An Overlapping Coalitional Game for Cooperative Spectrum Sensing and Access in Cognitive Radio Networks

Zhiyu Dai, Zehua Wang, Student Member, IEEE, and Vincent W.S. Wong, Fellow, IEEE

Abstract—Cognitive radio networks (CRNs) provide an effective solution to address the increasing demand for spectrum resources. The cooperation among secondary users (SUs) improves the sensing performance and spectrum efficiency. In this work, we study the traffic demand-based cooperation strategy of SUs in multichannel CRNs, in which each SU makes its own cooperative sensing decision according to its traffic demand. When an SU has a high traffic demand, it can choose to sense multiple channels in the sensing period and obtain more chances to use spectrum resources. If an SU has no data to transmit, it can choose not to perform spectrum sensing. We formulate this problem as a non-transferable utility (NTU) overlapping coalitional game. In this game, each SU implements a cooperation strategy according to its expected payoff, which takes into account the expected throughput and energy efficiency. We propose two different algorithms, namely an overlapping coalition formation (OCF) algorithm and a sequential coalition formation (SCF) algorithm, to obtain a coalition structure. The OCF algorithm guarantees that the coalition structure is stable, whereas the SCF algorithm has a lower computational complexity and less information exchange among SUs. Moreover, when being assigned a certain channel, SU implements adaptive transmission power control scheme to further improve its energy efficiency. Simulation results show that our proposed algorithms achieve a higher aggregate throughput than the disjoint coalition formation (DCF) algorithm from the literature, where each SU can join only one coalition.

Index Terms—Cognitive radio networks, traffic-demand based cooperation strategy, overlapping coalitional game.

I. INTRODUCTION

THERE is an increasing demand for spectrum resources due to the rapid development of the mobile applications. However, spectrum channels are under-utilized by licensed users [1]. The concept of cognitive radio networks (CRNs) provides a promising solution to utilize the spectrum holes and improve spectrum efficiency. In CRNs, secondary users (SUs) are allowed to access the licensed channels to transmit data as long as the transmission of primary users (PUs) is not interfered with. In order to protect the transmission of PUs, SUs have to perform spectrum sensing to detect the availability of channels before accessing them. To reduce the false detection caused by shadowing or path loss, some SUs can sense the spectrum cooperatively [2]. After the cooperative spectrum sensing, the available channels are shared among the SUs who participate in the cooperation. In the design of cooperative spectrum sensing and access strategy, there is a trade-off between throughput and energy efficiency. On the one hand, the SUs who participate in sensing are offered the opportunities to access the channels, so as to satisfy their traffic demands. On the other hand, cooperative spectrum sensing requires additional energy consumption. Therefore, how to let SUs obtain high throughput while maintaining high energy efficiency is an important issue.

Many studies have been conducted to address cooperative spectrum sensing and access problem in CRNs. They develop different strategies to improve the throughput or energy efficiency in CRNs. By optimizing the sensing parameters (e.g., detection threshold, sensing duration) in cooperative sensing or developing efficient scheduling methods, a better sensing performance and a higher network throughput can be achieved [3], [4]. When maximizing the network throughput, the protection for the transmission of PUs needs to be guaranteed. Therefore, the constraints of power consumption and sensing performance are taken into account [5], [6]. In order to improve the energy efficiency of CRNs, some works focus on developing scheduling algorithms to assign available channels to different SUs [7], [8]. In this way, the spectrum resources are allocated to SUs according to their channel states and traffic demands. Therefore, the energy efficiency is improved. In [9], the optimal sensing time, energy detection threshold, and the number of SUs are determined to maximize the energy efficiency at system level.

