Available online at www.sciencedirect.com
ScienceDirect
ICT Express 9 (2023) 776–782
www.elsevier.com/locate/icte

# Applying multi-agent deep reinforcement learning for contention window optimization to enhance wireless network performance

Chih-Heng Ke[a], Lia Astuti[b,*]

[a] Department of Computer Science and Information Engineering, National Quemoy University, Kinmen, 892, Taiwan
[b] Master Program of Information Technology and Applications, National Quemoy University, Kinmen, 892, Taiwan

Received 12 April 2022; received in revised form 1 July 2022; accepted 27 July 2022
Available online 10 August 2022

**Abstract**

This paper investigates the Contention Window (CW) optimization problem in multi-agent scenarios, where the fully cooperative among mobile stations is considered. A partially observable environment is employed to model and analyze the CW optimization problem, and Smart Exponential-Threshold-Linear with Deep $Q$-learning Network (SETL-DQN) Multi-Agent (MA) algorithm is proposed to obtain the optimal system throughput through the CW Threshold optimization. In the determined scenarios, SETL-DQN(MA) can effectively cope with the mutual interaction among mobile stations. The simulation results show that our proposed method is superior from both static and dynamic scenarios and has the highest optimum packet transmission efficiency.

*Keywords:* CW optimization; CW threshold; SETL-DQN multi-agent; System throughput

## 1. Introduction

The increasing number of connected devices in wireless networks results in the throughput degradation. Meanwhile, optimizing the throughput is essential, e.g., research in [1] optimizes the system throughput by considering the power consumption of degree satisfaction in machine-to-machine devices. Improving the throughput performance in wireless networks means reducing the collision rates. As a basic method to prevent the collision, Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) is realized by Binary Exponential Backoff (BEB) algorithm [2]. Each station waits a certain random number of time slots for data transmission before accessing the channel, determined by the Contention Window (CW). The CW optimization significantly impacts network performance in CSMA/CA [3]. In recent decades, various methods have been developed to select the appropriate CW size [4–6].

Reinforcement Learning (RL) [7], as a field of Machine Learning (ML), had a massive impact on addressing network issues successfully. In the Refs. [8,9], the authors investigated RL agents to dynamically analyze the problem in Unmanned Aerial Vehicle (UAV) assisted communication. Moreover, the $Q$-learning algorithm is one of the classic techniques of RL and also a suitable paradigm to model and analyze CW optimization [10,11]. Unfortunately, its drawback is costly for the agent in the early stages of learning since each state–action pair must be entered consistently to converge on the best policy. In these scenarios, a Deep $Q$-Learning (DQL) is an extension of $Q$-learning, which is a typical Deep Reinforcement Learning (DRL) method [12]. Instead of using a $Q$-table, the DQL technique uses a Deep $Q$-Network (DQN), where the network is from Deep Neural Network (DNN). Then, based on the DQN method, the CW optimization problem in a single-agent scenario was investigated in [13–15].

The authors [16–18] adopted a multi-agent scenario to improve the wireless network performance. Markov Decision Process (MDP) was formulated on the anti-jamming problem in [16,17]. Yet, since stations cannot observe the network thoroughly but are part of it, a Partially Observable Markov Decision Process (POMDP) model can be utilized for multi-agent DRL in [18]. However, the fully cooperative among stations for the CW optimization in wireless networks were not considered. Based on our previous work [15], the single-agent has been investigated to enhance the network performance through CW optimization. We use the proposed scheme, called Smart Exponential-Threshold-Linear (SETL) in [6], with the DQN (SETL-DQN) backoff algorithm in [15], which aims

* Corresponding author.
*E-mail addresses:* smallko@gmail.com (C.-H. Ke), liaastutii08@gmail.com (L. Astuti).

https://doi.org/10.1016/j.icte.2022.07.009
2405-9595/© 2022 The Author(s). Published by Elsevier B.V. on behalf of The Korean Institute of Communications and Information Sciences. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

**Table 1**
The tuples explanation of Decentralized Partially Observable Markov Decision Process (Dec-POMDP).

| Tuple | Meaning |
|---|---|
| $I$ | The finite set of agents |
| $\mathcal{S}$ | The set of states |
| $\mathcal{A}$ | The set of joint actions for each agent |
| $\mathcal{T}$ | The set of transition probabilities regarding transition between states |
| $\Omega$ | The set of joint observations |
| $\mathcal{O}$ | The set of observation probabilities |
| $\mathcal{R}$ | The reward functions |

to have better performance than Centralized Contention Window Optimization with DRL (CCOD) using DQN (CCOD-DQN) [14]. The appropriate CW Threshold ($CW_{Threshold}$) value is defined in the exploration phase through the DQN algorithm in the smaller or bigger contending stations to enhance the packet transmission efficiency.

