Songklanakarin J. Sci. Technol.
44 (4), 914–922, Jul. – Aug. 2022

http://www.sjst.psu.ac.th

Original Article

An adaptive traffic light control system using reinforcement learning

Kietikul Jearanaitanakij*, Chanayut Jamkhaw, Nattapat Puangpipat, and Tot Worasrivisal

Department of Computer Engineering, School of Engineering,
King Mongkut's Institute of Technology Ladkrabang, Lat Krabang, Bangkok, 10520 Thailand

Received: 31 July 2021; Revised: 27 February 2022; Accepted: 8 March 2022

Abstract

Traffic signal control (TSC) is a challenging issue in managing an urban transportation system. A fixed time TSC is easy to implement but has drawbacks in such measures as flow rate, waiting time, and traffic density. The situation gets worse when the arrival rates of vehicles periodically change over time, which is usual in most urban cities. We propose adaptive reinforcement learning (RL) to manage TSC with varying vehicle arrival rates. Our objectives are to improve the averages of flow rate and waiting time and reduce the wasteful green light problem by considering the vehicle densities of the current lane and the downstream directions. Experiments were conducted by Simulation of Urban MObility (SUMO) under three traffic layouts and various vehicle arrival rates. The proposed method not only reduced on average traffic density, waiting time, and queue length, but also increased the average flow rate and average speed, relative to the other algorithms tested.

Keywords: traffic signal control, transportation, reinforcement learning, adaptive green light time, wasteful green light problem

1. Introduction

According to a United Nations report released in 2018, more than half of the world's population lives in urban areas. This proportion may increase to 68% by 2050. Therefore, it is necessary to intelligently manage the traffic infrastructures to match the rapid population growth. The strategies of traffic control fall into 3 classes: fixed time, actuated, and adaptive (Feng, 2015). Fixed time control is a conventional method that predefines the timing of signal periods based on either statistical traffic information or a certain time interval. However, if the traffic flow has unpredictable fluctuations, such system may not handle the situation well. On the other hand, an actuated strategy adjusts a simple traffic light control parameter such as cycle length, green light extension, or phase sequence in response to sensor information. However, these adjustments are still limited within a set of predefined parameters. Apart from the previous approaches, an adaptive strategy is more advanced in that it utilizes the information from the sensors, responding to the actual traffic demand by changes in the traffic signal timing. This strategy can handle a wider range of traffic fluctuations than the actuated approach since the adjustments are applied to the traffic signal policy, not just being a temporary response.

Research on TSC has grown rapidly. Some recent interesting works are discussed below. (Mousavi, Schukat, & Howley, 2017) developed dual adaptive agents (deep policy-gradient and value-function) to predict the best traffic signal for an intersection. The first agent maps its observations to control signal policy while the second agent estimates values of control signals. (Liang, Du, Wang, & Han, 2019) applied deep RL to the data collected from various sensors and used a convolutional neural network to map states to rewards. Both approaches of (Mousavi *et al.*, 2017) and (Liang *et al.*, 2019) achieved promising results in the SUMO traffic simulator for one intersection. However, no simulations of multiple intersections are reported in their experiments. Another way for solving the TSC problem is to employ a heuristic-based strategy, as done in Araghi, Khosravi, Creighton, and Nahavandi (2017), Garcia-Nieto, Olivera, and Alba (2013), He, Head, and Ding (2011), and Zargari, Dehghani, and Mirzahossein (2018). Franco, Lindsay, Vallati, and McCluskey (2018) introduced a time-based highly informative heuristic for planning urban traffic control. The heuristic estimates the distance from the goal state by considering the expected input/output traffic flows. Their experimental results in both the city of Manchester (UK) and in challenge

*Corresponding author
Email address: kietikul.je@kmitl.ac.th

scenarios outperformed the state-of-the-art planning engine (Penna, Magazzeni, Mercorio, & Intrigila, 2009). Aside from centralized control approach, some studies have utilized a regional method to solve the TSC problem, e.g. Hiari and Nofal (2020), Jin and Ma (2017), Le, Kovacs, Walton, Vu, Andrew, and Hoogendoorn (2015), and Nilsson and Como (2018). A decentralized method (Wei *et al.*, 2019) utilized multiple RL agents in many intersections. Tan *et al.*, (2019) proposed a cooperative framework by decomposing the original RL task into subproblems with easier goals. A centralized global agent solves the original task by using the information gathered from multiple regional agents. According to their experimental results, the proposed framework reduce congestion by 30% in terms of the number of waiting vehicles in high traffic congestion. Wang *et al.*, (2020) proposed Spatio-Temporal RL using multiple agents. A traffic light adjacency graph was constructed to represent the spatial structure among traffic lights. Then, the temporal traffic information was combined with traffic structure via a recurrent neural network. Their experimental results provide insights into the mechanisms associated with multi-intersection traffic lights. Zang *et al.*, (2020) introduced the MetaLight framework to improve their previous RL model by upgrading its structure and updating strategy. As a result, the framework adapts to new traffic scenarios more quickly and steadily. Recently, Joo, Ahmed, and Lim (2020) applied Q-learning to boost the number of vehicles crossing a junction and balance the traffic signals among roads. However, their constant green light time may be wasteful when the density fluctuates.

