Paper

Dynamic channel bonding in WLANs by hierarchical laser chaos decision maker

Hiroshi Kanemasa1a), Aohan Li1, Yusuke Ito1,
Nicolas Chauvet2, Makoto Naruse2, and Mikio Hasegawa1

1 Department of Electrical Engineering, Tokyo University of Science
6-3-1, Niiyuku, Katsushika-ku, Tokyo 125-8585, Japan
2 Department of Information Physics and Computing, The University of Tokyo
7-3-1, Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

a) h-kanemasa@haselab.ee.kagu.tus.ac.jp

Received July 16, 2021; Revised August 31, 2021; Published January 1, 2022

Abstract: Laser chaos decision-maker has been demonstrated to enable ultrahigh-speed
decision-making in solving multi-armed bandit (MAB) problems in the GHz order. In addition
to recent intensive studies of photonic information processing devices and systems, the pursuit
of novel applications is important, which is also demanded from future technologies, including
Beyond 5G context. In this paper, we examine the applicability to dynamic channel bonding
(DCB), which has been introduced in wireless local area networks (WLANs), and demonstrate
a method for achieving higher data rate transmissions while avoiding interference. First, we
propose a DCB method utilizing laser chaos decision-maker. Second, we design two hierar-
chical trees for decision making, that is, DCB selection. Third, we experimentally implement
our proposed methods in a practical WLAN and confirm its operational ability. We analyze
the parameter of the proposed method and compare the proposed method with conventional
decision-making algorithms of $\epsilon$-greedy and UCB1-tuned. We show that our proposed method
demonstrates better DCB decisions than the other decision-making algorithms. Furthermore,
we demonstrate that in DCB, the design method of the hierarchical trees or the parameter for
the proposed method influences the performance of decision making.

Key Words: decision making, multi-armed bandit (MAB) problem, laser chaos, dynamic
channel bonding (DCB), wireless local area networks (WLANs)

1. Introduction

Ultrahigh-speed decision-maker using laser chaos has been demonstrated for solving multi-armed
bandit (MAB) problems at GHz order [1–4]. The goal of MAB problems is to maximize the total
reward from multiple slot machines in uncertain environments [5]. The inherent temporal dynamics
of chaotic lasers accelerates the solution of MAB problems with two arms [1, 2, 6], followed by its
scalable extensions up to 64 arms [3]. Homma et al. developed a compact (5 mm$^2$ area) and on-chip
photonic decision-maker [4].

The laser chaos decision-maker can be applied to various problems which require fast decision-
making in dynamic and uncertain environments [7–9]. Takeuchi et al. applied laser chaos decision-making to channel selection problems; they experimentally demonstrate autonomous and adaptive dynamic channel selection in IEEE 802.11a wireless LANs (WLAN) [7]. Duan et al. applied laser chaos decision making to the 5G NOMA systems and showed by simulation that it achieves higher throughput faster than conventional algorithms [8]. Uozumi et al. proposed a method of applying laser chaos decision-maker to the beam alignment problems in the MIMO systems and examined its effectiveness [9].

Channel bonding (CB) is one of the important decision-making problems in wireless communication systems. It allows wireless devices to use multiple neighboring channels simultaneously, thus potentially achieving higher communication throughputs [10]. However, this may lead to co-channel interference with other wireless access points (APs) within the bonded channels, which may induce degradation of throughput performances [11, 12]. Therefore, channel bonding bandwidth should be appropriately selected according to channel conditions. To prevent this problem, dynamic CB (DCB) is commonly used. DCB selects bonding bandwidth based on spectrum occupancy over time. In the past, several studies on DCB have been reported [13–17]. In [13, 14], DCB mechanism in IEEE 802.11ac was analyzed. Also, static CB (SCB) and DCB policies are analyzed and evaluated by using a continuous-time Markov network (CTMN) framework [15]. In [15], simulation results showed that DCB performs better in terms of both individual throughput and fairness among WLANs. The study [16] developed an analytical model to study the performance of DCB and found that the DCB in 802.11ac can boost the throughput of multi-channel users while providing a friendly coexistence with single-channel users in high-density WLANs. While [13–15] analytically showed that DCB provides much better performance than SCB, the optimization of DCB performance was not so well investigated. In [17], the interactions which occur in the operation of multiple neighboring WLANs when they use CB was analyzed, and an optimal centralized proportional fair channel allocation algorithm was formulated for WLANs under saturated conditions. However, [17] did not consider the dynamic traffic load change over time. Also, [13–17] are not practical experiment but simulation. Their works assume that all channel conditions are known. In reality, however, channel condition is unknown in advance. While [10] examined design guidelines to decide the best CB bandwidth, it takes a long time to collect data traces before the appropriate CB bandwidth can be estimated. In contrast to these works, we aim to conduct flexibly and appropriately CB decisions in dynamically changing uncertain environments.

In this paper, we propose a DCB method using the laser chaos-based MAB algorithm to achieve an ultrahigh-speed solution of DCB problems. We examine the applicability of the proposed method to the optimization of CB in dynamically changing uncertain environments. First, we design two hierarchical trees for solving DCB problems. Next, we experimentally implement the proposed method to practical IEEE 802.11ac networks. In order to account for dynamic changes in the communication environments, external traffic was generated by three personal computers connected to APs located nearby. Then, we evaluate two hierarchical trees. Also, we analyze the parameter dependencies of the proposed method. Furthermore, we compare our proposed method with other major MAB algorithms known as $\epsilon$-greedy and UCB1-tuned.

The remainder of this paper is as follows. We describe laser chaos decision-maker in Section 2. The system model is then described in Section 3. The proposed DCB method using laser chaos decision-maker is presented in Section 4. We implement and evaluate our proposed method in Section 5. Finally, Section 6 concludes the paper.

2. Laser chaos decision maker

2.1 Principle of decision making using laser chaos

In this subsection, we describe the principle of laser chaos decision-making. The details are found in [1]. Figure 1 shows the application of the laser chaos decision-maker to a two-armed bandit problem. Decision at time $t$ among two slot machines, referred to as slot machines A and B, can be made by comparing the sampled values of the laser chaos time series $s(t)$ with a threshold value $TH(t)$. When the laser chaos time series $s(t)$ is greater than the threshold $TH(t)$, slot machine A is selected.
Fig. 1. Decision making using laser chaos.