In order to achieve better network performance, we consider multi-agent domains that are fully cooperative and partially observable environments. On the one hand, through fully co-operated learning, all mobile stations attempt to maximize the system throughput of joint rewards. On the other hand, through a partially observable environment, the setup framing problem optimization is required to represent the mobile station's decision process. To analyze the packet transmission efficiency through the $CW_{Threshold}$ optimization problem in a multi-agent scenario, Decentralized POMDP (Dec-POMDP) is adopted, and SETL-DQN Multi-Agent (MA) algorithm is proposed. The main contributions of this paper are given as follows:

- Based on Dec-POMDP, the packet transmission efficiency through the $CW_{Threshold}$ optimization is investigated in multi-agent scenario, and fully cooperative among mobile stations is considered.
- We develop SETL-DQN(MA) algorithm to improve the network performance in the static and dynamic topologies.

In Section 2, we explain the system model and problem formulation. Section 3 will provide our proposed algorithm. Section 4 will describe the numerical results and discussion. Finally, Section 5 will present the conclusion of this paper.

## 2. System model and problem formulation

### 2.1. System model

The investigated model has $I$ mobile stations and one access point (AP). Each mobile station has intelligent capabilities, such as observing the state, taking action, decision making, and thus. Since all mobile stations transmit at the same frequency, there is just one AP shared by all mobile stations. The mobile station set is donated as $I = 1, 2, \dots, I$. This study will be used the saturation mode which considers that the packet will always be available and transmitted by the mobile stations.

**Channel state mechanism** The station must observe the channel for the whole backoff time to acquire accurate channel status information. When two or more stations send their data at the same time slot, the packets will collide. The delivery packets will be successful if just one station sends. The station performs the backoff method to detect an idle channel.

Our proposed backoff method is based on a Smart Exponential-Threshold-Linear (SETL) backoff algorithm using the $CW_{Threshold}$ value [6]. Furthermore, the $CW_{Threshold}$ is a fixed value, $CW_{Threshold} = 512$. The CW value will increase, and its increment will be decided by the $CW_{Threshold}$ value. When the CW value is smaller than $CW_{Threshold}$ value. The current CW value at unsuccessful packet transmission will be doubled, and the current CW value at successful packet transmission will be halved from the original CW value. In addition, when the CW value is bigger than $CW_{Threshold}$ value. The current CW value at unsuccessful packet transmissions will add 32 each time, and the current CW value at the successful packet transmission will subtract 32 each time. Nevertheless, our previous work [15] proposed the SETL-DQN algorithm and set different values of $CW_{Threshold}$ value under various network scenarios in the exploration training phase, where the Access Point (AP) will broadcast the $CW_{Threshold}$ value to each station via beacon, and each station will use this $CW_{Threshold}$ value to update its CW value based on the wireless traffic load whether in the light or heavy network loads.

### 2.2. Problem formulation

In this paper, we assume the mobile stations must learn fully cooperative behavior only from their observations. Motivated by [19], the SETL-DQN(MA) algorithm can be formulated by a Dec-POMDP, which is the extension of the MDP use in a multi-agent scenario. Mathematically, it can be expressed as $\mathcal{P} = (I, \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \Omega, \mathcal{O})$ tuple, see Table 1 for the explanations.

We will describe each element of the tuples as follows:

At each timestep $t$, each *agent* $i \in I$ located in each mobile station, where $I = 1, \dots, i, \dots, I$ is a finite set of agents.

The *state* $s_t \in \mathcal{S}$ is the current situation of the exact status of the currently connected devices to the wireless networks that observed by each agent.

The *joint actions* $a_t \in \mathcal{A}$ of each agent set the appropriate $CW_{Threshold}$ value, $CW_{Threshold} = 2^7 \times (1 + a)$, with the adjustment of action value is an integer action $a$ between 0 and 7, and the range of the new $CW_{Threshold}$ value is from 128 to 1024. Each decision $a$ regarding the new $CW_{Threshold}$ value taken by each agent $i$, when the joint action $a_t = \langle a_t^1, \dots, a_t^i \rangle$ is executed, the environment transit from $s_t$ to $s_{t+1}$ with a *set transition probability* $p(s_{t+1}|s_t, a_t) \in \mathcal{T}$.

The set of *joint observations* $P_{col} \in \Omega$ is the finite set of joint observations, where $P_{col}$ is the collision probability.

The set of *observation probabilities* $H_{(P_{col})} \in \mathcal{O}$ is when the agent observes the previous and current history of recently observed collision probabilities $H_{(P_{col})}$.