Among all the approaches mentioned above, none has completely addressed unpredictable changes in urban traffic conditions, such as sudden changes among multiple levels of vehicle arrival rate and temporally sparse congestion. Moreover, none has simultaneously sought to improve both flow rate and waiting time, which are likely the two main concerns of most drivers. To address these challenges, we propose a real-time RL approach that adapts to traffic incidents by collecting current traffic information from sensors and selecting the green light direction that maximizes the average flow rate and minimizes the average waiting time. It also automatically adjusts the green light duration based on the densities of the vehicles in the current and downstream lanes. The experimental results on various vehicle arrival rates and multi-intersections indicate that the proposed method outperforms both fixed time and state-of-the-art algorithms in all traffic layouts. The contributions of this paper are summarized as follows. First, the approach demonstrated concurrently improves the averages of flow rate and waiting time of vehicles in the system. This improvement addresses both the traffic measure in theory and the driver's needs in practice. Second, the approach automatically adjusts the green light time based on the densities of vehicles in the current and downstream lanes. In other words, it reduces the wasteful green light time problem when vehicles in a certain lane cannot proceed due to the traffic congestion ahead. Finally, it truly represents both flow rate and waiting time via the logistic sigmoid function with appropriate coefficients. As a result, its reward function can accurately distinguish the quality of states at high and low ranges of the averages flow rate and waiting time.

2. Materials and Methods

In this section, we provide fundamental background knowledge on reinforcement learning and important terms in traffic engineering. The components of the proposed method are explained thereafter.

2.1 Reinforcement learning

Reinforcement learning is an approach that maps situations to actions to maximize a numerical reward signal. Sutton and Barto (2018), typically over multiple actions as time progresses. The agent must discover which actions yield the most reward by trial-and-error search. Q-Learning algorithm trains the value of an action in a particular state by finding an optimal policy to maximize the expected total reward cumulated over all successive steps. We calculate the new Q value, Q_{new} (S_t, A_t), associated with the state S_t and action A_t pair by the following function.

$$Q_{new}(S_t, A_t) = Q(S_t, A_t) + \alpha\left[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)\right],$$

where $Q(S_t, A_t)$ is the Q value for the current state S associated with action A at time t, α is the learning rate, R_{t+1} is the observed reward for entering the next state at time t + 1, γ is the discount factor for determining the current importance of future rewards, and $\max_a Q(S_{t+1}, a)$ is the maximum Q value for the next state associated with action a.

2.2 Flow rate, speed, and density

According to McDowall and Dampney (2006), the three primary measures of traffic stream characteristics are flow rate, speed, and density. Flow rate is the number of vehicles passing a point on a given lane or direction of a road in one hour. Speed is defined as a rate of motion in distance per unit of time. In a moving traffic stream, vehicles move at different speeds. Therefore, a proper way to analyze speed of the system is to average the speeds across of all vehicles. Density is defined as the number of vehicles occupying a given length of lane, normally expressed as vehicles per unit of length. The relationship between the three measures --flow rate (υ), speed (S), and density (D)-- for a given traffic stream is as follows.

$$\upsilon = S * D$$

Speed and density are measures that refer to a specific range or area, whereas flow rate is a point measure. Under stable flow conditions, where no queues are forming, the flow rate computed by the above equation can apply to any point within the range. In contrast, if a queue is forming the flow rate can only represent an average for all points within the range.

2.3 Proposed method

Since we employ the traditional RL algorithm by using a single agent scenario for all intersections with individual state spaces, we define only terms which are customized for our approach.

2.3.1 Traffic layout

Figure 1(a) illustrates an intersection layout used in SUMO. Each road has two lanes in each direction with left-hand traffic as the rule of the road. The vehicles in the left lane can either move straight or turn left (if applicable) while those in the right lane are forced to turn right. This kind of traffic layout conforms with (Joo *et al.*, 2020). It tends to have more traffic congestion than other layouts when the vehicle arrival rate suddenly changes.

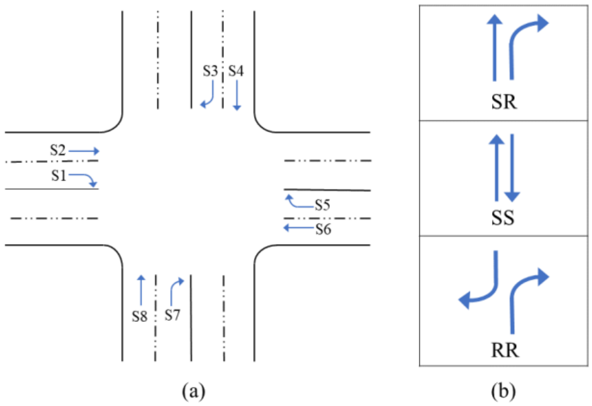


Figure 1. Traffic layout, states, and actions

2.3.2 States and actions

Each 4-legged intersection has its own state space defined as {S1, S2, **…**, S8}, where Si is the state at the i-th lane in Figure 1(a). Suppose the current state is Si, vehicles in the i-th lane can move to the corresponding direction depending on the action selected by the RL algorithm which learns each intersection individually. The green light is assigned to the directions specified by arrows in the action. Three possible actions are Straight-Right (SR), Straight-Straight (SS), and Right-Right (RR), as shown in Figure 1(b). An action of the current state determines a set of possible next states. For instance, a set of possible next states after performing the action SR at the state S1 is {S3, S4, S5, S6, S7, S8}. S1 and S2 are excluded from the set because vehicles in both states just move at the current time step. All possible state transitions in the Markov model are shown in Table 1.