Otherwise, that is, \( s(t) \) is less than \( TH(t) \), slot machine B is selected. The threshold \( TH(t) \) is updated based on the reward, which is either ‘win’ or ‘fail’ information of a slot machine play. The reward at time \( t \) is fed back to the threshold adjuster value \( TA(t) \). The threshold value \( TH(t) \) at time \( t \) is given by

\[
TH(t) = k \times \lfloor TA(t) \rfloor
\]

where \( \lfloor TA(t) \rfloor \) is the nearest integer to the threshold adjuster value \( TA(t) \) rounded to zero, and \( k \) is the step width of the threshold adjuster value \( TA(t) \) \[1\]. \( \lfloor TA(t) \rfloor \) takes the values \(-N, \cdots, -1, 0, 1, \cdots, N\) where \( N \) is a natural number. Therefore, the number of the thresholds is given by \( 2^N + 1 \). The range of \( TH(t) \) is limited between \(-kN\) and \( kN \) by setting \( TH(t) = kN \) when \( \lfloor TA(t) \rfloor \) is greater than \( N \) \[1\], as well as \( TH(t) = -kN \) when \( \lfloor TA(t) \rfloor \) is smaller than \(-N \) \[1\]. Threshold adjuster \( TA(t) \), which is updated based on the reward, is defined as

\[
\begin{align*}
TA(t+1) &= -\Delta + \alpha TA(t) \quad \text{slot machine A wins}, \\
TA(t+1) &= +\Delta + \alpha TA(t) \quad \text{slot machine B wins}, \\
TA(t+1) &= +\Omega + \alpha TA(t) \quad \text{slot machine A fails}, \\
TA(t+1) &= -\Omega + \alpha TA(t) \quad \text{slot machine B fails},
\end{align*}
\]

where \( \alpha \) (\( 0 \leq \alpha \leq 1 \)) is a forgetting parameter to reduce the influences of past experiences. \( \Delta \) and \( \Omega \) concern to what extents the threshold adjuster variable should be revised when the action results in the ‘win’ and ‘fail’, respectively. In this study, \( \Delta \) is set as a constant value (i.e., \( \Delta = 1 \)). \( \Omega \) is given by

\[
\Omega = \frac{P_A(t) + P_B(t)}{2 - (P_A(t) + P_B(t))},
\]

where \( P_i \) is the estimated reward probability of slot machine \( i \) at time \( t \), which is expressed as follows.

\[
P_i(t) = \frac{R_i(t)}{N_i(t)},
\]

where \( N_i(t) \) is the number of times of selecting slot machine \( i \) until time \( t \), and \( R_i \) is the number of ‘win’ times by selecting slot machine \( i \) until time \( t \). \( \Delta, \Omega \) and the estimated reward probability \( P_i \) have been considered important in this decision making mechanism \[1, 3, 18, 19\].

2.2 Scalable decision making using laser chaos

To realize scalable decision-making systems applicable to MAB problems with more than two arms, Naruse et al. cascaded multiple two-armed laser chaos bandits in a hierarchical structure \[3\]. MAB problems with up to 64 arms were successfully solved using laser chaos. Figure 2 shows scalable decision making of \( 2^M \)-armed bandit problems. In this section, we describe the method of the decision for each slot machine and the threshold updating scheme.

\( 2^M \) slot machines are distinguished by assigning \( M \)-bit binary code given by \( S_1S_2 \cdots S_M \) with \( S_i(i = 1, \cdots, M) \) being 0 or 1. The slot machine is selected by determining bit by bit from the most significant bit (MSB) to the least significant bit (LSB) in a pipelined manner. For each of the bits, the
decision is made by comparing between the measured chaotic signal level and the designated threshold value.

Firstly, we decide the MSB. Chaotic signal $s(t_1)$ measured at $t = t_1$ is compared with the threshold value $TH_1$. If $s(t_1)$ is less than or equal to $TH_1$, the MSB of the slot machine to be chosen is 1, which we denote as $D_1 = 1$. Otherwise, the MSB is determined to be 0 ($D_1 = 0$).

Secondly, we decide the second-MSB. If $s(t_2)$ measured at $t = t_2$ is compared with the threshold value $TH_{2,1}$. The first number 2 in the suffix means that the threshold value is related to the second-MSB, and the second number 1 in the suffix means that the previous decision was to be 1 ($D_1 = 1$). When $s(t_2) \leq TH_{2,1}$, the second-MSB of the slot machine to be chosen is 1, which we denote as $D_2 = 1$. Otherwise, the MSB is determined to be 0 ($D_2 = 0$).

Finally, we discuss the third-MSB. If $D_1 = 1$ and $D_2 = 1$, that is, $s(t_3) \leq TH_{2,1}$, the chaotic signal $s(t_3)$ measured at $t = t_3$ is compared with the threshold value $TH_{3,1,1}$. The number 3 means that the threshold value is related to the third-MSB. When $s(t_3) \leq TH_{3,1,1}$, the third-MSB of the slot machine to be chosen is 1, which we denote as $D_3 = 1$. Otherwise, the third-MSB is determined to be 0($D_3 = 0$). Until all $M$ bits of information $(D_1, D_2, \ldots, D_M)$ which specify the slot machine have been determined, such threshold comparisons continue.

Each threshold value is given by Eq. (1). Here, the threshold adjuster value $TA_{K,S_1,S_2,\ldots,S_{K-1}}(t)$ corresponds to $TH_{K,S_1,S_2,\ldots,S_{K-1}}(t)$. In a general form, the threshold adjuster value for the $K$-th bit is given as follows.

If the selected slot machine wins, the threshold adjuster value for the $K$-th bit is updated as follows.

\[
\begin{align*}
TA_{K,S_1,S_2,\ldots,S_{K-1}}(t+1) &= +\Delta + \alpha TA_{K,S_1,S_2,\ldots,S_{K-1}}(t) & \text{if } S_K = 0, D_1 = S_1, \ldots, D_{K-1} = S_{K-1}, \\
TA_{K,S_1,S_2,\ldots,S_{K-1}}(t+1) &= -\Delta + \alpha TA_{K,S_1,S_2,\ldots,S_{K-1}}(t) & \text{if } S_K = 1, D_1 = S_1, \ldots, D_{K-1} = S_{K-1},
\end{align*}
\]

where $\alpha$ ($0 \leq \alpha \leq 1$) is a forgetting parameter and $\Delta$ is the reward, which is set as a constant value (i.e., $\Delta = 1$).