The *reward functions* $r \in \mathcal{R}$ are obtained after the action performed by each agent to adjust the new $CW_{Threshold}$ value. The main objective of each agent is to obtain its throughput by maximizing the total system throughput and piggyback the throughput to the AP. The AP will sum up all the throughputs received by all mobile stations and send it back to each mobile station via beacon. Each mobile station will use the total throughput as a reward by finding a deterministic joint policy $\pi = \langle \pi^1, \ldots, \pi^i \rangle$. Moreover, the agent $i$ learns an optimal control policy $\pi_i$ to maximize the cumulative discounted future reward at the time $t_0$, it shows in Eq. (1), where $\gamma \in [1, 0]$ is a discount factor that weights the importance of rewards.

$$\mathcal{R}_t = \sum_{t=t_0}^{\infty} \gamma^{\tau} r_{t+\tau+1} \tag{1}$$

But the complexity here is how the action-value of a given policy $\pi$ represents the utility of action $a$ at state $s$. In the next section, we propose a multi-agent DRL scheme to solve this problem formulation.

## 3. Proposed scheme

Each mobile station has been given an agent in the proposed scheme, using the multi-agent DRL to interact among mobile stations and obtain the total system throughput. All of the mobile stations' agents have access to the same policy network, and the saturation mode is used. We consider fully cooperative scenarios in the static and dynamic topologies. Relay and information exchange in wireless networks will be considered between all mobile stations based on the observation of the correlated state [20]. Referring to [16], the "decision-feedback adjustment" approach is used to optimize packet transmission efficiency based on mobile station interaction.

To solve the formulated optimum cumulative discounted future reward for objective in Eq. (1), the SETL-DQN(MA) backoff algorithm is proposed in this work to enhance the wireless network performance, where adopting in [15] through applying the appropriate $CW_{Threshold}$ value according to the current traffic condition. We defined the $Q$-function associated with a policy as the expected reward when action is performed in a given state, i.e., $Q^{\pi}(s,a) = \mathbb{E}_{\pi}[R_t | s_t = s, a_t = a]$.

The $Q$-function satisfies the Bellman equation [21], as an action-value function:

$$Q^{\pi}(s,a) = \mathcal{R}(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathcal{T}_{ss'}^{a} \left( \sum_{a' \in \mathcal{A}} \pi(s',a') Q^{\pi}(s',a') \right) \tag{2}$$

where $\mathcal{R}(s,a) = \mathbb{E}_{\pi}[\mathcal{R}_{t+1} | s_t = s, a_t = a]$ is the expected reward of taking action $a$ at state $s$. $\mathcal{T}_{ss_{t+1}}^{a}$ is the transition probability from given state $s$ to the state $s'$ with action $a$. Similar to [16], the mobile station $i$ updates its $Q$-values according to Eq. (3), where λ is a learning rate.

$$q(s_t, a_t) \leftarrow (1-\alpha) q(s_t, a_t) + \lambda \left( r_{t+1} + \gamma max_{a_{t+1}} (s_{t+1}, a_{t+1}) \right) \tag{3}$$

However, our scenarios used large state and action space, where the classical $Q$-learning fails due to the rarely visited of the many states. In addition, similar with [22], we will use DQN to remedy this problem and expressed as $q(s, a, \boldsymbol{\theta})$, where the parameter is represented as a real-valued vector $\boldsymbol{\theta}$. The agent $i$ forms a dataset $\boldsymbol{D}$ by collecting an experience replay until time $t$ in the form of $(s, a, r', s')$. We define two DQNs: the target DQN, which has the parameters $\boldsymbol{\theta}_t^{target}$, and the train DQN, which has the parameters $\boldsymbol{\theta}_t^{train}$. $\boldsymbol{\theta}_t^{target}$ is adjusted to be the same as $\boldsymbol{\theta}_t^{train}$. The learning update at each iteration $i$ uses a loss function to train DQN for a random mini-batch $\boldsymbol{D}_t$ at time $t$, based on:

$$L\left(\boldsymbol{\theta}_t^{train}\right) = \sum_{(s,a,r',s') \in D_t} (\mathcal{Y}_t^{DQN}(r', s') - q(s, a; \boldsymbol{\theta}_t^{train}))^2 \tag{4}$$

where the target is:

$$\mathcal{Y}_t^{DQN}(r', s') = r' + \lambda max_{a'} q(s', a'; \boldsymbol{\theta}_t^{target}) \tag{5}$$

The state transitions of their common environment in a multi-agent learning system are dependent on the agents' joint actions as the output of the training meta-agent via DQN.