2.3.3 Reward function

Since one of the objectives of this research is to simultaneously optimize both the averages of waiting time and flow rate, let us call it WTFR from now on. Our reward function is defined as follows.

$$Reward = \frac{\widehat{FR}}{1 + \widehat{WT}},$$

where $\widehat{FR}$ and $\widehat{WT}$ are normalized terms (between $0{\sim}1$) of averages of flow rate and waiting time, respectively. We normalize both values by using a logistic sigmoid function.

Table 1. State transitions

Current State	Action	Possible next states
S1	SR	{s3, s4, s5, s6, s7, s8}
	RR	{s2, s3, s4, s6, s7, s8}
s2	SR	{s3, s4, s5, s6, s7, s8}
	SS	{s1, s3, s4, s5, s7, s8}
s3	SR	{s1, s2, s5, s6, s7, s8}
	RR	{s1, s2, s4, s5, s6, s8}
s4	SR	{s1, s2, s5, s6, s7, s8}
	SS	{s1, s2, s3, s5, s6, s7}
s5	SR	{s1, s2, s3, s4, s7, s8}
	RR	{s2, s3, s4, s6, s7, s8}
s6	SR	{s1, s2, s3, s4, s7, s8}
	SS	{s1, s3, s4, s5, s7, s8}
s7	SR	{s1, s2, s3, s4, s5, s6}
	RR	{s1, s2, s4, s5, s6, s8}
s8	SR	{s1, s2, s3, s4, s5, s6}
	SS	{s1, s2, s3, s5, s6, s7}

$$\widehat{FR} = \frac{1}{1 + e^{-\beta(FR-\gamma)}},$$

$$\widehat{WT} = \frac{1}{1 + e^{-\beta(WT-\gamma)}},$$

where *FR* is the average flow rate and *WT* is the average waiting time. To avoid the saturated values of lower and upper plateaus, we apply the approach of McDowall and Dampney (2006) to calculate appropriate coefficients β and γ for the sigmoid function. The general form of the logistic sigmoid function (Kent, Drane, Blumenstein, & Manning, 1972) of an input X is as follows.

$$Y = \frac{\alpha}{1 + e^{-\beta(X-\gamma)}} + \delta,$$

where α is the *Y* range (1 in our case), β is the gain coefficient, and γ is the midpoint of the *X*-axis. The value of γ can be either a half of the maximum flow rate or half of the maximum waiting time, depending on whether we are calculating ($\widehat{FR}$ or $\widehat{WT}$). δ is the value of Y at the bottom plateau (0 in our case). Assume by default that the logistic sigmoid function saturates within 5% of the upper and lower plateaus, i.e., the saturated values of *Y* on both ends are 0.05 and 0.95. A coefficient β can be calculated by the following equations.

$$0.95 = \gamma - \frac{2.944}{\beta},$$

$$\beta = \frac{2.944}{\gamma - 0.95}.$$

For example, given $\gamma = 50$ and $\beta = \frac{2.944}{50-0.95} = 0.06$, the logistic sigmoid function can be nicely normalized as shown in Figure 2. In practice, we need to estimate the maximum flow rate and the maximum waiting time to find the appropriate midpoint γ for them in the preliminary run. This logistic sigmoid function with an appropriate slope can produce distinct Y values when X values are large (or small)

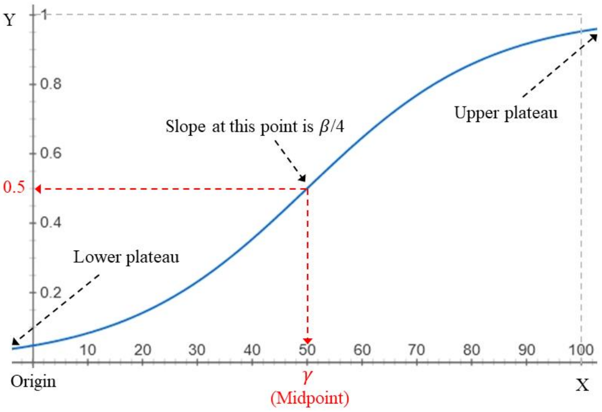


Figure 2. Logistic sigmoid function with appropriate slope

since the Y range nicely scales within the range of 0.5 and 0.95 without plateaus. As a result, WTFR can discern the reward values when the averages of flow rate and waiting time are large (or small).

2.3.4 Adaptive green light time

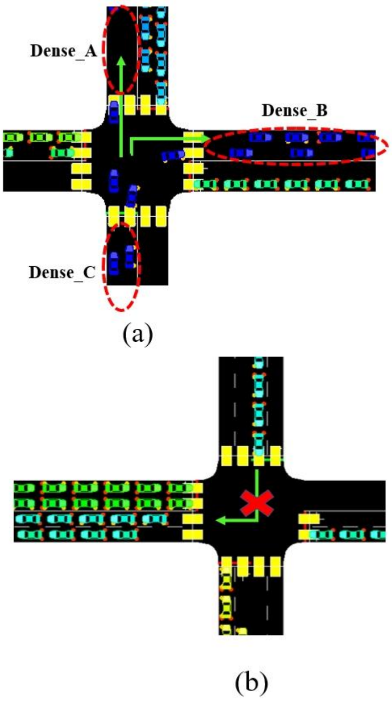
Another important factor that improves the adaptability of WTFR is the automatic adjusting of green light time (GT). GT of action varies directly with the vehicle density of the current state (*Dense_C*) and varies inversely to the average vehicle density of the downstream lanes (*Dense_A* and *Dense_B*).