Game theory is widely applied to study the cooperation of SUs in CRNs. In [10], the problem of cooperative spectrum sensing in CRNs with multiple channels is formulated as an evolutionary game, where each SU makes its own decision on whether to participate in sensing the spectrum. An entropy-based coalition formation algorithm is developed to help SUs select which channel to sense. In [11], Jiang et al. study
the channel access problem in CRNs. They first propose an approach to estimate the channel states based on Bayesian learning, and then make the channel access decision for each SU by a Markov chain decision process. In order to obtain a better cooperative sensing performance or efficiently distribute the spectrum resources among SUs, the concept of coalitional game is also applied in cooperation strategy design for CRNs. In [12], Saad et al. investigate the trade-off between spectrum sensing and spectrum access. The problem is formulated as a disjoint coalition formation (DCF) game, where each SU can join one coalition to maximize its utility. A collection of coalitions is referred to as a coalition structure. They propose a distributed algorithm to obtain the Nash-stable coalition structure. In [13], Hao et al. apply the hedonic coalition formation game theory to the cooperative spectrum sensing and access problem. The coalition payoff is related to energy efficiency and sensing accuracy. They propose a DCF algorithm to find a stable coalition structure. In [14], a coalitional game approach is applied to study the spectrum access of SUs, where SUs serve as cooperative relays of PUs and obtain the channel access as their payments. It is shown that SUs need to form the grand coalition structure to maximize the system utility.

Although several algorithms have been proposed to improve either the energy efficiency or throughput of SUs in CRNs, most of them do not consider the traffic demand of individual SU. In most of the previous works (e.g., [12] and [13]), SUs are assumed to have infinite data to transmit and the channels are always fully utilized when they are assigned to SUs. This, however, may not be the case in practice. The traffic demands of SUs may change from time to time and vary from one to another. The amount of data that an SU needs to transmit depends on its application (e.g., an SU with a video streaming application usually requires a higher throughput than an SU running a best-effort application). In addition, the traffic demand of an SU may change over time (e.g., an SU with an environmental monitoring application aims to report the changes of temperature). Therefore, when investigating the problem of spectrum sensing and channel access in CRNs, it is necessary to take into account the traffic demands of SUs. Although in [7] and [15], the traffic demands of SUs are considered when studying the spectrum resource allocation problem, they are not considered in the spectrum sensing. Moreover, their objective is to maximize the aggregate system utility instead of developing a cooperative strategy from the perspective of individual SU.

Most of the existing works assume that all SUs should participate in cooperative sensing (e.g., [8], [16], and [17]). However, in an energy-constrained CRN, for an SU having no data to transmit during a certain period of time, it may prefer to stay in idle to conserve energy for future transmission instead of participating in cooperative sensing. Thus, it is reasonable to let SU make its own decision on whether to join in the cooperative sensing according to its traffic demand. The work in [18] uses an evolutionary game approach to determine the cooperative sensing strategy for SUs. However, it does not consider the energy efficiency and traffic demands of SUs.

Coalitional game theory is widely used in developing spectrum sensing and access strategies in CRNs. Most of the previous works aim at formulating the problem as a DCF game and finding a non-overlapping coalition structure (e.g., [12] and [13]). They assume that an SU can join only one coalition and perform cooperative sensing within that coalition. This assumption restricts the cooperation of SUs and limits the improvement of system utility. To relax this assumption, the overlapping coalitional game theory is considered, where a player is allowed to join multiple coalitions. When the overlapping coalitional game theory is applied to the CRNs, an SU can contribute to sensing multiple channels at the same time to increase its probability for channel access. For example, the work in [19] studies the cooperative sensing of SUs. The problem is formulated as an overlapping coalitional game, and a distributed algorithm is proposed to find a stable coalition structure. However, this work focuses on improving sensing performance and does not consider spectrum resource allocation. Moreover, it does not consider the additional energy consumption caused by joining a coalition.