The joint action can be given by Eq. (6).

$$a \in arg\, \underset{\boldsymbol{a'}}{max} \sum_{i=1}^{I} q(s', a'; \boldsymbol{\theta}_t^{target}) \tag{6}$$

In this work, each mobile station $i$ has the same copy of the DQN with the parameters $Q_t^{target}$ at the time slot.

The exchanged information between mobile stations was able to send and receive frames. Referring to [7], we balance between exploration and exploitation using $\varepsilon$-greedy with $0 \leq \varepsilon(t) \leq 1$. The mobile station acts randomly with probability $\varepsilon$ and according to current policy with probability $1 - \varepsilon(t)$. Based on the aforementioned analysis, the SETL-DQN(MA) backoff algorithm is proposed. In one time slot, each mobile station has its own agent and uses the collision probability $P_{col}$ as the finite set of joint observations, where $P_{col}$ can be calculated as follows:

$$P_{col} = \frac{Packet\ collision}{Packet\ collision + Packet\ succesful} \tag{7}$$

where each mobile station knows whether the transmission of the packet is unsuccessful or successful.

**Algorithm 1**: A fully cooperative of Smart Exponential-Threshold-Linear with Deep Q-learning Network (DQN) Multi-Agent (MA)

```
1: Initialize replay memory D and Q-network Q with random weights θ
2: for episode=1 to T do
3:     Each agent observes its current state $\mathcal{S}$, and selects the new $CW_{Threshold}$ value according to the following rules:
         • Select a random action $a_t$ with probability $\varepsilon$, otherwise select $a_t$ based on Eq. (6)
         • Use the collision probability $P_{col}$ as the finite set of joint observations based on Eq. (7)
         • Observe the previous and current history of recently observed collision probabilities $H_{(P_{col})}$
         • Generate the new $CW_{Threshold} = 128 \times (1 + a)$
         • Form $< s_t, a_t, r_t, s_{t+1} >$ in D
4:     Each agent working collaboratively to maximize the system throughput of joint rewards
5:     The state is transferred into $p(s_{t+1}|s_t, a_t) \in \mathcal{T}$
6:     The weights of the neural network then optimized by using Adam optimizer with respect to the network parameter $\boldsymbol{\theta}$ to minimize the loss at Eq. (4)
7: end for
```

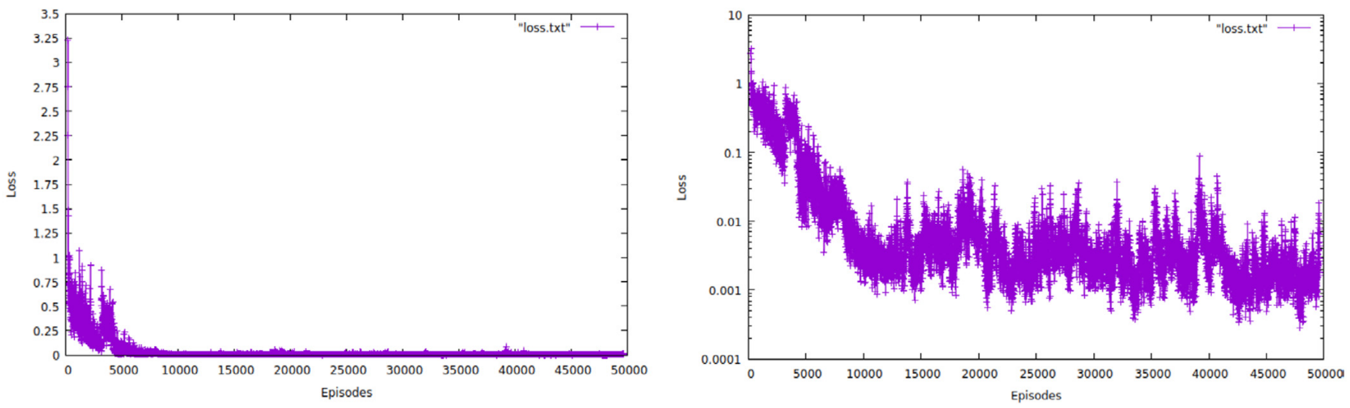

**Fig. 1.** A variation of train loss/error with number of episodes.

The agent in each mobile station $i$ observes the previous and current of $H_{(P_{col})}$ to estimate the influence of its own and its neighbor's joint actions. With the help of DQN in the multi-agent scenario, the agent determines its own $CW_{Threshold}$ based on collision probability history (observations) and the total system throughput (reward), and selects the CW value based on the rule of the $CW_{Threshold}$ changes. When the station has unsuccessful packet transmission, if the CW value is lower than the $CW_{Threshold}$, the CW value will be doubled, and if the CW value is larger than the $CW_{Threshold}$, the CW value will add 32 each time. Since the mobile stations are working collaboratively to maximize the total system throughput, the mobile stations will piggyback their own throughput to the AP. The AP will sum up all mobile stations' throughput and send the total throughput back to each mobile station's rewards. Finally, based on the above analysis, a fully cooperative Smart Exponential-Threshold-Linear with Deep Q-learning Network (SETL-DQN) Multi-Agent (MA) is proposed, and its implementation procedure is shown in Algorithm 1.