$$GT = \frac{Dense_C}{0.5 * (Dense_A + Dense_B)} \times 40,$$

where *Dense_A*, *Dense_B* are densities of lanes downstream and *Dense_C* is the density of the current state, as shown in Figure 3(a). The fraction term determines whether to extend or reduce the default 40-second green light duration. To prevent the domination of a particular lane, the green light time calculated from the above equation is limited within the range between 20 and 60 seconds. This adaptive green light time significantly reduces the wasteful green light time problem when vehicles in a certain lane cannot proceed due to the traffic congestion ahead as shown in Figure 3(b). Although receiving the green light signal, vehicles in the north lane cannot turn right since the west lane has reached its capacity. In other words, the adaptive GT adjusts the green light duration in proportion to the traffic congestion levels of both the current (*Dense_C*) and downstream (*Dense_A* and *Dense_B*) lanes. As a result, this mechanism can reduce the chance of vehicle overflow as the length of the waiting vehicles is less than the distance between intersections.

3. Results and Discussion

To understand the adaptability of various TSC algorithms, we experiment under the condition where the vehicle arrival rate changes frequently. A total of 18,000 vehicles are randomly fed into the system with different routes. The arrival rate of 6 consecutive intervals is varied in the following chronological order: 4,500 – 6,000 – 4,500 – 3,600 – 4,500 – 6,000 vehicles per hour. The last vehicle arrives at 13,000th second. This fluctuating arrival rate causes a severe problem to most TSC algorithms, as we will see in the simulation. Other experimental settings are listed in Table 2. The maximum simulation hours for 4-way, 16-way, and 36-way traffic layouts are 3, 4.5, and 6.1, respectively.



(a)

(b)

Figure 3. Densities of 3 lanes and the wasteful green time problem

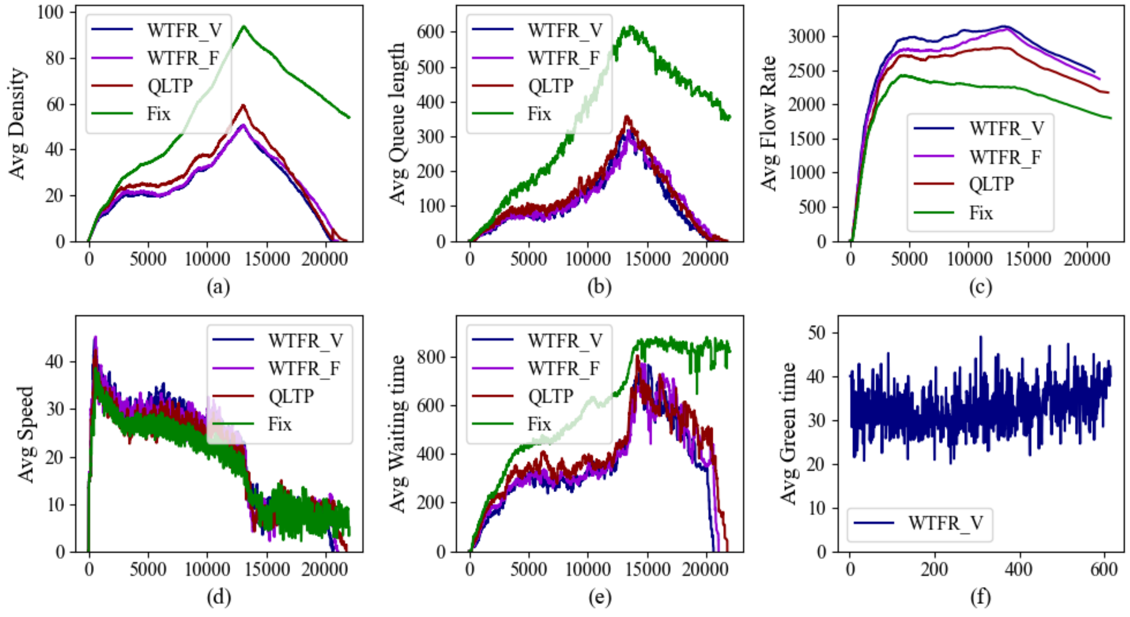
Table 2. Experimental settings

Setting	Value
Traffic Layout (No. of intersections / No. of ways)	1 / 4-way 4 / 16-way 9 / 36-way
Vehicle arrival rate (vehicles/hour)	Variation of arrival rates in 6 intervals 4,500 – 6,000 – 4,500 – 3,600 – 4,500 – 6,000
Road length between 2 intersections	2.5 km
Vehicle length	4.7 m
Min. gap between vehicles	1.3 m
Learning rate	0.1
Discount factor	0.9

We compare the proposed method with the traditional fixed time approach and the recent state-of-the-art RL algorithm (Joo *et al.*, 2020). Let us label the latter approach with QLTP since it focuses on optimizing queue lengths and throughput. For each leg of the intersection, the green light duration of the fixed time method for the Straight-Right (SR) action is 60 seconds. We pick a 60-second green signal because its preliminary simulation showed the best result among tested durations. Similarly, QLTP employs a constant green light time. Its green light time is also 60 seconds for a fair comparison. To see the impact of the green light time variation, we split the proposed method (WTFR) into two versions, i.e., a fixed 60-second green light time (WTFR_F) and the varied green light time (WTFR_V). To solve unpredictable changes in vehicle arrival rate, all RL algorithms continuously learn the traffic environment until there is no vehicle left in the system.

