Hierarchical Traffic Signal Coordination with Priority-based Optimization via Deep Reinforcement Learning

H. Kim[a], H. Tak[a], H. Yu[a] and H. Yeo[a,*]

[a] Department of Civil and Environmental Engineering, Korea Advanced Institute of Science and Technology, Daejeon, South Korea
{hyunsookim, hytak, yhp7237, hwasoo}@kaist.ac.kr * Corresponding author

Extended abstract submitted for presentation at the Conference in Emerging Technologies in Transportation Systems (TRC-30)
September 02-03, 2024, Crete, Greece

April 29, 2024

Keywords: Traffic signal coordination, Priority-based optimization, Deep reinforcement learning, Real-world application

1 INTRODUCTION

As urbanization progresses, cities increasingly grapple with traffic congestion, which impedes economic and environmental health. Effective traffic signal control (TSC) strategies are crucial for alleviating these impacts. Traditional methods setting fixed traffic variables, often fail to address complex traffic patterns. In contrast, recent developments in model-free multi-agent deep reinforcement learning (MARL) provide dynamic solutions that adapt to traffic changes, showing potential in traffic management Wei *et al.* (2019), Zeng *et al.* (2022).

While MARL shows promise for TSC, it faces key challenges. The first one is the misalignment between the RL agent's immediate goal and the broader TSC objective. The primary aim of TSC is to improve overall traffic conditions by reducing network delay or total travel time. Previous researches Wei *et al.* (2019), Zeng *et al.* (2022) have used local intersection rewards like queue length or pressure to optimize network performance. However, these approaches can lead to biased results or localized optimization peaks Zhang *et al.* (2020).

Another challenge is the real-world applicability of these methods. Most RL-based strategies Wei *et al.* (2019) adopt an adaptive control strategy that decides whether to change the current phase at every short interval. Although such methods demonstrate timely and flexible control capabilities, they currently face high transmission costs and the risk of traffic accidents, considering real-world deployment Zeng *et al.* (2022).

To address the aforementioned challenges, we propose a novel RL-based strategy using the Cross-Entropy Method (CEM) to hierarchically optimize traffic signal variables: offset and phase split. This approach prioritizes control targets sequentially by Level of Service (LoS), starting with offset control at the upper level, followed by phase split optimization at the lower level. In each level, control targets are optimized sequentially, starting with the highest priority target first, followed by the next in line. Unlike prior studies, this method integrates a global reward, specifically network delay time, which aligns local actions with the broader traffic management goals, thus eliminating the gap between local rewards and the ultimate goal of TSC. By controlling the traffic signal variables, offset and phase split, the proposed method can be seamlessly applied to the real network. To align with existing standard transition policy for offset control from the National Police Agency in South Korea, we regulate the amount of offset adjusted each transition cycle.

2 PRELIMINARY

2.1 Cross-entropy method (CEM)

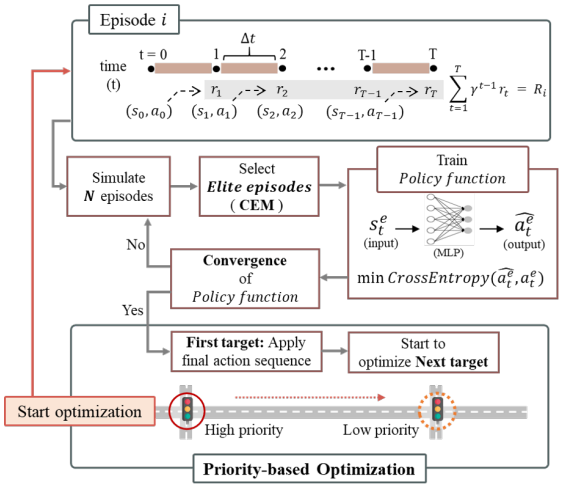
The Cross-Entropy Method (CEM) is a model-free, policy-based reinforcement learning algorithm designed for optimizing decision-making processes in both continuous and discrete action spaces. The method involves refining a probability distribution over actions, $q(a)$, to maximize the expected cumulative reward expressed as $J(\pi) = E_\pi[\sum_t^T \gamma^{t-1} r_t]$, where γ is the discount factor, r_t is the reward at time t, and $\pi(a|s)$ is the policy dictating the action a in state s.

