, L., Gallo, M., Montella, B. (2012) "An Ant Colony Optimisation algorithm for solving the asymmetric traffic assignment problem", European Journal of Operational Research, 217(2), pp. 459–469.
https://doi.org/10.1016/j.ejor.2011.09.035

El-Tantawy, S., Abdulhai, B. (2010) "An agent-based learning towards decentralized and coordinated traffic signal control", In: 13th International IEEE Conference on Intelligent Transportation Systems, Funchal, Portugal, pp. 665–670.
https://doi.org/10.1109/itsc.2010.5625066

Fan, H., Jia, B., Tian, J., Yun, L. (2014) "Characteristics of traffic flow at a non-signalized intersection in the framework of game theory", Physica A: Statistical Mechanics and its Applications, 415, pp. 172–180.
https://doi.org/10.1016/j.physa.2014.07.031

Gora, P. (2011) "A Genetic Algorithm Approach to Optimization of Vehicular Traffic in Cities by Means of Configuring Traffic Lights", In: Ryżko, D., Rybiński, H., Gawrysiak, P., Kryszkiewicz, M. (eds.) Emerging Intelligent Technologies in Industry, Springer-Verlag, Berlin, Heidelberg, Germany, pp. 1–10.
https://doi.org/10.1007/978-3-642-22732-5_1

Guo, J., Harmati, I. (2019) "Optimization of Traffic Signal Control with Different Game Theoretical Strategies", In: 2019 23rd International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania, pp. 750–755.
https://doi.org/10.1109/icstcc.2019.8885458

He, J., Hou, Z. (2012) "Ant colony algorithm for traffic signal timing optimization", Advances in Engineering Software, 43(1), pp. 14–18.
https://doi.org/10.1016/j.advengsoft.2011.09.002

Hoyer, R., Jumar, U. (1994) "Fuzzy control of traffic lights", In: Proceedings of 1994 IEEE 3rd International Fuzzy Systems Conference, Orlando, FL, USA, pp. 1526–1531.
https://doi.org/10.1109/fuzzy.1994.343921

Jabbarpour, M. R., Noor, R. M., Khokhar, R. H. (2015) "Green vehicle traffic routing system using ant-based algorithm", Journal of Network and Computer Applications, 58, pp. 294–308.
https://doi.org/10.1016/j.jnca.2015.08.003

Khamis, M. A., Gomaa, W., El-Shishiny, H. (2012) "Multi-objective traffic light control system based on Bayesian probability interpretation", In: 2012 15th International IEEE Conference on Intelligent Transportation Systems, Anchorage, AK, USA, pp. 995–1000.
https://doi.org/10.1109/itsc.2012.6338853

Liang, X., Du, X., Wang, G., Han, Z. (2019) "A Deep Reinforcement Learning Network for Traffic Light Cycle Control", IEEE Transactions on Vehicular Technology, 68(2), pp. 1243–1253.
https://doi.org/10.1109/tvt.2018.2890726

Teo, K. T. K., Kow, W. Y., Chin, Y. K. (2010) "Optimization of Traffic Flow within an Urban Traffic Light Intersection with Genetic Algorithm", In: 2010 Second International Conference on Computational Intelligence, Modelling and Simulation, Tuban, Indonesia, pp. 172–177.
https://doi.org/10.1109/cimsim.2010.95

Turky, A. M., Ahmad, M. S., Yusoff, M. Z. M., Hammad, B. T. (2009) "Using Genetic Algorithm for Traffic Light Control System with a Pedestrian Crossing", In: International Conference on Rough Sets and Knowledge Technology, Gold Coast, QLD, Australia, pp. 512–519.
https://doi.org/10.1007/978-3-642-02962-2_65

van der Pol, E., Oliehoek, F. A. (2016) "Coordinated Deep Reinforcement Learners for Traffic Light Control", In: 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, pp. 1–8. [online] Available at: https://pdfs.semanticscholar.org/4763/2b66387d00d19b66e71560ba462847b78006.pdf [Accessed: 08 May 2016]

Wiering, M. (2000) "Multi-agent reinforcement learning for traffic light control", In: Machine Learning: Proceedings of the Seventeenth International Conference (ICML'2000), Stanford, CA, USA, pp. 1151–1158.

Yousef, K. M. A., Shatnawi, A., Latayfeh, M. (2019) "Intelligent traffic light scheduling technique using calendar-based history information", Future Generation Computer Systems, 91, pp. 124–135.
https://doi.org/10.1016/j.future.2018.08.037

Zhang, M. M., Jia, L. (2011) "Mathematical model of traffic flow on arteries with coordinated control system", Control Theory & Applications, 28(11), pp. 1679–1684. [online] Available at: http://en.cnki.com.cn/Article_en/CJFDTotal-KZLY201111024.htm [Accessed: 15 April 2011]