Six average measures of a 9-intersection traffic map (36-way) are shown in Figure 4. It is worth noting that each horizontal axis of Figures 4(a-e) is time in seconds while Figure 4(f) displays value against epoch. Averages of density and queue length in Figures 4(a-b) vary in similar patterns. They indicate the accumulation of successive vehicles with different arrival rates. Both measures rapidly increase until the traffic capacity has reached the critical point around 13,000th second as the last vehicle arrives. Cumulative vehicles during this moment form severe traffic congestion, as we noticed in SUMO that many vehicles cannot move forward despite receiving the green light signal. As a result, green light time in some directions is wasteful. Figure 4(c) indicates that the average flow rates of all methods quickly rise to their near-saturation points around the 4,000th second and begin to decrease due to the traffic congestion. Combining information from Figures 4(a-b), the severe traffic congestion occurs for 4,000th – 13,000th seconds, which is long enough to evaluate the performance of TSC algorithms. According to the average vehicle speed in Figure 4(d), WTFR_V and WTFR_F allocate the top speed band up until the critical point, which causes the average speeds of the 4 methods to sharply drop. Interestingly, WTFR_V and WTFR_F are the first to clear all vehicles and terminate the simulation as their average speed reaches zero around 21,000th second. The average waiting time in Figure 4(e) is another factor that influences reward in our method. WTFR_V and WTFR_F have a lower average waiting time than the other methods, while the fixed-time strategy cannot finish the simulation within 22,000 seconds. Figure 4(f) illustrates the variation of the average green light time by epoch. WTFR_V is the only method that takes advantage of varying the green light duration to reduce the wasteful green light problem. It is interesting that WTFR_V has an average green light time of around 33 seconds and never sets its green light time to 60 seconds. We will see how much of the average waiting time that WTFR_V can save in the experimental comparison.

The comparisons of averages of 4 measures among the alternative RL strategies for various traffic layouts, namely 4-way (1 intersection), 16-way (2x2 = 4 intersections), and 36-way (3x3 = 9 intersections), are illustrated in Figures 5(a-d), 5(e-h), and 5(i-l), respectively. The comparison of average speeds is not shown here because it is similar to that of the average flow rates. The three RL strategies outperformed the fixed time method in all measures. Although WTFR_V, WTFR_F, and QLTP look competitive, there are significant gaps among them. Figure 6 shows the percentage of improvement by the three RL algorithms over the baseline fixed time method in 4-way (Figures 6(a-d)), 16-way (Figures 6(e-h)), and 36-way (Figures 6(i-l)) cases. In the density and queue length averages in Figures 6(a-b, e-f, i-j), the RL algorithms achieved improvements over the fixed time approach by approximately 46-55% in all traffic layouts. Moreover, WTFR_V had a better improvement than QLTP by approximately 5%. Figures 6(c-d, g-h, k-l) shows two further comparisons of terms that are crucial in calculating the reward of WTFR. The improvement of the average flow rate and waiting time of WTFR_V are better than those of QLTP by 12% and 5-6%, respectively. These results support the first contribution in that WTFR_V improved the averages of flow rate and waiting time over the fixed time and the recent state-of-the-art QLTP methods. Improving the waiting time is harder than improving the flow rate because SUMO counts the number of seconds the vehicle's velocity stays below 0.1m/s as waiting time even though the vehicle moves slowly.



(a) (b) (c)

(d) (e) (f)