CEM operates by sampling action sequences from $q(a)$, evaluating them based on their generated rewards, and updating $q(a)$ towards actions that yield higher rewards. This update process is iteratively performed by selecting elite actions that performed best and adjusting $q(a)$ to increase the likelihood of these elite actions in future samplings, as outlined in $q'(a_t^e|s_t^e) = q(a_t^e|s_t^e) + \Delta q_{update}$ where a_t^e and s_t^e are action and state in elite episodes, respectively. The efficacy of CEM relies on a good initial distribution and is enhanced by extensive exploration of the action space. The approach provides a practical framework for the sequential optimization of decision variables.

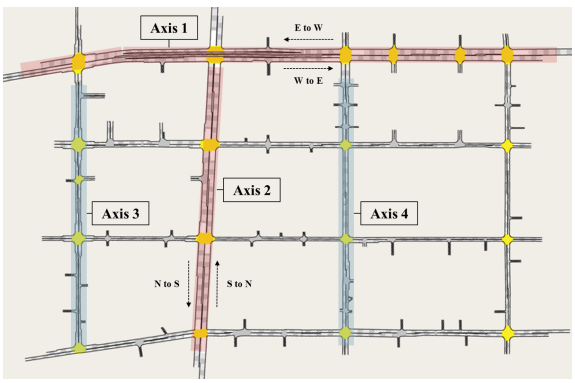
3 METHODOLOGY

3.1 Framework for priority-based optimization

As mentioned earlier, both offset and phase split are subjected to priority-based optimization (Figure 1a). Priority-based optimization assigns higher control priorities to control targets with poorer Levels of Service (LoS), and sequentially optimizes each target one by one, according to priority. The LoS is assessed by the average approaching delay, which is calculated by summing the delay times of each link, weighted by the link flow, and then divided by the total flow.



(a) *Framework for priority-based optimization*



(b) *Control target network and axises for offset control*

Figure 1 – *Framework for optimization and target network. The 'number' behind 'Axis' is the priority order for offset control.*

3.1.1 Upper-level: offset control

As depicted in Figure 1, the control target unit for offset control is based on the Axis, with each Axis (subarea, SA) comprising several intersections. Thus, offset control determines the offsets

for the intersections within a control target axis. The higher control priority is assigned to the Axis with poorer LoS. The LoS of an Axis is defined as the averaged LoS of the intersections it contains.

According to the standard specifications for traffic signal controllers issued by the Korean National Police Agency, the transition of offsets is restricted to a maximum of 17% and 33% of the cycle for negative and positive direction transitions respectively, with positive direction transitions being the default. We perform the offset control in accordance with these transition standards, typically completing within three cycles. Offset control is performed once every hour.

3.1.2 Lower-level: phase split optimization

After the completion of the upper-level offset control, the lower-level phase split optimization is conducted. The control target unit for phase split is the individual intersection, and priority-based optimization is performed for each intersection based on its LoS. Phase split adjustments are made every 15 minutes.

3.2 Markov decision process (MDP)

Offset control and phase split optimization utilize the same state and reward. In this study, the ***state*** is defined as the average queue length over the previous time step (Δt) at the incoming links of target and neighbor intersections, $s_t = [\bar{q}^i_{n,\Delta t}, \bar{q}^i_{s,\Delta t}, \bar{q}^i_{e,\Delta t}, \bar{q}^i_{w,\Delta t}]$. As an exception, at the beginning of an episode where no Δt exists, the queue at $t = 0$ is taken as the state.

For the ***reward***, we adopt the network delay time as a global reward that is the average delay time per vehicle per kilometre, $r_t = -D_{network,\Delta t}$. The objective of TSC is to improve the network efficiency by minimizing the average travel time or the average delay of network. In this respect, using network delay time as a global reward will make it easier to reach the optimal solution.

Regarding the ***action*** in *offset control*, it involves combining the offsets of intersections along the same axis. For example, if there are three intersections on an axis, an action could be represented as [10, 30, 60], where each offset is adjusted in 10 units ($T_{\text{offset}} \in [0, T_c]$), and T_c denotes the cycle time. For the *phase split* in this study, the action is the proportion of the remaining green time to allocate, $a_t = [prop_i],\ where\ i \in \{1, ..., n\}$, $prop_i \in \{0, 5, 10, ..., 95\}$, $\sum_i prop_i = 100(\%)$. The remaining green time is returned by subtracting the minimum green time of each phase (p_i) from the cycle length. We distribute a certain proportion of this remaining green time to each phase constituting a cycle. These two actions are defined in a discrete action space.

4 EXPERIMENTS

4.1 Dataset, compared methods and evaluation metrics

We evaluated the proposed method using the AIMSUN simulator on a 4x4 grid intersection network in Bucheon city, South Korea, during AM peak time from 8 to 9 a.m. on September 9, 2021.