In this paper, we study a traffic demand-based joint cooperative spectrum sensing and access strategy in CRNs. To improve the cooperation opportunities for SUs, we apply overlapping coalitional game theory to develop a cooperation strategy from the perspective of individual SU. When an SU makes a coalition formation decision, it takes into account the expected throughput and energy efficiency. In our proposed strategy, an SU can sense multiple channels in a sensing period and join multiple coalitions to increase its payoff. When an SU has a low traffic demand, it can choose not to join any coalition and quit sensing. In summary, the main contributions of our work are as follows:

- We study a cooperation strategy in CRNs with multiple channels, where each SU makes individual decisions on how many and which channels to sense according to its own traffic demand.
- We formulate this problem as a non-transferable utility (NTU) overlapping coalition formation game, where each SU serves as a player.
- We propose an overlapping coalition formation (OCF) algorithm and prove that it converges to a stable coalition structure.
- We also propose a sequential coalition formation (SCF) algorithm which requires a lower number of iterations and less information exchange among SUs than the OCF algorithm.
- An adaptive transmission power control strategy is proposed to minimize the energy consumption spent on transmission while guaranteeing that the maximum expected throughput is obtained.
- Simulation results show that our proposed OCF and SCF algorithms provide a higher aggregate throughput than the disjoint coalition formation (DCF) algorithm, where each SU can join at most one coalition.

The rest of this paper is organized as follows. Section II describes the system model. In Section III, we formulate the cooperative spectrum sensing and access problem as an overlapping coalitional game and propose two algorithms for the overlapping coalition formation. The properties of these
algorithms are also analyzed. In Section IV, we propose the adaptive transmission power control scheme. Section V presents the performance of our proposed algorithms by comparing them with an existing coalition formation algorithm. Conclusions are given in Section VI.

II. System Model

We consider a CRN with $M$ PUs, $N$ SUs, and a base station for these SUs. Each PU transmits data via a licensed channel. There are $M$ licensed channels. Let $\mathcal{M} = \{1, \ldots, M\}$ denote the set of PUs and $\mathcal{N} = \{1, \ldots, N\}$ denote the set of SUs. The CRN works in a time slotted manner and the slot duration is $T$. At the beginning of each time slot, if an SU chooses to participate in cooperative spectrum sensing and access, it will perform sensing before accessing the available channels. The length of the sensing period is $\tau < T$. Each SU has multiple energy detectors to sense multiple channels in a sensing period [20]. We consider that each detector is used to sense a specific channel and these detectors can finish sensing within the sensing period. There are multiple groups, where each group consists of all the SUs that sense the same channel. When a channel is detected as idle, the BS allocates the channel to a group member. That is, an idle channel can only be accessed by one SU at a time. This is to avoid interference between transmission of different SUs. Moreover, we consider an SU can access at most one channel during a time slot. Each SU has different amount of data in its buffer waiting to be transmitted and only those SUs participating in sensing can obtain access for channel use. SUs selectively participate in cooperative sensing based on the knowledge of the traffic demand and channel capacity.

Let $S_j$ denote the set of SUs choosing to sense and access channel $j \in \mathcal{M}$. We use $P_{d,i,j}$ and $P_{f,i,j}$ to denote the detection probability and false alarm probability of SU $i \in \mathcal{N}$ after sensing channel $j \in \mathcal{M}$, respectively. Specifically, the detection probability is the probability that the channel is detected as busy when it is indeed busy. The false alarm probability is the probability that the channel is detected as busy when it is idle. According to the work in [21], the detection probability of channel $j$ on user $i$ is

$$P_{d,i,j}(\varepsilon, \gamma_{i,j}) = Q\left(\frac{\varepsilon}{\sigma_n^2} - \gamma_{i,j} + 1\right)\sqrt{\frac{N_s}{2\gamma_{i,j} + 1}},$$

where $Q(.)$ is the tail probability for the standard normal distribution, $\varepsilon$ is the detection threshold, $\sigma_n^2$ denotes noise power, $\gamma_{i,j}$ is the received SNR at SU $i$ when it senses channel $j$, and $N_s$ is the number of sensing samples during the sensing period in a time slot. Meanwhile, the false alarm probability of SU $i$ at channel $j$ can be expressed as

$$P_{f,i,j}(P_{d,i,j}, \gamma_{i,j}) = Q\left(\sqrt{2\gamma_{i,j} + 1}Q^{-1}(P_{d,i,j}) + \sqrt{N_s}\gamma_{i,j}\right).$$