If the selected slot machine fails, the threshold adjuster value for the $K$-th bit is updated as follows.

\[
\begin{align*}
TA_{K,S_1,S_2,\ldots,S_{K-1}}(t+1) &= -\Omega_{K,S_1,S_2,\ldots,S_{K-1}} + \alpha TA_{K,S_1,S_2,\ldots,S_{K-1}}(t) & \text{if } S_K = 0, D_1 = S_1, \ldots, D_{K-1} = S_{K-1}, \\
TA_{K,S_1,S_2,\ldots,S_{K-1}}(t+1) &= +\Omega_{K,S_1,S_2,\ldots,S_{K-1}} + \alpha TA_{K,S_1,S_2,\ldots,S_{K-1}}(t) & \text{if } S_K = 1, D_1 = S_1, \ldots, D_{K-1} = S_{K-1},
\end{align*}
\]

where $\Omega_{K,S_1,S_2,\ldots,S_{K-1}}$ attenuates the degree of changes induced in $TA_{K,S_1,S_2,\ldots,S_{K-1}}(t)$ which is given by
Dynamic channel selections using laser chaos in IEEE 802.11a [7].

\[ \Omega_{K,s_1,s_2,\ldots,s_{K-1}} = \frac{P_{S_1,s_2,\ldots,s_{K-1},s_K=0} + P_{S_1,s_2,\ldots,s_{K-1},s_K=1}}{2 - (P_{S_1,s_2,\ldots,s_{K-1},s_K=0} + P_{S_1,s_2,\ldots,s_{K-1},s_K=1})}, \tag{7} \]

where \( P_{S_1,s_2,\ldots,s_{K-1},s_K=1} \) is the estimated reward probability of the slot machine. Let the number of times of the \( K \)th bits are selected be denoted by \( F_{S_1,s_2,\ldots,s_{K-1},s_K} \). Let the number of times of ‘wins’ when the machines was selected is given by \( L_{S_1,s_2,\ldots,s_{K-1},s_K} \). Similar to Eq. (4), the estimated reward probability of slot machines is as follows.

\[ P_{S_1,s_2,\ldots,s_{K-1},s_K} = \frac{L_{S_1,s_2,\ldots,s_{K-1},s_K}}{F_{S_1,s_2,\ldots,s_{K-1},s_K}}. \tag{8} \]

This scalable decision-maker made it possible to be applied to various decision-making problems in information and communication applications [7–9]. Takeuchi et al. applied the laser chaos-based MAB algorithm to channel selection in a wireless LAN system and demonstrated autonomous and adaptive dynamic channel selection [7]. Figure 3 shows dynamic channel selections using laser chaos in IEEE 802.11a. In this study, slot machines (arms) are transformed to channels, and the notion of the reward is transformed to communication throughput; a reward is dispensed by comparing between the resulting throughput and the average throughput over time. The threshold value is updated based on Eqs. (5) and (6). In this way, each threshold value is updated to decide better channel selection. As a result, autonomous and adaptive dynamic channel selection is successfully demonstrated in an IEEE802.11a-based, four-channel WLAN.

3. System model

3.1 Channel bonding

Channel bonding (CB) in IEEE 802.11 wireless LANs is a technique to combine multiple adjacent channels. This technique can potentially achieve higher communication throughput [10]. Channel bonding support in IEEE 802.11n is completely optional, while IEEE 802.11ac and IEEE 802.11ax devices are required to support transmitting and receiving via 40 MHz and 80 MHz CB. Furthermore, IEEE 802.11ax is expected to bond non-contiguous channels, which is called channel aggregation (CA) [20]. In this paper, we focus on IEEE 802.11ac WLAN systems in which CB enables to transmit over multiple adjacent channels which have the bandwidth of 40 MHz or 80 MHz. Figure 4 shows
the available channels in 5.17 GHz to 5.25 GHz using in IEEE 802.11ac. Each channel’s identifier is standardly represented as shown in Fig. 4 [21]. In Fig. 4, four 20 MHz channels are indexed as 36, 40, 44, 48, respectively. On the other hand, two 40 MHz channels are indexed as 38, 46. The bonding of CH36 and CH40 is expressed as CH38. The bonding of CH44 and CH48 is expressed as CH46. Also, 80 MHz channel is indexed as 42. The bonding of CH36, CH40, CH44 and CH48 is expressed as CH42.

3.2 Flexible dynamic channel bonding in WLANs
CB can significantly improve throughput performance by combing multiple channels. However, when multiple APs use the same channel, it will cause co-channel interference between them [11, 12]. Therefore, each AP should choose an appropriate CB bandwidth according to channel conditions. We employ DCB technique in order to prevent such problems.

DCB model is illustrated in Fig. 5. The best CB bandwidth may change over time because of changing network traffic. An appropriate selection of CB bandwidth on an AP can help efficient channel access. Hence, we aim to a laser chaos-based method to find an optimal CB bandwidth for the AP over time in a real dynamic environment. We consider the dynamic CB decision by the AP among 2 channels bonding, 4 channels bonding, or single, independent-channel without bonding.

4. Dynamic channel bonding using hierarchical laser chaos decision maker
In this paper, we apply the laser chaos-based scalable decision-making system [3] to the DCB problems. Figure 6 shows decision making for DCB using laser chaos decision-maker. In our proposed method, slot machines (arms) are transformed to bonding or non-bonding channels, and the notion of the reward is transformed to communication throughput. Since the scalable decision-maker’s accuracy
depends on tree structures, we consider two hierarchical trees: ‘Bandwidth-based tree’ and ‘Channel-based tree’. The details of each hierarchical tree are described as follows.

### 4.1 Bandwidth-based tree

In this subsection, we describe the overview and operation of the bandwidth-based tree as shown in Fig. 7. The bandwidth-based tree makes CB decisions by preferentially comparing arms with different bonding bandwidths. The decision process of the bandwidth-based tree is described as follows.

CB selection is chosen by comparing the sampled values of the laser chaos time series $s(t_i)$ with a threshold value $TH$. The flowchart of CB process is shown in Fig. 7(b).

Firstly, by comparing $s(t_1)$ measured at $t_1 = t$ with the threshold value $TH_1$, it is selected whether to bond or not. If $s(t_1) > TH_1$, CB technique is adopted, while if $s(t_1) < TH_1$, the selection is made not to use CB technique. Assuming $s(t_1) > TH_1$, in the next stage, $s(t_2)$ is compared with $TH_{2,0}$. In this stage, we select whether to use 40 MHz CB or 80 MHz CB. If $s(t_2) > TH_{2,0}$, CH42 (80 MHz bandwidth) is selected. Otherwise, 40 MHz CB is selected. In the same way, the sampled values of the laser chaos time series are compared with threshold values ($TH$). Eventually, the channel used to transmit data is selected. After the channel is selected, the communication throughput through the decided CB frequency band is measured. Then, the $TH(t)$ is updated by whether the measured throughput at time $t$ was greater or smaller than the average throughput during the last $n$ times decision. The update formula of the threshold adjuster value $TA(t)$ can be summarized as follows.

$$
\begin{align*}
TA(t+1) &= \pm \Delta + \alpha TA(t) \quad \text{if } \Gamma_t > \frac{1}{n} \sum_{i=t-n}^{t-1} \Gamma_i, \\
TA(t+1) &= \mp \Omega + \alpha TA(t) \quad \text{otherwise},
\end{align*}
$$

where $\Gamma_t$ is the measured throughput at time $t$.