Our approach was compared against both traditional (COSMOS (Cycle, Offset, Split Model for Seoul)) and RL-based methods (IQN(Independent Queue Learning), and PressLight). "Ours-offset" controls only offset under our framework, whereas "Ours" includes both offset and phase split optimizations.

Performance metrics, detailed in Table 1, include the global ones of average delay time, speed, queue, and # of stops. Additionally, local metrics such as space mean speed and average approaching delay (LoS) are evaluated to assess the impacts of our method on traffic flow in more detailed view.

Table 1 – *Numerical statistics with different evaluation metrics*

	Global Metrics				Space Mean Speed				Avg. Approaching Delay (LoS)			
	Delay time (sec/km)	Speed (km/h)	Queue (veh)	# of stops	Axis1 (W2E, E2W)	Axis2 (N2S, S2N)	Axis3 (N2S, S2N)	Axis4 (N2S, S2N)	Axis1	Axis2	Axis3	Axis4
COSMOS	198.52	17.71	1,622.54	66,523	(12.69, 15.26)	(13.29, 10.46)	(21.84, 6.66)	(12.46, 14.65)	98.87	65.14	46.24	37.55
IQL	216.92	**26.36**	3,591.84	**39,255**	(15.77, 14.21)	(15.06, 13.41)	(**30.19**, **19.73**)	(14.35, **19.30**)	69.33	**11.37**	**36.35**	79.09
PressLight	186.54	19.09	1,535.66	71,983	(18.81, 14.93)	(12.94, **14.08**)	(21.38, 7.24)	(**14.83**, 15.84)	77.62	49.81	38.77	**34.36**
Ours-offset	176.39	18.86	1,275.46	63,432	(18.09, 16.70)	(14.96, 12.59)	(22.24, 6.63)	(14.28, 14.97)	68.95	49.24	39.11	37.75
Ours	**162.21**	20.27	**1,207.46**	61,363	(**21.93**, **17.39**)	(**17.35**, 12.60)	(21.29, 8.33)	(11.51, 14.57)	**54.23**	50.52	38.33	38.44

4.2 Performance evaluation

As detailed in Table 1, the proposed method consistently outperforms other strategies across key global metrics, demonstrating its robustness in managing complex traffic scenarios. While the IQL algorithm excels in average speed and number of stops, its performance in terms of average delay time and queue length is sub-optimal. This discrepancy is particularly notable in its management of the LoS, where it significantly enhances LoS for Axis2 and Axis3, but at the cost of deteriorating conditions on Axis4. This indicates that while IQL can optimize specific segments of traffic, it may do so by shifting rather than alleviating congestion.

In contrast, the proposed method achieves a more equitable balance across all axes, suggesting a superior performance to distribute traffic management benefits more uniformly. This is achieved by a strategic optimization that does not favor one axis over others, thus preventing the transfer of congestion from one part of the network to another. The reduced variance in LoS across the axes underlines the our method's effectiveness in ensuring consistent service levels throughout the network.

This balanced optimization is largely attributed to the incorporation of a global reward that focuses on reducing network delay, and overall network health is a pivotal aspect of our proposed method. This strategic approach ensures that the system is not only optimized for immediate, localized benefits but also contribute to the systemic improvement of traffic conditions across the entire network. Furthermore, while "Ours-offset" effectively manages offset controls, the integration of phase split optimizations in "Ours" provides a more thorough and balanced traffic management solution, making it ideal for managing traffic in urban settings with complex dynamics.

5 ACKNOWLEDGEMENTS

This work was supported by Korea Institute of Police Technology (KIPoT) grant funded by the Korea government (KNPA) (092021C29S01000, Development of Traffic Congestion Management System for Urban Network)

References

Wei, Hua, Chen, Chacha, Zheng, Guanjie, Wu, Kan, Gayah, Vikash, Xu, Kai, & Li, Zhenhui. 2019. Presslight: Learning max pressure control to coordinate traffic signals in arterial network. *Pages 1290–1298 of: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining.*

Zeng, Jing, Xin, Jie, Cong, Ya, Zhu, Jiancong, Zhang, Yihao, Jiang, Weihao, & Pu, Shiliang. 2022. Halight: Hierarchical deep reinforcement learning for cooperative arterial traffic signal control with cycle strategy. *Pages 479–485 of: 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC).* IEEE.

Zhang, Huichu, Kafouros, Markos, & Yu, Yong. 2020. PlanLight: learning to optimize traffic signal control with planning and iterative policy improvement. *IEEE Access*, **8**, 219244–219255.