Due to path loss and shadowing, an individual sensing result may not be accurate. In our system model, SUs perform cooperative sensing for channels. The BS collects the sensing results from SUs. The BS, which also serves as a fusion centre, uses the OR rule to decide the availability for each channel [21]. The detection probability of the set of SUs $S_j$ who sense channel $j \in \mathcal{M}$ is given by

$$P_{d,j} = 1 - \prod_{i \in S_j}(1 - P_{d,i,j}).$$

A high detection probability leads to a high false alarm probability. To protect the transmission of PU $j$, we set a desired target value $P_{d,j}$ for SUs in set $S_j$. When the OR rule is used to make a sensing decision based on the sensing results of SUs in set $S_j$, the cooperative sensing performance needs to satisfy the target value $P_{d,j}$. We can compute the individual target detection probability of SU $i \in S_j$ as [22]

$$P_{d,i,j} = 1 - (1 - P_{d,j})^{\frac{1}{|S_j|}},$$

where $|S_j|$ denotes the number of SUs sensing channel $j$.

We substitute $P_{d,i,j}$ obtained by (4) to (2), and obtain the value of $P_{f,i,j}$. According to the OR rule, we can calculate the false alarm probability at channel $j$ as

$$P_{f,j} = 1 - \prod_{i \in S_j}(1 - P_{f,i,j}).$$

Let $W_{t,i,j}$ denote the transmit power of SU $i$ when it performs transmission on channel $j$. Let $B_j$ denote the bandwidth of channel $j$. SU $i$ can achieve a transmission rate of $R_{i,j}$ over channel $j$ as

$$R_{i,j} = B_j \log_2\left(1 + \frac{|g_i|^2 W_{t,i,j}}{\sigma_n^2}ight),$$

where $g_i$ denotes the channel gain of transmission link of SU $i$. One approach to estimate the channel gain is by sending pilot signals and using the minimum mean square error estimation or the maximum likelihood estimation [23, pp. 105] [24]. The optimal value of $W_{t,i,j}$ can be obtained from our proposed power control scheme, which will be presented in Section IV.

We use $P_{t,j}$ to denote the probability that channel $j$ is idle. The probability that channel $j$ is correctly detected as idle is $P_{t,j}(1 - P_{f,j})$. When a channel is detected as idle, we assume that each member in the same coalition has an equal chance to access the channel. The probability that SU $i \in S_j$ can access channel $j$ given that this channel is detected as idle is $\frac{1}{|S_j|}$. Since the channel assignment is coordinated by the BS, an idle channel is assigned to only one SU who has participated in cooperative sensing. That is, if channel $j \in \mathcal{M}$ is detected as idle and is assigned to SU $i \in S_j$, then SU $i$ is the only one that can use channel $j$ to transmit data during the time slot. Thus, the probability that SU $i$ is allowed to perform transmission over channel $j$ without interfering the transmission of PU is

$$P_{t,i,j} = P_{t,j}(1 - P_{f,j}) \frac{1}{|S_j|}.$$
Thus, the throughput that SU $i$ can achieve in the time slot is

$$U_{i,j} = \frac{R_{i,j} t_{i,j}}{T}.$$  

(9)

There are two cases that an SU can perform transmission over channel $j \in \mathcal{M}$. The first case is that channel $j$ is detected as idle and it is indeed idle, which has the probability $P_{I,j}(1 - P_{d,j})$. The second case is that channel $j$ is detected as idle but actually it is busy, which has the probability $(1 - P_{I,j})(1 - P_{d,j})$. Therefore, the probability that SU $i$ is assigned channel $j$ to transmit data is

$$P_{i,j}^E = ((1 - P_{I,j})(1 - P_{d,j}) + P_{I,j}(1 - P_{f,j})) \frac{1}{|S_j|}.$$  