If the measured throughput at time $t$ was greater than the average throughput during the last $n$ times decision, the channel decision was considered to be ‘correct’. Therefore, the threshold adjuster $TA(t)$ is updated as follows.

$$
\begin{align*}
TA_1(t+1) &= -\Delta + \alpha TA_1(t) \quad \text{if CH38, 42 or 46 is selected,} \\
TA_1(t+1) &= +\Delta + \alpha TA_1(t) \quad \text{if CH36, 40, 44 or 48 is selected,}
\end{align*}
$$
Fig. 7. Architecture for the bandwidth-based tree. (a) DCB method using laser chaos for the bandwidth-based tree. (b) CB selection process in the bandwidth-based tree.

\[
\begin{align*}
T_{A_{2,0}}(t+1) &= -\Delta + \alpha T_{A_{2,0}}(t) \quad \text{if CH42 is selected,} \\
T_{A_{2,0}}(t+1) &= +\Delta + \alpha T_{A_{2,0}}(t) \quad \text{if CH38 or 46 is selected,} \\
T_{A_{2,1}}(t+1) &= -\Delta + \alpha T_{A_{2,1}}(t) \quad \text{if CH44 or 48 is selected,} \\
T_{A_{2,1}}(t+1) &= +\Delta + \alpha T_{A_{2,1}}(t) \quad \text{if CH36 or 40 is selected,}
\end{align*}
\]

\[
\begin{align*}
T_{A_{3,0,1}}(t+1) &= -\Delta + \alpha T_{A_{3,0,1}}(t) \quad \text{if CH46 is selected,} \\
T_{A_{3,0,1}}(t+1) &= +\Delta + \alpha T_{A_{3,0,1}}(t) \quad \text{if CH38 is selected,} \\
T_{A_{3,1,0}}(t+1) &= -\Delta + \alpha T_{A_{3,1,0}}(t) \quad \text{if CH48 is selected,} \\
T_{A_{3,1,0}}(t+1) &= +\Delta + \alpha T_{A_{3,1,0}}(t) \quad \text{if CH44 is selected,} \\
T_{A_{3,1,1}}(t+1) &= -\Delta + \alpha T_{A_{3,1,1}}(t) \quad \text{if CH40 is selected,} \\
T_{A_{3,1,1}}(t+1) &= +\Delta + \alpha T_{A_{3,1,1}}(t) \quad \text{if CH36 is selected,}
\end{align*}
\]

where \(0 \leq \alpha \leq 1\) is a forgetting parameter and \(\Delta\) is the reward, which is set as a constant value (i.e., \(\Delta = 1\)).

If the measured throughput at time \(t\) was smaller than the average throughput during the last \(n\) times decision, the channel decision was considered to be ‘wrong’. Therefore, the threshold adjuster \(T_A(t)\) is updated as follows.
\begin{align}
    T A_1(t + 1) &= +\Omega_1 + \alpha T A_1(t) \quad \text{if CH38, 42 or 46 is selected,} \\
    T A_1(t + 1) &= -\Omega_1 + \alpha T A_1(t) \quad \text{if CH38, 40, 44 or 48 is selected,}
\end{align}

\begin{align}
    T A_2,0(t + 1) &= +\Omega_2,0 + \alpha T A_2,0(t) \quad \text{if CH42 is selected,} \\
    T A_2,0(t + 1) &= -\Omega_2,0 + \alpha T A_2,0(t) \quad \text{if CH38 or 46 is selected,} \\
    T A_2,1(t + 1) &= +\Omega_2,1 + \alpha T A_2,1(t) \quad \text{if CH44 or 48 is selected,} \\
    T A_2,1(t + 1) &= -\Omega_2,1 + \alpha T A_2,1(t) \quad \text{if CH36 or 40 is selected,}
\end{align}

\begin{align}
    T A_{3,0,1}(t + 1) &= +\Omega_{3,0,1} + \alpha T A_{3,0,1}(t) \quad \text{if CH46 is selected,} \\
    T A_{3,0,1}(t + 1) &= -\Omega_{3,0,1} + \alpha T A_{3,0,1}(t) \quad \text{if CH38 is selected,} \\
    T A_{3,1,0}(t + 1) &= +\Omega_{3,1,0} + \alpha T A_{3,1,0}(t) \quad \text{if CH48 is selected,} \\
    T A_{3,1,0}(t + 1) &= -\Omega_{3,1,0} + \alpha T A_{3,1,0}(t) \quad \text{if CH44 is selected,} \\
    T A_{3,1,1}(t + 1) &= +\Omega_{3,1,1} + \alpha T A_{3,1,1}(t) \quad \text{if CH40 is selected,} \\
    T A_{3,1,1}(t + 1) &= -\Omega_{3,1,1} + \alpha T A_{3,1,1}(t) \quad \text{if CH36 is selected},
\end{align}

where \( \Omega_j \) is the reward for \( T A_j \) which represents the ‘fail’ information and is expressed as follows.

\begin{align}
    \Omega_1 &= \frac{P_0(t) + P_1(t)}{2 - (P_0(t) + P_1(t))} \\
    \Omega_{2,0} &= \frac{P_{0,0}(t) + P_{0,1}(t)}{2 - (P_{0,0}(t) + P_{0,1}(t))}, \\
    \Omega_{2,1} &= \frac{P_{1,0}(t) + P_{1,1}(t)}{2 - (P_{1,0}(t) + P_{1,1}(t))}, \\
    \Omega_{3,0,1} &= \frac{P_{0,0,0}(t) + P_{0,0,1}(t)}{2 - (P_{0,0,0}(t) + P_{0,0,1}(t))}, \\
    \Omega_{3,1,0} &= \frac{P_{1,0,0}(t) + P_{1,0,1}(t)}{2 - (P_{1,0,0}(t) + P_{1,0,1}(t))}, \\
    \Omega_{3,1,1} &= \frac{P_{1,1,0}(t) + P_{1,1,1}(t)}{2 - (P_{1,1,0}(t) + P_{1,1,1}(t))},
\end{align}

where \( P_{s_1,s_2,\ldots,s_{K-1},s_K} \) is described in Eq. (8).