## 4. Numerical results and discussions

The Medium Access Control (MAC) method communication standard 802.11ac has been simulated with the AP periodically broadcasting packets. The simulation was performed using Parl and Paddle–Paddle provided by Baidu [23] and using the python language. Refer to [24], we set the experiment parameters of IEEE 802.11ac, which has: using VHT (Very high throughput) and maximum MDPU length, 3895 bytes of the packet payload, 128 bits of PHY header, ACK is 112 bits + PHY header, 9 µs of the slot time ( $\sigma$), DIFS is 34 µs, SIFS is 16 µs, propagation delay is 1 µs, with $CW_{min}$ = 16, and $CW_{max}$ = 1024. The simulation environment was tested in the Linux operation system, with the ubuntu system, 18.04 version, 8 processors cores, and 6MB memory. Because the accessible APs do not have GPUs, training should be done before the agent is installed on the AP in real-world deployments.

There are train and evaluation phases. In train mode there is more exploration and in evaluation mode we do not update the DNN parameters anymore. We set the number of episodes $T$ = 50,000, 20,000 of memory size, 32 samples of batch size, 0.001 of learning rate ($\lambda$), 0.99 of discount factor ($\gamma$), of $\varepsilon$-greedy with 1e-6 decrement. The proposed SETL-DQN(MA) algorithm uses Adam optimizer algorithm, the network structure consists of three DNN hidden layers with 128 fully connected networks, and rectified linear unit (ReLU) used as a layer activation function.

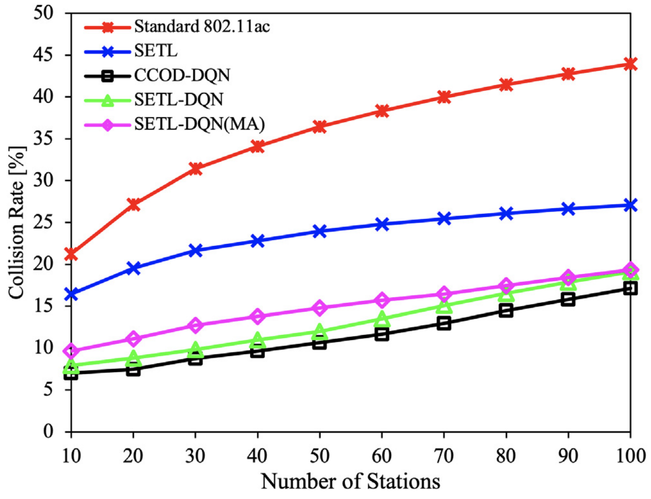

**Fig. 2.** The collision rate comparison for static topology.

Before we investigate how good of the proposed SETL-DQN(MA) algorithm under various network conditions. Fig. 1. shows the loss function result to measures model errors of the network training analysis. The general train loss decreases in every episode and approaches the value 0. In the whole training process, starting from around 8,000 episodes, the train loss is stable which around 0.001 to 0.01. This means our proposed SETL-DQN(MA) algorithm can adapt well to the complexity and convergence during the network training process.

To investigate the various performance aspects of the SETL-DQN(MA) method, it was tested in two main topologies: static and dynamic topologies. Note that in all figures where the results run multiple times in order to get the desired results. To evaluate the simulation performance, this paper will compare our proposed method; SETL-DQN(MA) algorithm