Figure 4. Measures of traffic stream characteristics in the 36-way layout

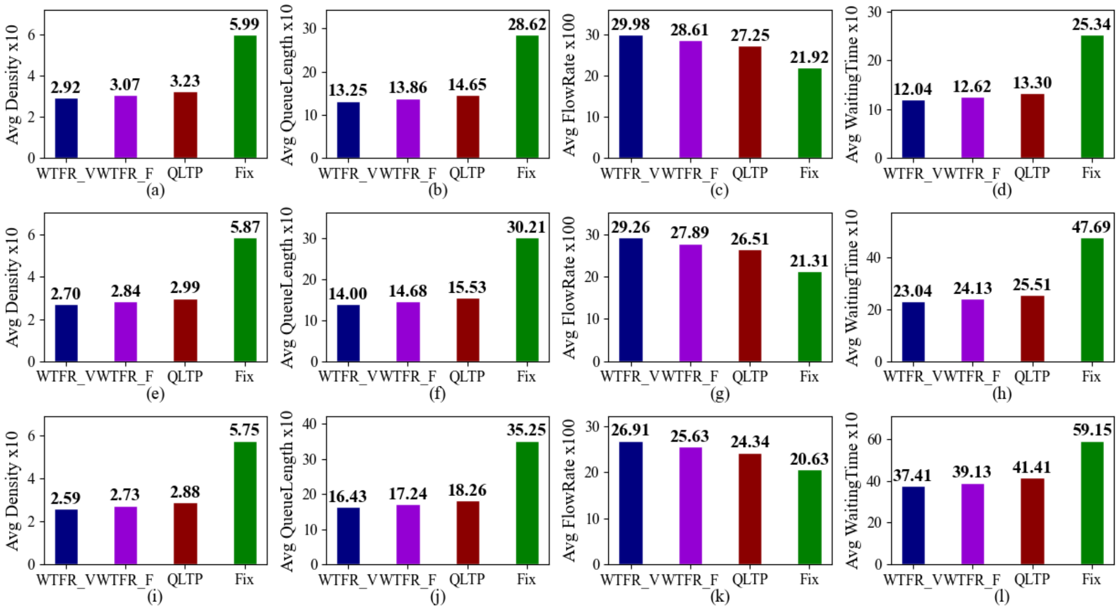


Figure 5. Comparison of the four methods in three traffic layouts

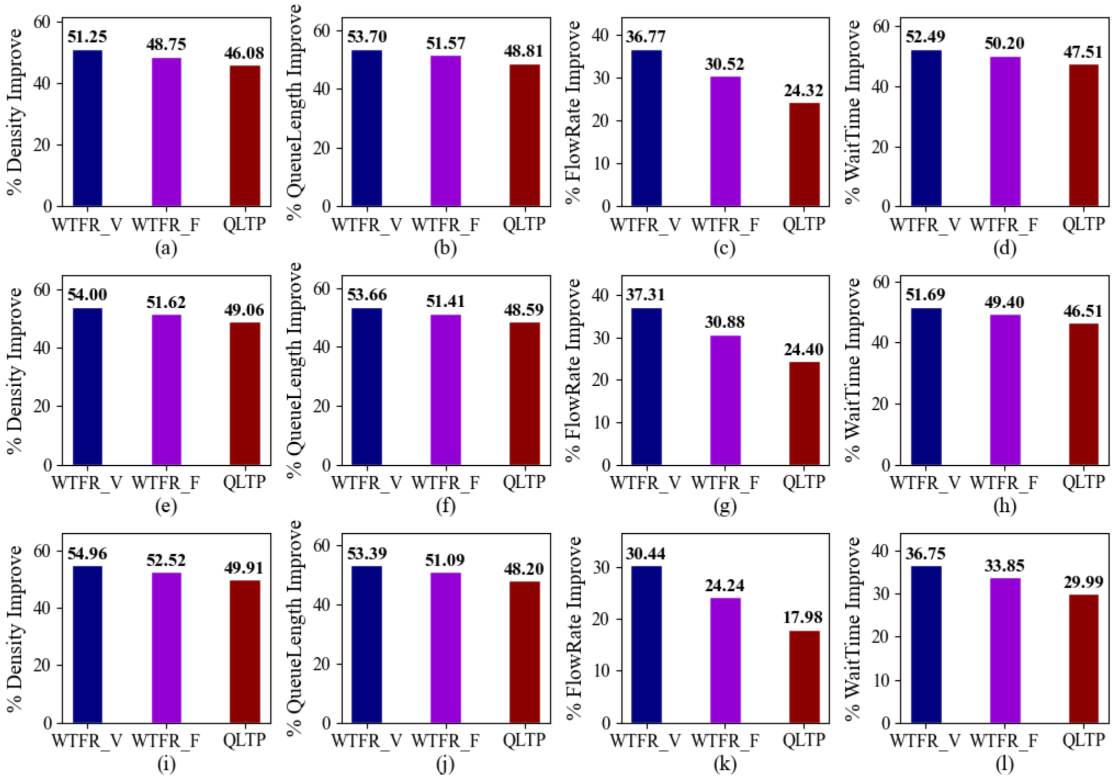


Figure 6. Improvements over the fixed-time method

The constant green light time strategy seems to be inefficient in controlling the traffic balance between routes having different densities, leading to a wasteful green light in some directions. Unfortunately, SUMO does not provide an API to measure wasteful green light time. Therefore, we will indirectly see the effect of varying the green light time via the improvement of average of flow rate. Figure 6 also compares the results between WTFR_V (our varied green light time version) and WTFR_F (our constant green light time version). Compared to WTFR_F, Figures 6(c, g, k) indicate that

WTFR_V has a further improvement of the average of flow rate by about 6%. This indirectly supports the second contribution in that it reduces the wasteful green light time problem by increasing the average flow rate. Comparing to QLTP in Figures 6(a-l), WTFR_F is still better in all measures, even though these employ the same constant green light time. The following discussion explains the reason for this result.

To understand the advantage of using the logistic sigmoid function with appropriate coefficients, we created another version of WTFR_F called WTFR_F2 by simply substituting 1 for β and 0 for γ in the sigmoid function that took the following form.

$$Y = \frac{1}{1+e^{-X}}$$

This simple form of the logistic sigmoid function has a saturation problem when the input is in the high or the low range. We prove this statement by comparing the improvements over the fixed time method by the three algorithms in Figure 7. According to all measures in Figures 7(a-l), WTFR_F2 has a similar performance to QLTP that also employs the simple logistic sigmoid function to calculate the adaptive weighting factor based on the arrival of vehicles. When using appropriate coefficients of the logistic sigmoid function, Figures 7(c, g, k) indicates that WTFR_F produced a better average flow rate than WTFR_F2 and QLTP by 6%. A logistic sigmoid function with inappropriate coefficients saturates at high and low input ranges, resulting in inability to discern the rewards in these input ranges. This experimental result endorses the third contribution in that our reward function can accurately distinguish the quality of states at high and low ranges of average flow rate and waiting time, leading to improved traffic measures.