### 4.2 Channel-based tree

In this subsection, we describe the overview and operation of the channel-based tree as shown Fig. 8. Unlike the above demonstrated bandwidth-based tree, the channel-based tree makes CB decisions by preferentially comparing arms with different channels. The decision process of the channel-based tree is described as follows.

CB selection is chosen by comparing the sampled values of the laser chaos time series \( s(t_i) \) with threshold values \( TH \), similar to the bandwidth-based tree. The flowchart of the CB process is summarized in Fig. 8(b). Firstly, \( s(t_1) \) measured at \( t = t_1 \) is compared with the threshold value \( TH_1 \). If \( s(t_1) > TH_1 \), CH42 (80 MHz bandwidth) is selected. Otherwise, \( s(t_2) \) is compared with \( TH_{2,1} \). Then, we select whether to use channel sets with higher or lower frequency band. If \( s(t_2) > TH_{2,1} \), the channel set with lower frequency band (CH36, CH38, CH40) is selected. Otherwise, the channel set with higher frequency band (CH44, CH46, CH48) is selected. Assuming \( s(t_2) > TH_{2,1} \), in the next stage, \( s(t_3) \) is compared with \( TH_{3,1,0} \). If \( s(t_3) > TH_{3,1,0} \), CH38 (40 MHz bandwidth) is selected. Otherwise, \( s(t_4) \) is compared with \( TH_{4,1,0} \). In the same way, the sampled values of the laser chaos time series are compared with threshold values \( TH \). Eventually, the channel used to transmit data is selected.

After the channel is selected, the communication throughput through the selected CB frequency band is measured. Then, \( TH(t) \) is updated by whether the measured throughput at time \( t \) was greater or less than the average throughput during the last \( n \) times decision. Threshold adjuster value \( TA(t) \) is updated by Eq. (9).

If the measured throughput at time \( t \) was greater than the average throughput during the last \( n \) times decision, the channel decision was considered to be ‘correct’. Therefore, the threshold adjuster \( TA(t) \) is updated as follows.
Fig. 8. Architecture for the channel-based tree. (a) DCB method using laser chaos for the channel-based tree. (b) CB selection process in the channel-based tree.

\[
\begin{align*}
T_{A1}(t + 1) &= -\Delta + \alpha T_{A1}(t) \quad \text{if CH42 is selected,} \\
T_{A1}(t + 1) &= +\Delta + \alpha T_{A1}(t) \quad \text{otherwise,}
\end{align*}
\]

(19)

\[
\begin{align*}
T_{A2,1}(t + 1) &= -\Delta + \alpha T_{A2,1}(t) \quad \text{if CH36, 38 or 40 is selected,} \\
T_{A2,1}(t + 1) &= +\Delta + \alpha T_{A2,1}(t) \quad \text{if CH44, 46 or 48 is selected,}
\end{align*}
\]

(20)

\[
\begin{align*}
T_{A3,1,0}(t + 1) &= -\Delta + \alpha T_{A3,1,0}(t) \quad \text{if CH38 is selected,} \\
T_{A3,1,0}(t + 1) &= +\Delta + \alpha T_{A3,1,0}(t) \quad \text{if CH36, or 40 is selected,} \\
T_{A3,1,1}(t + 1) &= -\Delta + \alpha T_{A3,1,1}(t) \quad \text{if CH46 is selected,} \\
T_{A3,1,1}(t + 1) &= +\Delta + \alpha T_{A3,1,1}(t) \quad \text{if CH44, or 48 is selected,}
\end{align*}
\]

(21)

\[
\begin{align*}
T_{A4,1,0,1}(t + 1) &= -\Delta + \alpha T_{A4,1,0,1}(t) \quad \text{if CH36 is selected,} \\
T_{A4,1,0,1}(t + 1) &= +\Delta + \alpha T_{A4,1,0,1}(t) \quad \text{if CH40 is selected,} \\
T_{A4,1,1,1}(t + 1) &= -\Delta + \alpha T_{A4,1,1,1}(t) \quad \text{if CH44 is selected,} \\
T_{A4,1,1,1}(t + 1) &= +\Delta + \alpha T_{A4,1,1,1}(t) \quad \text{if CH48 is selected,}
\end{align*}
\]

(22)
where $TA_i(t)$ is the threshold adjuster value of the threshold value $TH_i(t)$. $\alpha (0 \leq \alpha \leq 1)$ is a forgetting parameter and $\Delta$ is a constant value (i.e., $\Delta = 1$).

If the measured throughput at time $t$ was smaller than the average throughput during the last $n$ times decision, the channel decision was considered to be ‘wrong’. Therefore, the threshold adjuster $TA(t)$ is updated as follows.

\[
\begin{align*}
    TA_1(t + 1) &= +\Omega_1 + \alpha TA_1(t) & \text{if CH42 is selected,} \\
    TA_1(t + 1) &= -\Omega_1 + \alpha TA_1(t) & \text{otherwise,}
\end{align*}
\]

\[
\begin{align*}
    TA_{2,1}(t + 1) &= +\Omega_{2,1} + \alpha TA_{2,1}(t) & \text{if CH36, 38 or 40 is selected,} \\
    TA_{2,1}(t + 1) &= -\Omega_{2,1} + \alpha TA_{2,1}(t) & \text{if CH44, 46 or 48 is selected,}
\end{align*}
\]

\[
\begin{align*}
    TA_{3,1,0}(t + 1) &= +\Omega_{3,1,0} + \alpha TA_{3,1,0}(t) & \text{if CH38 is selected,} \\
    TA_{3,1,0}(t + 1) &= -\Omega_{3,1,0} + \alpha TA_{3,1,0}(t) & \text{if CH36, 40 is selected,} \\
    TA_{3,1,1}(t + 1) &= +\Omega_{3,1,1} + \alpha TA_{3,1,1}(t) & \text{if CH46 is selected,} \\
    TA_{3,1,1}(t + 1) &= -\Omega_{3,1,1} + \alpha TA_{3,1,1}(t) & \text{if CH44, 48 is selected,}
\end{align*}
\]

\[
\begin{align*}
    TA_{4,1,0,1}(t + 1) &= +\Omega_{4,1,0,1} + \alpha TA_{4,1,0,1}(t) & \text{if CH36 is selected,} \\
    TA_{4,1,0,1}(t + 1) &= -\Omega_{4,1,0,1} + \alpha TA_{4,1,0,1}(t) & \text{if CH40 is selected,} \\
    TA_{4,1,1,1}(t + 1) &= +\Omega_{4,1,1,1} + \alpha TA_{4,1,1,1}(t) & \text{if CH44 is selected,} \\
    TA_{4,1,1,1}(t + 1) &= -\Omega_{4,1,1,1} + \alpha TA_{4,1,1,1}(t) & \text{if CH48 is selected,}
\end{align*}
\]