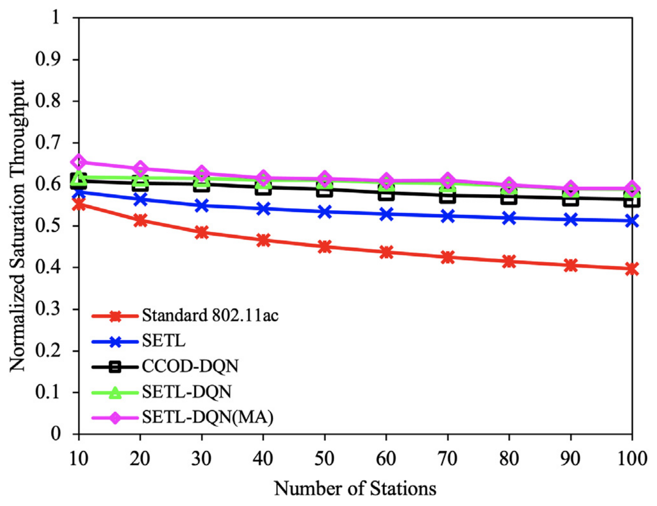

**Fig. 3.** The normalized saturation throughput comparison for static topology.

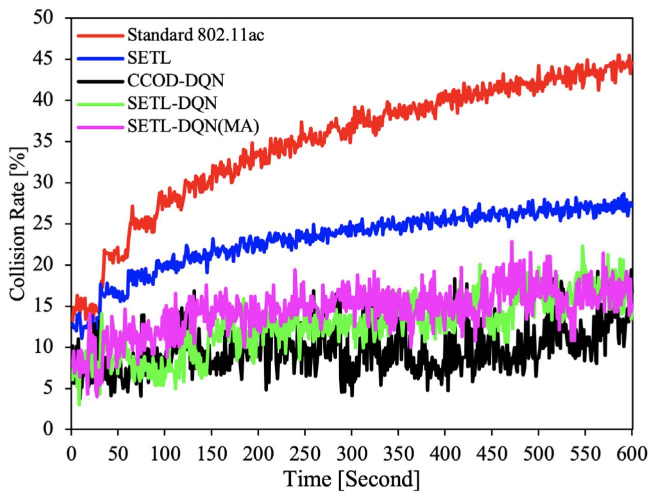

**Fig. 4.** The collision rate comparison for dynamic topology (from 5 to 100 stations).

with, SETL-DQN [12], CCOD-DQN [11], SETL [5], and Standard 802.11ac.

### 4.1. Static topology

Throughout the experiment, a fixed number of stations were connected to the AP for each data point in a static topology. It has been set from 10 until 100 stations, with 10 intervals. Meanwhile, the SETL and Standard 802.11ac are non-learning methods. Fig. 2 shows the results that the standard 802.11ac and SETL performance degrade for larger network loads, yet, SETL has a better result than standard 802.11ac. Similar to our previous work [15], CCOD-DQN has a better collision rate than SETL-DQN in single-agent scenarios. In this current work, the fully cooperative mobile stations on the SETL-DQN(MA) algorithm have a slightly higher collision rate than both CCOD-DQN and SETL-DQN.

Fig. 3 presents the normalized saturation throughput comparison. From those five algorithms, SETL-DQN(MA) has

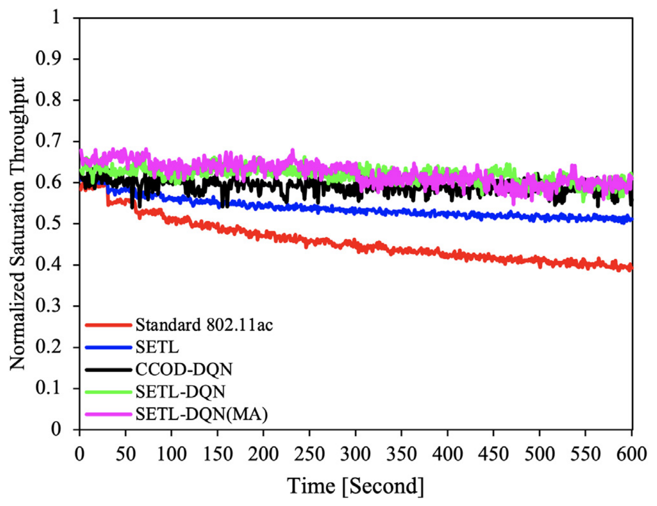

**Fig. 5.** The normalized saturation throughput for dynamic topology (from 5 to 100 stations).

superior performance with the help of a fully collaborative multi-agents DQN algorithm, where all mobile stations obtain the system throughput from the AP. Compared with the other four algorithms, our proposed algorithm in the multi-agent scenario gives the best-normalized saturation throughput, especially from 10 to 30 stations, which in the light network load; this optimization is due to the cumulative rewards learned among mobile stations to interact with each other in the partially observable environment.

### 4.2. Dynamic topology

The number of transmitting stations continuously rose from 5 to 100 stations with 5 intervals in the dynamic topology. It continues with the time changes every 30 s until 600 s for every 5 added stations intervals over the course of a single experiment. This scenario is designed to test whether the proposed algorithm is capable of reacting to the keep-changing network environment. Fig. 4 shows how the number of stations increased in a simulation run and how the collision rate was updated accordingly. Overall, the result showed that the instantaneous collision rate for the SETL-DQN(MA) algorithm is slightly higher than CCOD-DQN and SETL-DQN algorithms.