(10)

SUs can sense multiple channels during the sensing period by participating in multiple coalitions. We denote the set of channels that SU $i$ chooses to sense as $A_i$. Now we consider the expected throughput that SU $i$ can achieve after sensing the channels in set $A_i$. Let $K$ denote a subset of $A_i$ which contains the channels assigned to SU $i$. Specifically, set $K \subseteq A_i$ is obtained on SU $i$ with a probability given by $\prod_{j \in K}(P_{i,j}^E)^{P_{i,j}^E} \prod_{j \in A_i \setminus K}(1 - P_{i,j}^E)$. Although all the channels in set $K$ are available for SU $i$ to perform its data transmission, SU $i$ is allowed to transmit data over one channel only. Among these $|K|$ channels, we consider the channel which can maximize its throughput (i.e., $\max_{j \in K}\{U_{i,j}\}$) is selected by SU $i$. When SU $i$ chooses channel $\arg\max_{j \in K}\{U_{i,j}\}$ to access and transmit data, its transmission may not succeed due to mis-detection of transmission from the PU. The probability that SU $i$ can successfully transmit data over this channel is $P_{i,j}^E_{\arg\max_{j \in K}\{U_{i,j}\}}$. Thus, the expected throughput that SU $i$ can obtain is given by the following function of channel set $A_i$:

$$U_i(A_i) = \sum_{K \subseteq A_i} \left( \prod_{j \in K} P_{i,j}^E \prod_{j \in A_i \setminus K} (1 - P_{i,j}^E) \times \frac{P_{i,j}^E_{\arg\max_{j \in K}\{U_{i,j}\}}}{P_{i,j}^E_{\arg\max_{j \in K}\{U_{i,j}\}}} \max\{U_{i,j}\} \right).$$  

(11)

As shown in (11), the expected throughput of SU $i$ may increase with the size of set $A_i$. This is because by choosing more channels to sense, SU $i$ can obtain more opportunity to access a channel.

For an energy-constrained SU, it should limit the energy spent on sensing in order to save its energy for data transmission. Now we consider the expected power consumption when SU $i$ chooses channel set $A_i$ to sense and access. The power consumption contains two parts: sensing power consumption and data transmission power consumption. Let $W_{s,i,j}$ denote the sensing power of SU $i$ to channel $j$. Thus, the energy that SU $i$ spends to sense channel $j$ is $E_{s,i,j} = W_{s,i,j} \tau$. Note that $W_{t,i,j}$ is the transmission power of SU $i$ on channel $j$, the energy spent by SU $i$ on data transmission over channel $j$ is $E_{t,i,j} = W_{t,i,j} t_{i,j}$. The energy spent on sensing the channels in set $A_i$ is inevitable for SU $i$. The energy spent on transmission occurs when SU $i$ transfers data over the selected channel $\arg\max_{j \in K}\{U_{i,j}\}$. Thus, the expected power consumption of SU $i$ is a function of the set of channels $A_i$ that it prefers to sense, which is given by

$$E_i(A_i) = \frac{1}{T} \left( \sum_{K \subseteq A_i} \left( \prod_{j \in K} P_{i,j}^E \prod_{j \in A_i \setminus K} (1 - P_{i,j}^E) \times E_{i,j}^{\arg\max_{j \in K}\{U_{i,j}\}} \right) + \sum_{j \in A_i} E_{i,j}^s \right).$$  

(12)

To balance the throughput and power consumption, we use energy efficiency as the metric to evaluate an SU’s decision on cooperation. We define the expected energy efficiency of SU $i$ as [25]

$$\eta_i(A_i) = \frac{U_i(A_i)}{E_i(A_i)}.$$  

(13)

The objective of each SU is to maximize its throughput subject to an energy efficiency constraint. That is, during a time slot, each SU aims to transmit the data in its buffer as much as possible under the condition that its energy efficiency is above a predefined threshold. When the traffic demand of an energy-constrained SU is low, it may not want to spend energy to participate in the cooperative sensing to transmit just several information bits. In this case, the SU can simply choose not to perform sensing, since the cost of participating in the cooperation outweighs the payoff. The energy can be saved for data transmission in the future when there are enough information bits in the buffer. Meanwhile, the SUs who choose to sense the same channel have the equal probability to access it. Thus, the problem is how each SU should cooperate with other SUs to sense the channels. We will address this problem in the next section.