where $\Omega_j$ controls the degree of change induced in $TA_j$ which is given by

\[
\Omega_j = \frac{P_0 + P_1}{2 - (P_0 + P_1)}
\]

\[
\begin{align*}
    \Omega_{2,1} &= \frac{P_{1,0}(t) + P_{1,1}(t)}{2 - (P_{1,0}(t) + P_{1,1}(t))}, \\
    \Omega_{3,1,0} &= \frac{P_{1,0,0}(t) + P_{1,0,1}(t)}{2 - (P_{1,0,0}(t) + P_{1,0,1}(t))}, \\
    \Omega_{3,1,1} &= \frac{P_{1,1,0}(t) + P_{1,1,1}(t)}{2 - (P_{1,1,0}(t) + P_{1,1,1}(t))}, \\
    \Omega_{4,1,0,1} &= \frac{P_{1,0,1,0}(t) + P_{1,0,1,1}(t)}{2 - (P_{1,0,1,0}(t) + P_{1,0,1,1}(t))}, \\
    \Omega_{4,1,1,1} &= \frac{P_{1,1,1,0}(t) + P_{1,1,1,1}(t)}{2 - (P_{1,1,1,0}(t) + P_{1,1,1,1}(t))},
\end{align*}
\]

where $P_{S_1,S_2,\ldots,S_{K-1},S_K}$ is described in Eq. (8).

5. Experiment

5.1 Implementation and evaluation methods

We experimentally implemented the proposed method to an AP and examined its operational ability on practical WLAN systems. To accurately evaluate the performance, the experiment was conducted in a shield box to prevent external noises and the reflection of radio waves, as shown in the overview of the experimental apparatus in Fig. 9. A schematic diagram of the experimental setup is shown in Fig. 10. An AP based on the IEEE 802.11ac WLAN system was implemented by a microprocessor (Raspberry Pi 3 MODEL B+) with a wireless LAN adapter (ELECOM, WDC-433DU2HBK). Then, an AP communicates with a personal computer (HP ProBook 4340s, 1.9-GHz central processing unit (CPU), 4-GB random-access memory (RAM)) with Ubuntu operating system. The goal of the experiment is to find the optimal CB for a particular AP in dynamically changing uncertain communication environments and is to achieve higher throughput between the AP and the edge device. The laser chaos time series data, which is sampled to decide channel selections, is stored in the memory of the AP. Sampling rate of the time series data used in this study is 50 ps. The data length is 10000 bytes. Also, the time series data has an 8-bit output, which takes the values from
-127 to 128. The communication throughputs were measured using the command iperf. External traffic was generated by three personal computers connected to arranged 4 APs nearby, where PCs sent packets through any of the 4 channels (i.e., channel 36, 40, 44, and 48) with 8000 bytes every 2 millisecond using command ping. In this paper, we have evaluated performance in 5 cases: with the traffic loads on channel 36, with on channel 40, with on channel 44, with on channel 48, and without any traffic load. In the case that the traffic is added on CH36 or CH40, the best selection will be CH46. When the traffic is added on CH44 or CH48, the best selection will be CH38. The traffic load is added in the following order, CH36, 44, 40, 48. From the 1st to the 100th cycle, there is no load; from the 101st to the 300th cycle, external traffic load is added to CH36; from the 301st to the 500th cycle, external traffic load is added to CH44; from the 501st to the 700th cycle, external traffic load is added to CH40; from the 701st to the 900th cycle, external traffic load is added to CH48 and from the 901st to the 1000th cycle, there is no load. Therefore, the best channel, namely, the most vacant or high-throughput channel, was configured as CH42 during the first 100 cycles and followed by CH46, CH38, CH46, CH38 during the 200 cycles, respectively and followed by CH42 during the last 100 cycles. The forgetting parameter $\alpha$ was set as 0.9, 0.99 or 0.999 and the $\Delta$ was set as 1. The thresholds $TH(t)$ are initially set as 0. $k$ which is the step width of the $TA(t)$ was set as 64, and $N$ was set as 2. $n$ which is the number of throughputs used to calculate the average throughput is set as 10.

In this paper, we first evaluated our proposed methods under dynamically changing environments. Second, we changed the forgetting parameter $\alpha$ and analyzed the effect on the throughputs. Finally, we compared our proposed methods with other MAB algorithms known as $\epsilon$-greedy and UCB1-tuned [22, 23]. $\epsilon$-greedy algorithm plays the arm with the highest reward with probability $1-\epsilon$ and
plays a randomly chosen arm with probability \( \epsilon \). Therefore, \( \epsilon \)-greedy explores with probability \( \epsilon \) and exploits with probability \( 1-\epsilon \). In this study, the notion of the reward for the \( \epsilon \)-greedy algorithm is the throughput; a reward is dispensed by comparing between the resulting throughput and the average throughput during the last 10 times decision. The selection \( j^* \) at time \( t \) for \( \epsilon \)-greedy is represented as follows.

\[
j^* = \begin{cases} 
\arg \max_{j=1}^{K} p_j(t) & \text{with probability } 1 - \epsilon, \\
\text{random selection} & \text{with probability } \epsilon
\end{cases}
\]

where \( K \) is the number decisions. \( p_j(t) \) is the estimated reward probability, which is expressed as follows.

\[
p_j(t) = \frac{r_j(t)}{n_j(t)}
\]

where \( n_j(t) \) is the number of times of selection \( j \) until time \( t \), and \( r_j(t) \) is the number of times of getting reward by selection \( j \) until time \( t \). In this experiment, \( \epsilon \) set as 0.2. In [23], UCB1-tuned is known as the best algorithm which yielded the highest performance for the MAB problems. The notion of the reward for the UCB1-tuned algorithm is the throughput as well as \( \epsilon \)-greedy; a reward is dispensed by comparing between the resulting throughput and the average throughput during the last 10 times decision. The selection \( j^* \) at time \( t \) for UCB1-tuned is represented as follows.

\[
j^* = \arg \max_{j=1}^{K} \left\{ p_j(t) + \sqrt{\frac{2 \ln(N)}{n_j(t)}} \min \left\{ \frac{1}{4}, \frac{p_j(t)}{n_j(t)} + \sqrt{\frac{2 \ln(N)}{n_j(t)}} \right\} \right\},
\]

where \( N \) is iteration and \( p_j(t) \) is given by Eq. (32). \( \hat{p}_j(t) \) is a variance of \( p_j(t) \).