The DQN strongly influences the way to increase the throughput performance. In Fig. 5, standard 802.11ac leads to a decrease of up to 24.6% of the normalized saturation throughput following the increasing number of stations. The SETL-DQN(MA) is able to maintain the efficiency than SETL-DQN and CCOD-DQN with 0%–2% differences. Ultimately, the SETL-DQN(MA) still has a slightly higher collision rate than SETL-DQN and CCOD-DQN (Fig. 6). Otherwise, the operation of the SETL-DQN(MA) algorithm in the dynamic topology leads to improved network performance (Fig. 7), wherein the 10 stations, the SETL-DQN(MA) has a better 1.87% than SETL-DQN. However, in the 100 stations, the result of SETL-DQN(MA) (has 0.598) is slightly lower than SETL-DQN (has 0.608). Therefore, from the

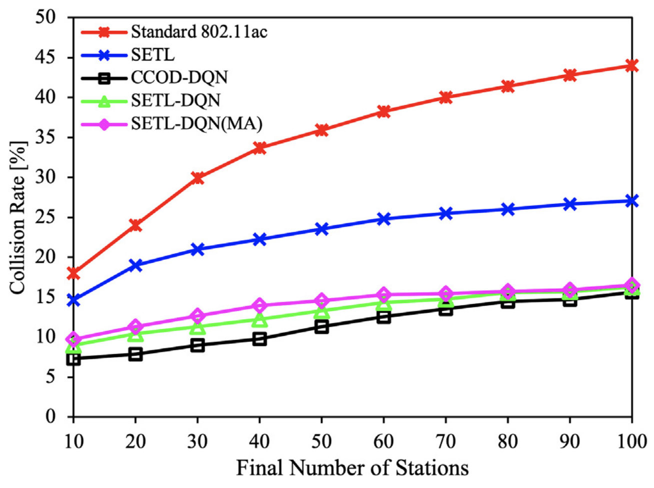

**Fig. 6.** The collision rate comparison for dynamic topology.

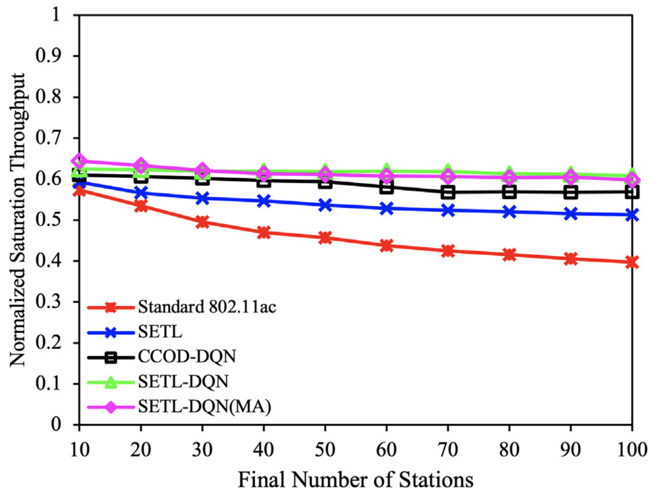

**Fig. 7.** The normalized saturation throughput comparison for dynamic topology.

aforementioned result analysis, the proposed SETL-DQN(MA) algorithm has well scalability and robustness in small and large scales network load systems.

## 5. Conclusion

In this paper, we consider the fully cooperative and partially observable environment. The Smart Exponential-Threshold-Linear with Deep $Q$-learning Network (SETL-DQN) Multi-Agent (MA) scenario is proposed to obtain better network performance. Finally, SETL-DQN(MA) algorithm succeeded in exceeding the normalized saturation throughput in both static and dynamic topologies than the other four algorithms. In the collision rate comparison, the SETL-DQN(MA) algorithm is slightly higher than both SETL-DQN and CCOD-DQN in the single-agent scenario. However, it still can cope to optimize the packet transmission efficiency via the $CW_{Threshold}$ value optimization. The $CW_{Threshold}$ optimization is applicable universally under various wireless networks scenarios. In future works, the SETL-DQN single-agent and multi-agent also can be applied with more realistic network conditions to optimize the multi-rate transmission in the wireless environment.