III. OVERLAPPING COALITIONAL GAME FOR COOPERATION STRATEGY

In this section, we formulate the cooperative spectrum sensing and access problem as an NTU overlapping coalitional game. The payoff for each SU captures both the energy efficiency and the expected throughput that can be obtained by joining multiple coalitions. For SUs to make distributed decisions on coalition formation, we define three move rules that take into account both social welfare and individual payoff. We first propose the OCF algorithm to enable SUs to form a stable coalition structure. The convergence of this algorithm is analyzed. Then, we propose the SCF algorithm which requires fewer iterations and less information to be exchanged among SUs than the OCF algorithm.

A. NTU Overlapping Coalitional Game Formation

In this cooperative spectrum sensing and access strategy, SUs choose different channels to maximize their expected throughput while satisfying the energy efficiency requirements. For each channel, the SUs who participate in sensing it form a coalition to improve the sensing performance. These SUs share the spectrum resource of this channel if it is detected as idle. That is, we have a coalitional game for each channel in set $\mathcal{M}$. Each SU in set $\mathcal{N}$ can choose multiple channels and contribute to multiple coalitions. Thus, we have the
overlapping coalitional game where an SU is allowed to join multiple coalitions and each coalition is formed on one channel that the SU prefers to sense. The payoff of an SU in one coalition is constrained by its traffic demand, so the utility obtained by an SU in an overlapping coalition game cannot be transferred to other SUs. Hence, our overlapping coalitional game is an NTU game. We first define the NTU overlapping coalitional game as follows:

**Definition 1** [26]. An NTU overlapping coalitional game $G = (\mathcal{N}, v)$ is given by a set of players $\mathcal{N} = \{1, \ldots, N\}$ and a function $v : S \rightarrow \mathbb{R}^{[S]}$, where $S \subseteq \mathcal{N}$ denotes a coalition formed by players and $v(\emptyset) = 0$.

Corresponding to our system model, the players of this game refer to $N$ SUs. They cooperate with each other and form different coalitions to sense and access different channels. A coalition refers to a set of SUs choosing to sense the same channel. The value function $v$ maps each coalition $S \subseteq \mathcal{N}$ to an $|S|$-dimensional vector. We define the coalition value as the expected payoff that each SU can obtain in the coalition. Specifically, we define $v(S_j) = (x_i(S_j), \forall i \in S_j)$, where $x_i(S_j) = P_i(U_{i,j})$ is the expected payoff that SU $i$ can obtain in an overlapping coalitional game since SU $i$ may join additional coalitions to sense other channels besides channel $j$. We will discuss the total payoff of SU $i$ in the following context.

In a DCF game, a coalition structure is a partition of $\mathcal{N}$ [27]. In an overlapping coalitional game, SUs are allowed to join multiple coalitions at the same time. We define the coalition structure $\Pi$ in an overlapping coalitional game as follows:

**Definition 2** [28]. An overlapping coalitional game structure $\Pi$ over a set $\mathcal{N}$ is defined as a set $\Pi = \{S_1, \ldots, S_K\}$, where $K$ is the number of coalitions. $\forall 1 \leq j \leq K, S_j \subseteq \mathcal{N}$ and $\bigcup_{j=1}^{K} S_j = \mathcal{N}$. Since coalitions can be overlapped, $\exists S_j, S_k \in \Pi, j \neq k$ such that $S_j \cap S_k \neq \emptyset$.