5.2 Experimental results

Figures 11 and 12 summarize experimental demonstrations of the proposed methods based on the bandwidth-based and the channel-based tree, respectively. Under both proposed methods, the AP selected CH42 during the first 100 cycles, followed by CH46, CH38, CH46, CH38 during the 200 cycles, respectively and CH42 during the last 100 cycles as observed in Figs. 11(a) and 12(a). That is, the proposed methods indeed successfully accomplished appropriate decisions in changing environments and achieved higher throughput as observed in Figs. 11(b) and 12(b).

Next, we compared our proposed methods with other MAB algorithms of \( \epsilon \)-greedy and UCB1-tuned. Figure 13 shows the average throughputs of the ten times execution of the proposed and other MAB algorithms. The error bars indicate the standard deviation. We can observe that the proposed methods performed as equally or better than UCB1-tuned and \( \epsilon \)-greedy. Moreover, the proposed method based on the channel-based tree achieved greater throughput than the method based on the bandwidth-based tree. The reason is that when the traffic load of one among channels bonding bandwidth is high, the proposed method based on the channel-based tree is more likely to explore another frequency band than the method based on the bandwidth-based tree.

Finally, we examined the impact of forgetting parameter \( \alpha \) on the throughput performance. Figures 14 and 15 show the average throughput of ten times executions of the channel-based tree method when \( \alpha \) is set as 0.9, 0.99, and 0.999, respectively. The error bars indicate the standard deviation. We can observe that the proposed methods performed as equally or better than UCB1-tuned and \( \epsilon \)-greedy. Moreover, the proposed method based on the channel-based tree achieved greater throughput than the method based on the bandwidth-based tree. The reason is that when the traffic load of one among channels bonding bandwidth is high, the proposed method based on the channel-based tree is more likely to explore another frequency band than the method based on the bandwidth-based tree.

Finally, we examined the impact of forgetting parameter \( \alpha \) on the throughput performance. Figures 14 and 15 show the average throughput of ten times executions of the channel-based tree method when \( \alpha \) is set as 0.9, 0.99, and 0.999, respectively. The error bars indicate the standard deviation. We can observe that the proposed methods performed as equally or better than UCB1-tuned and \( \epsilon \)-greedy. Moreover, the proposed method based on the channel-based tree achieved greater throughput than the method based on the bandwidth-based tree. The reason is that when the traffic load of one among channels bonding bandwidth is high, the proposed method based on the channel-based tree is more likely to explore another frequency band than the method based on the bandwidth-based tree.

5.2 Experimental results

Figures 11 and 12 summarize experimental demonstrations of the proposed methods based on the bandwidth-based and the channel-based tree, respectively. Under both proposed methods, the AP selected CH42 during the first 100 cycles, followed by CH46, CH38, CH46, CH38 during the 200 cycles, respectively and CH42 during the last 100 cycles as observed in Figs. 11(a) and 12(a). That is, the proposed methods indeed successfully accomplished appropriate decisions in changing environments and achieved higher throughput as observed in Figs. 11(b) and 12(b).

Next, we compared our proposed methods with other MAB algorithms of \( \epsilon \)-greedy and UCB1-tuned. Figure 13 shows the average throughputs of the ten times execution of the proposed and other MAB algorithms. The error bars indicate the standard deviation. We can observe that the proposed methods performed as equally or better than UCB1-tuned and \( \epsilon \)-greedy. Moreover, the proposed method based on the channel-based tree achieved greater throughput than the method based on the bandwidth-based tree. The reason is that when the traffic load of one among channels bonding bandwidth is high, the proposed method based on the channel-based tree is more likely to explore another frequency band than the method based on the bandwidth-based tree.

Finally, we examined the impact of forgetting parameter \( \alpha \) on the throughput performance. Figures 14 and 15 show the average throughput of ten times executions of the channel-based tree method when \( \alpha \) is set as 0.9, 0.99, and 0.999, respectively. The error bars indicate the standard deviation. We can observe that the proposed methods performed as equally or better than UCB1-tuned and \( \epsilon \)-greedy. Moreover, the proposed method based on the channel-based tree achieved greater throughput than the method based on the bandwidth-based tree. The reason is that when the traffic load of one among channels bonding bandwidth is high, the proposed method based on the channel-based tree is more likely to explore another frequency band than the method based on the bandwidth-based tree.

Finally, we examined the impact of forgetting parameter \( \alpha \) on the throughput performance. Figures 14 and 15 show the average throughput of ten times executions of the channel-based tree method when \( \alpha \) is set as 0.9, 0.99, and 0.999, respectively. The error bars indicate the standard deviation. We can observe that the proposed methods performed as equally or better than UCB1-tuned and \( \epsilon \)-greedy. Moreover, the proposed method based on the channel-based tree achieved greater throughput than the method based on the bandwidth-based tree. The reason is that when the traffic load of one among channels bonding bandwidth is high, the proposed method based on the channel-based tree is more likely to explore another frequency band than the method based on the bandwidth-based tree.

Finally, we examined the impact of forgetting parameter \( \alpha \) on the throughput performance. Figures 14 and 15 show the average throughput of ten times executions of the channel-based tree method when \( \alpha \) is set as 0.9, 0.99, and 0.999, respectively. The error bars indicate the standard deviation. We can observe that the proposed methods performed as equally or better than UCB1-tuned and \( \epsilon \)-greedy. Moreover, the proposed method based on the channel-based tree achieved greater throughput than the method based on the bandwidth-based tree. The reason is that when the traffic load of one among channels bonding bandwidth is high, the proposed method based on the channel-based tree is more likely to explore another frequency band than the method based on the bandwidth-based tree.
Fig. 11. Experimental results of the bandwidth-based tree method. (a) Selected channel identifier. (b) Throughput.

Fig. 12. Experimental results of the channel-based tree method. (a) Selected channel identifier. (b) Throughput.

Fig. 13. Average throughput performance of each method.

6. Conclusion
In this paper, we applied laser chaos-based decision-making to DCB problems. We designed two hierarchical trees for DCB and experimentally implemented the proposed methods to practical IEEE
802.11ac networks. Experimental results show that our proposed methods can make appropriate decisions in a real dynamic environment. Moreover, we found that the design of the hierarchical tree impacts the performance of decision making for DCB. We also experimentally demonstrated that a smaller forgetting parameter $\alpha$ realizes prompt adaptation to follow dynamic communication environments. Furthermore, we compared proposed method with state-of-the-arts MAB algorithms known as $\epsilon$-greedy policy and UCB1-tuned. Experimental results demonstrated that the proposed method accomplishes superior CB decisions than $\epsilon$-greedy and UCB1-tuned. In future work, we will analyze the hierarchical tree further and explore the optimal parameter of the proposed method to improve the communication performance in dynamic environments. Our proposed method in this study employs a constant reward, but we will study a continuous-valued reward that takes the measured throughput into account as stated in [24]. Also, in this current system, although the decision is fast, it takes time to measure the throughput. In order to make this system run faster, we will study ways to get the state of channel faster. Besides, we will consider extending the proposed DCB method to other IEEE 802.11 standards.