Figure 8 describes the variations of reward in one intersection of the RL methods. The rewards of WTFR_V and WTFR_F are similar in pattern, while the reward of QLTP is much higher. The reason for the gap between QLTP and the other two algorithms is the difference in reward function. QLTP calculates reward by taking the logarithm (based 0 – 1) function to the input which is composed of the standard deviation of queue length and the decreasing exponential of throughput. During the high traffic period, the input of the logarithm function is small resulting in a large reward value.

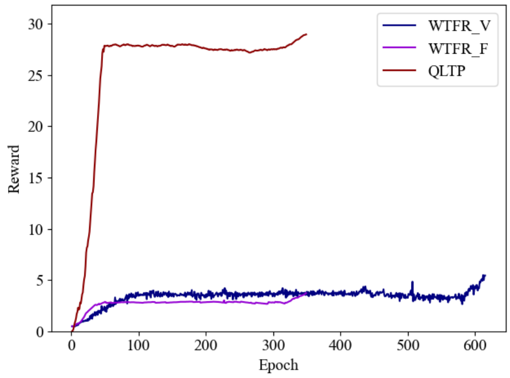


Figure 8. Rewards of three methods

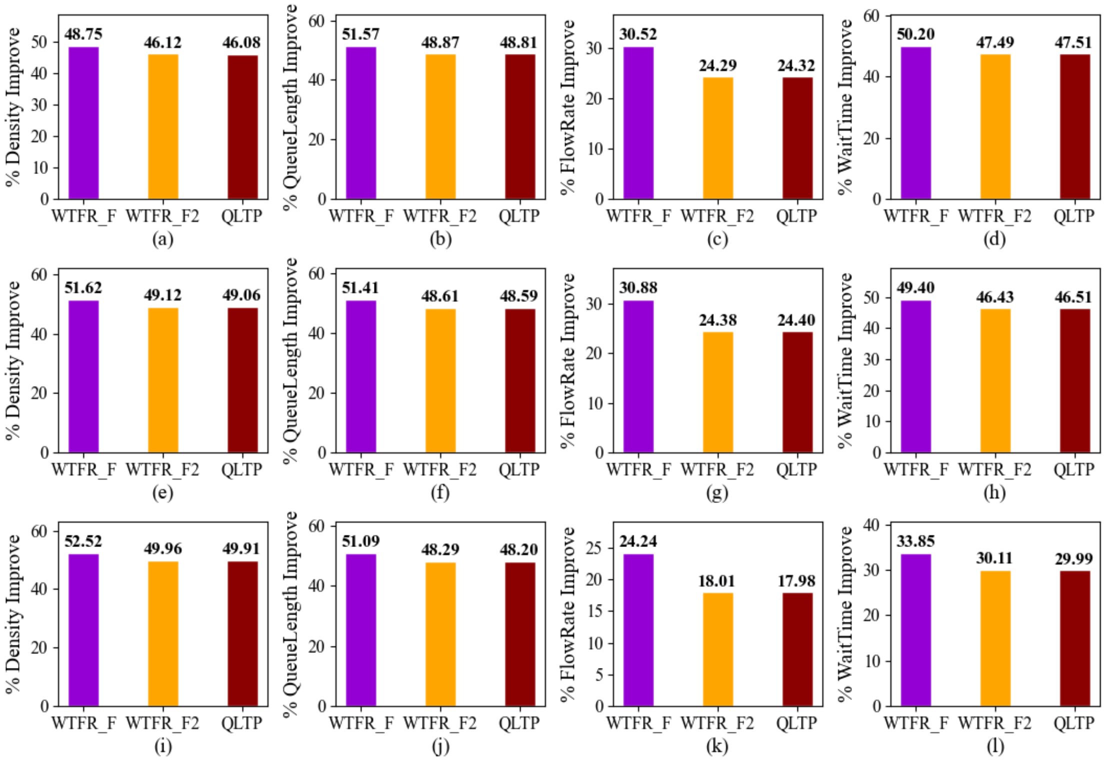


Figure 7. Advantage of having appropriate coefficients in the logistic sigmoid function

However, the larger reward does not reflect the higher performance since all alternative states in QLTP have high rewards as well. The learning epoch count of WTFR_V is longer than those of the other two approaches because its average green light time is shorter than a constant green light. Therefore, WTFR_V needs a relatively short time to simulate one cycle of all intersections. As a result, it consumes more epochs than the other two RL algorithms.

4. Conclusions

We have proposed an adaptive traffic light control system using the RL algorithm. The simulation results in SUMO endorse successful pursuit of our objectives. Compared to the other algorithms tested, the proposed method improved averages of important traffic measures, especially the flow rate and the waiting time. Moreover, it also reduced the wasteful green light problem by automatically adjusting the green light duration based on the densities of vehicles in the current and downstream lanes. A limitation of the proposed system is the requirement of sensors for flow rate, waiting time, and density of vehicles. Without those data, the proposed system cannot calculate the reward function and the green light time. Since WTFR uses a reinforcement learning scheme, it can lead to the exploitation-exploration tradeoff. Exploitation repeats existing actions to maximize the long-term reward, but may not be optimal. In contrast, exploration randomly chooses a new action in the hope of achieving near-optimal rewards. A suitable situation for deploying the proposed method is an environment with unpredictable changes in traffic conditions. Avoid applying it in a fix-green-time environment since there will be no adaptivity benefit from the RL strategy. One possible future extension of this research is to apply hierarchical RL that combines reward tables from all intersections and optimizes the overall reward. This pyramid approach needs a proper design, as otherwise it will significantly increase the running time and cannot be applied to a real-time TSC system.