Recall that there are $M$ channels. Those SUs who choose to sense and access channel $j \in M$ will join coalition $S_j$. Moreover, some SUs with low traffic demand may prefer to stay in idle to save energy for future transmission. Thus, they may choose not to perform sensing. We denote the set of SUs that do not sense and access any channel. Thus, they stay in idle to save energy for future transmission. Thus, they may choose not to perform sensing. We denote the set of SUs that do not sense and access any channel. Thus, it does not belong to any other coalitions. Therefore, $S_{M+1}$ is an isolated coalition, which is not overlapped with any other coalitions.

In some existing works (e.g., [12] and [13]), the SUs are selfish, which means an SU seeks to maximize its own payoff without considering the benefits of other SUs. In this case, the channel may be occupied by SUs with low traffic demands while SUs with high traffic demands are still in short of spectrum resources. This leads to a low utilization of the channels. We now propose our preference order of SUs, which takes into account both social welfare and individual payoff. The proposed preference order can help SUs compare two coalition structures and select the one which can provide more social welfare and individual payoff at the same time. We first define the social welfare of a coalition structure $\Pi$ as follows:

$$u(\Pi) = \sum_{i \in \mathcal{N}} p_i(\Pi).$$

The social welfare $u(\Pi)$ of a coalition structure $\Pi$ is referred to as the value of coalition structure $\Pi$, which is the sum of the payoff of each individual player. It also represents the expected overall throughput of SUs by meeting their energy efficiency constraints. We now define the preference order of coalition structures by taking both individual payoff and social welfare into account as follows:

**Definition 3**. In an overlapping coalitional game $G = (\mathcal{N}, v)$, given two coalition structures $\Pi_p$ and $\Pi_q$ over $\mathcal{N}$, $\Pi_p$ is $i$-preferred over $\Pi_q$, where $i \in \mathcal{N}$, is equivalent to $p_i(\Pi_p) > p_i(\Pi_q)$ and $u(\Pi_p) > u(\Pi_q)$. This preference order is represented as

$$\Pi_p \succ_i \Pi_q \iff p_i(\Pi_p) > p_i(\Pi_q) \land u(\Pi_p) > u(\Pi_q).$$

According to Definition 3, a coalition structure is preferred by an SU over another if and only if the total payoff of the coalition structure and the individual payoff of the SU are both increased from one to the other. This preference order not only guarantees the increase of social welfare during the coalition formation, but also keeps the spectrum efficiency above a certain level. In Theorem 1 of Section III-B1, we will show that by using our proposed preference order, the
SU can reach a stable coalition structure in a finite number of iterations during their coalition formation.

**B. Coalition Formation Algorithms**

During the process of overlapping coalition formation, an SU makes its own decision on joining or leaving a coalition according to our proposed preference order. We now define three move rules for the possible moves of an SU. First, the SU joins a new coalition that it does not belong to. Second, the SU leaves one of its current coalitions. Third, the SU switches from one of its current coalitions to a new coalition. To provide a mechanism through which SUs can form different coalitions by performing these moves, we define three move rules as follows:

**Definition 4. Join rule:** Consider a coalition structure \( \Pi_p \) over a set of players \( \mathcal{N} \), where coalition \( S_j \in \Pi_p \) and SU \( i \in \mathcal{N} \setminus S_j \). A new coalition structure is defined as \( \Pi_q = \{\Pi_p \setminus S_j \} \cup \{S_j \cup \{i\}\} \). If \( \Pi_q \succ_i \Pi_p \), SU \( i \) joins coalition \( S_j \) and coalition structure \( \Pi_p \) changes into coalition structure \( \Pi_q \).