**Acknowledgments**

This work was supported in part by the CREST project (JPMJCR17N2) funded by the Japan Science and Technology Agency.

One of the author, H. Kanemasa, would like to thank Mr. Kentaro Honda for his valuable supports in the preparation of this paper.
References

[1] M. Naruse, Y. Terashima, A. Uchida, and S.J. Kim, “Ultrafast photonic reinforcement learning based on laser chaos,” *Scientific Reports*, vol. 7, 8772, August 2017.

[2] M. Naruse, N. Chauvet, A. Uchida, A. Drezet, G. Bachelier, S. Huant, and H. Hori, “Decision making photonics: solving bandit problems using photons,” *IEEE J. Sel. Topics Quantum Electron*, vol. 26, 7700210, January 2020.

[3] M. Naruse, T. Mihana, H. Hori, H. Saigo, K. Okamura, M. Hasegawa, and A. Uchida, “Scalable photonic reinforcement learning by time-division multiplexing of laser chaos,” *Scientific Reports*, vol. 8, 10890, July 2018.

[4] R. Homma, S. Kochi, T. Niiyama, T. Mihana, Y. Mitsui, K. Kanno, A. Uchida, M. Naruse, and S. Sunada, “On-chip photonic decision maker using spontaneous mode switching in a ring laser,” *Scientific Reports*, vol. 9, 9429, July 2019.

[5] L. Lai, H.E. Gamal, H. Jiang, and H.V. Poor, “Cognitive medium access: exploration, exploitation, and competition,” *IEEE Trans. Mobile Comput*, vol. 10, no. 2, pp. 239–253, February 2011.

[6] N. Okada, M. Hasegawa, N. Chauvet, A. Li, and M. Naruse, “Analysis on Effectiveness of Surrogate Data-Based Laser Chaos Decision Maker,” *Complexity*, vol. 2021, 8877660, February 2021.

[7] S. Takeuchi, M. Hasegawa, K. Kanno, A. Uchida, N. Chauvet, and M. Naruse, “Dynamic channel selection in wireless communications via a multi-armed bandit algorithm using laser chaos time series,” *Scientific Reports*, vol. 10, 1574, January 2020.

[8] Z. Duan, N. Okada, A. Li, M. Naruse, N. Chauvet, and M. Hasegawa, “High-speed optimization of user pairing in NOMA system using laser chaos based MAB algorithm,” *International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, April 2021.

[9] A. Uozumi, N. Okada, M. Naruse, N. Chauvet, A. Li, and M. Hasegawa, “Ultra-fast beam selection using laser chaos decision maker in massive MIMO System,” *IEICE General Conference*, March 2021.

[10] L. Deek, E. Garcia-Villegas, E. Belding, S.J. Lee, and K. Almeroth, “Intelligent channel bonding in 802.11n WLANs,” *IEEE Trans. Mobile Comput.*, vol. 13, no. 6, pp. 1242–1255, June 2014.

[11] L. Deek, E. Garcia-Villegas, E. Belding, S.J. Lee, and K. Almeroth, “The impact of channel bonding on 802.11n network management,” Proc. 7th CONEXT, pp. 1–12, December 2011.

[12] M.Y. Arslan, K. Pelechrinis, L. Broustis, S.V. Krishnamurthy, S. Addepalli, and K. Papagiannaki, “Auto-configuration of 802.11n WLANs,” Proc. 6th Int. CONEXT, pp. 1–12, November 2010.

[13] M. Han, S. Khairy, L.X. Cai, and Y. Cheng, “Performance analysis of opportunistic channel bonding in multi-channel WLANs,” Proc. *IEEE Global Commun. Conf. (GLOBECOM)*, December 2016.

[14] M. Han, S. Khairy, L.X. Cai, Y. Cheng, and F. Hou, “Capacity analysis of opportunistic channel bonding over multi-channel WLANs under unsaturated traffic,” *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1552–1566, March 2020.

[15] S.B. Munoz, F. Wilhelmni, and B. Bellalta, “Dynamic channel bonding in spatially distributed high-density WLANs,” *IEEE Trans. Mobile Comput.*, vol. 19, no. 4, pp. 821–835, April 2020.

[16] S. Khairy, M. Han, L.X. Cai, Y. Cheng, and Z. Han, “A renewal theory based analytical model for multi-channel random access in IEEE 802.11ac/ax,” *IEEE Trans. Mobile Comput.*, vol. 18, no. 5, pp. 1000–1013, May 2019.

[17] B. Bellalta, A. Checco, A. Zocca, and J. Barcelo, “On the interactions between multiple overlapping WLANs using channel bonding,” *IEEE Trans. Veh. Technol.*, vol. 65, no. 2, pp. 796–812, February 2016

[18] S.J. Kim and M. Aono, “Amoeba-inspired algorithm for cognitive medium access,” *NOLTA*, vol. 5, no. 2, pp. 198–209, 2014.

[19] S.J. Kim, M. Aono, and E. Nameda, “Efficient decision-making by volume-conserving physical
object,” *New Journal of Physics*, vol. 17, 083023, August 2015.

[20] B. Bellalta, “IEEE 802.11ax: high-efficiency WLANs,” *IEEE Wireless Commun.*, vol. 23, no. 1, pp. 38–46, February 2016.

[21] *IEEE standard for information technology—telecommunications and information exchange between systems local and metropolitan area networks—specific requirements—part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications*, IEEE Standard P802.11, December 2016.

[22] R. Sutton and A. Barto, *Reinforcement learning: An introduction*. The MIT Press, Cambridge, Massachusetts London, England, 1998.

[23] P. Auer, N. Cesa-Bianchi, and P. Fischer, “Finite-time analysis of the multiarmed bandit problem,” *Machine Learning*, vol. 47, pp. 235–256, May 2002.

[24] K. Kuroda, H. Kato, S.J. Kim, M. Naruse, and M. Hasegawa, “Improving throughput using multi-armed bandit algorithm for wireless LANs,” *NOLTA*, vol. 9, 74–81, January 2018.