References

Araghi, S., Khosravi, A., Creighton, D., & Nahavandi, S. (2017). Influence of meta-heuristic optimization on the performance of adaptive interval type2-fuzzy traffic signal controllers. *Expert Systems with Applications, 71*, 493–503. doi:10.1016/j.eswa. 2016.10.066.

Feng, Y., Head, L., Khoshmagham, S., & Zamanipour M. (2015). A real-time adaptive signal control in a connected vehicle environment. *Transportation Research Part C: Emerging Technologies, 55*, 460-473. doi:10.1016/j.trc.2015.01.007

Franco, S., Lindsay, A., Vallati, M., & McCluskey, T. L. (2018). An innovative heuristic for planning-based urban traffic control. In Y. Shi *et al.* (Eds.), *Lecture Notes in Computer Science: Vol. 10860* (pp. 181-193). Cham, Switzerland: Springer. doi:10.1007/ 978-3-319-93698-7_14

Garcia-Nieto, J., Olivera, A. C., & Alba, E. (2013). Optimal cycle program of traffic lights with particle swarm optimization. *IEEE Transactions on Evolutionary Computation, 17*(6), 823–839. doi:10.1109/TEVC. 2013.2260755

He, Q., Head, L., & Ding, J. (2011). Heuristic algorithm for priority traffic signal control. *Transportation Research Record: Journal of the Transportation Research Board*, 2259(1), 1-7. doi:10.3141/2259-01

Hiari, O., & Nofal, I. (2020). A dynamic decentralized traffic light management system: A TCP inspired approach. *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, 1-4, doi:10.1109/NOMS47738.2020.9110461

Jin, J., & Ma, X. (2017). A decentralized traffic light control system based on adaptive learning. *20th IFAC World Congress*, 50(1), 5301-5306. doi:10.1016/j.ifacol. 2017.08.958

Joo, H., Ahmed, S. H., & Lim, Y. (2020). Traffic signal control for smart cities using reinforcement learning. *Computer Communications*, *154*(1), 324-330. doi:10.1016/j.comcom.2020.03.005

Kent, B. B., Drane, J. W., Blumenstein, B., & Manning, J. W. (1972). A mathematical model to assess changes in the baroreceptor reflex. *Cardiology*, *57*(5), 295–310. doi:10.1159/000169528

Le, T., Kovacs, P., Walton, N., Vu, H. L., Andrew, L. L., & Hoogendoorn, S. S. (2015). Decentralized signal control for urban road networks. *Transportation Research Part C: Emerging Technologies*, *58*, 431-450. doi:10.1016/j.trc.2014.11.009

Liang, X., Du, X., Wang, G., & Han, Z. (2019). Deep reinforcement learning for traffic light control in vehicular networks. *IEEE Transactions on Vehicular Technology*, *68*(2), 1-11. doi:10.1109/ TVT.2018.2890726

McDowall, L. M., & Dampney, R. A. L. (2006). Calculation of threshold and saturation points of sigmoidal Baroreflex function curves. *American Journal of Physiology - Heart and Circulatory Physiology, 291*(4), H2003-H2007. doi:10.1152/ajpheart.00219. 2006

Mousavi, S., Schukat, M., & Howley, E. (2017). Traffic light control using deep policy-gradient and value-function based reinforcement learning. *IET Intelligent Transport Systems*, *11*(7), 1-8. doi:10. 1049/iet-its.2017.0153

Nilsson, G., & Como, G. (2018). Evaluation of decentralized feedback traffic light control with dynamic cycle length. *15th IFAC Symposium on Control in Transportation Systems CTS 2018*, *51*(9), 464-469. doi:10.1016/j.ifacol.2018.07.076

Penna, G. D., Magazzeni, D., Mercorio, F., & Intrigila, B. (2009). UPMurphi: A tool for universal planning on PDDL+ problems. *Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS-09)* (pp. 106 - 113).

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction (Adaptive Computation and Machine Learning series)*. Cambridge, MA: MIT Press.

Tan, T., Bao, F., Deng, Y., Jin, A., Dai, Q., & Wang, J. (2019). Cooperative deep reinforcement learning for large-scale traffic grid signal control. *IEEE Transactions on Cybernetics*, *50*(6), 2687-2700. doi:10.1109/TCYB.2019.2904742

Wang, Y., Xu, T., Niu, X., Tan, C., Chen, E., & Xiong, H. (2020). STMARL: A spatio-temporal multi-agent reinforcement learning approach for cooperative traffic light control. *IEEE Transactions on Mobile Computing*. doi:10.1109/TMC.2020.3033782

Wei, H., Chen, C., Wu, K., Zheng, G., Yu, Z., Gayah, V. V., & Li, Z. (2019). Deep reinforcement learning for traffic signal control along arterials. *Workshop on Deep Reinforcement Learning for Knowledge Discovery*, 1-7.

Zang, X., Yao, H., Zheng, G., Xu, N., Xu, K., & Li, Z. (2020). MetaLight: Value-based meta-reinforcement learning for traffic signal control. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(01), 1153-1160. doi:10.1609/aaai.v34i01.5467

Zargari, S. A., Dehghani, N., & Mirzahossein, H. (2018). Optimal traffic lights control using meta heuristic algorithms in high priority congested networks. *The International Journal of Transportation Research*, *10*(3), 1-13. doi:10.1080/19427867.2016.1241921