## CRediT authorship contribution statement

**Chih-Heng Ke:** Conceptualization, Methodology, Software, Validation, Formal analysis, Supervision, Project administration, Funding acquisition. **Lia Astuti:** Methodology, Software, Validation, Formal analysis, Data curation, Writing – Original Draft, Writing – review & editing, Visualization.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

[1] E.E. Tsiropoulou, G. Mitsis, S. Papavassiliou, Interest-aware energy collection & resource management in machine-to-machine communications, Ad Hoc Netw. 68 (2018) 48–57.
[2] W.K. Kuo, C.-C.J. Kuo, Enhanced backoff scheme in CSMA/CA for IEEE 802.11, in: Proceeding of IEEE 58th Vehicular Technology Conference, Vol. 5, VTC, 2003, pp. 2809–2813.
[3] V.K. Garg, Chapter 21 - Wireless local area networks, Wirel. Commun. Netw. (2007) 713–776.
[4] I. Syed, B.H. Roh, Adaptive backoff algorithm for contention window for dense IEEE 802.11 WLANs, Mob. Inf. Syst. (2016).
[5] M. Karaca, S. Bastani, B. Landfeldt, Modifying backoff freezing mechanism to optimize dense IEEE 802.11 networks, IEEE Trans. Veh. Technol. 66 (10) (2017) 9470–9482.
[6] C.H. Ke, C.C. Wei, K.W. Lin, J.W. Ding, A smart exponential-threshold-linear backoff mechanism for IEEE 802.11 WLANs, Int. J. Commun. Syst. 24 (8) (2011) 1033–1048.
[7] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, second ed., MIT Press, Cambridge, MA, 2018.
[8] C. Wang, D. Deng, L. Xu, W. Wang, Resource scheduling based on deep reinforcement learning in UAV assisted emergency communication networks, IEEE Trans. Commun. 70 (6) (2022) 3834–3848.
[9] R. Ding, F. Gao, X.S. Shen, 3D UAV trajectory design and frequency band allocation for energy-efficient and fair communication: a deep reinforcement learning approach, IEEE Trans. Wireless Commun. 19 (12) (2020) 7796–7809.
[10] N. Zerguine, M. Mostefai, Z. Aliouat, Y. Slimani, Intelligent CW selection mechanism based on Q-learning (MISQ), Int. Inf. Eng. Technol. Assoc. 25 (6) (2020) 803–811.
[11] T.-W. Kim, G.-H. Hwang, Performance enhancement of CSMA/CA MAC protocol based on reinforcement learning, J. Inf. Commun. Convergence Eng. 19 (1) (2021) 1–7.
[12] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, S. Petersen, Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529–533.
[13] A. Kumar, G. Verma, C. Rao, A. Swami, S. Segarra, Adaptive contention window design using deep Q-learning, in: Proceeding of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 2021, pp. 4950–4954.
[14] W. Wydmański, S. Szott, Contention window optimization in IEEE 802.11ax networks with deep reinforcement learning, in: Proceeding of IEEE Wireless Communications and Networking Conference, WCNC, 2021, pp. 1–6.
[15] C.H. Ke, L. Astuti, Applying deep reinforcement learning to improve throughput and reduce collision rate in IEEE 802.11 networks, KSII Trans. Internet Inf. Syst. 16 (1) (2022) 334–349.

[16] F. Yao, L. Jia, A collaborative multi-agent reinforcement learning anti-jamming algorithm in wireless networks, IEEE Wirel. Commun. Lett. 8 (4) (2019) 1024–1027.

[17] M.A. Aref, S.K. Jayaweera, A cognitive anti-jamming and interference-avoidance stochastic game, in: Proceeding IEEE 16th International Conference on Cognitive Informatics & Cognitive Computing, ICCI*CC, 2017, pp. 520–527.

[18] S.B. Janiar, V. Pourahmadi, Deep-reinforcement learning for fair distributed dynamic spectrum access in wireless networks, in: Proceeding of IEEE 18th Annual Consumer Communications & Networking Conference, CCNC, 2021, pp. 1–4.

[19] H.R. Lee, T. Lee, Multi-agent reinforcement learning algorithm to solve a partially-observable multi-agent problem in disaster response, European J. Oper. Res. 291 (1) (2021) 296–308.

[20] Y. Xu, J. Wang, Q. Wu, A. Anpalagan, Y. Yao, Opportunistic spectrum access in cognitive radio networks: global optimization using local interaction games, IEEE J. Sel. Top. Sign. Proces. 6 (2) (2012) 180–194.

[21] B. Richard, A markovian decision process, J. Math. Mech. 6 (5) (1957) 679–684.

[22] Y.S. Nasir, D. Guo, Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks, IEEE J. Sel. Areas Commun. 37 (10) (2019) 2239–2250.

[23] M. Sun, B. Jiang, H. Xiong, Z. He, H. Wu, H. Wang, Baidu neural machine translation systems for WMT19, in: Proceeding of the 4th Conf. on Machine Translation, 2, 2019, pp. 374–381.

[24] T. Moriyama, R. Yamamoto, S. Ohzahata, T. Kato, Frame aggregation size determination for IEEE 802.11 ac WLAN considering channel utilization and transfer delay, STA 4(s3) (s4) (2017).