According to Definition 4, in coalition structure \( \Pi_p \), SU \( i \) does not belong to coalition \( S_j \) at first. We assume that SU \( i \) joins coalition \( S_j \) and the current coalition structure \( \Pi_p \) changes into a new coalition structure \( \Pi_q \). If coalition structure \( \Pi_q \) is preferred by SU \( i \) over coalition structure \( \Pi_p \) according to Definition 3, SU \( i \) joins coalition \( S_j \) and the current coalition structure \( \Pi_p \) is replaced by \( \Pi_q \). Although the decision is made by SU \( i \), SU \( i \) does not act selfishly without considering the effects of its move to other SUs. This is because an SU joins a new coalition only if its own payoff and the coalition structure value are both improved by his move. For example, if SU \( i \) can increase its payoff by joining coalition \( S_j \) but its join move is detrimental to some SUs in the game and leads to a decrease of the total payoff of SUs, SU \( i \) is not allowed to join coalition \( S_j \). Therefore, our join rule takes both the individual payoff and social welfare into account.

**Definition 5. Quit rule:** Consider a coalition structure \( \Pi_p \) over a set of players \( \mathcal{N} \), where coalition \( S_j \in \Pi_p \) and SU \( i \in \mathcal{N} \cap S_j \). A new coalition structure is defined as \( \Pi_q = \{\Pi_p \setminus S_j \} \cup \{S_j \setminus \{i\}\} \). If \( \Pi_q \succ_i \Pi_p \), then SU \( i \) leaves coalition \( S_j \) and coalition structure \( \Pi_p \) changes into coalition structure \( \Pi_q \).

According to quit rule, SU \( i \) leaves one of its current coalitions \( S_j \) and \( \Pi_p \) changes into \( \Pi_q \) if this newly formed coalition structure is preferred by SU \( i \) over the current one. This occurs in the following two situations. The first situation is that there are too many SUs in coalition \( S_j \) to sense channel \( j \), so the channel has a low chance to be assigned to SU \( i \). In the other situation, SU \( i \) has only several information bits to transmit in a time slot. Therefore, SU \( i \) may prefer to leave coalition \( S_j \) to avoid a negative payoff.

**Definition 6. Switch rule [13]:** Consider a coalition structure \( \Pi_p \) over a set of players \( \mathcal{N} \), where coalitions \( S_j, S_k \in \Pi_p \) and SU \( i \in \mathcal{N} \) satisfies \( i \in S_j \) and \( i \notin S_k \). A new coalition structure is defined as \( \Pi_q = \{\Pi_p \setminus \{S_j, S_k\}\} \cup \{S_j \cup \{i\}\} \cup \{S_k \setminus \{i\}\} \). If \( \Pi_q \succ_i \Pi_p \), then SU \( i \) switches from coalition \( S_j \) to coalition \( S_k \), and coalition structure \( \Pi_p \) changes into coalition structure \( \Pi_q \).

**Algorithm 1** The overlapping coalition formation (OCF) algorithm in CRN for SU \( i \in \mathcal{N} \).

1. **Initialization:** \( S_{M+1} = \mathcal{N} \); \( S_i = \emptyset \) for all \( j \in \mathcal{M} \); \( \Pi := \{S_1, \ldots, S_{M+1}\} \); \( \Pi_i := \Pi \) for each SU \( i \in \mathcal{N} \).
2. SU \( i \) executes \( i.\text{broadcast}(i, D_t) \).
3. SU \( i \) executes \( i.\text{receive}(\ell, D_t) \) for each SU \( \ell \in \mathcal{N} \setminus \{i\} \).
4. **repeat**
   5. SU \( i \) randomly selects \( k \in A_i \cup \{M + 1\} \) and \( j \in \mathcal{M} \setminus A_i \).
   6. \( \Pi_{\text{Quit}} := (\Pi \setminus S_k) \cup (S_k \setminus \{i\}) \).
   7. SU \( i \) calculates \( u(\Pi_{\text{Quit}}) \) and \( p_i(\Pi_{\text{Quit}}) \).
   8. \( \Pi_{\text{Join}} := (\Pi \setminus S_j) \cup (S_j \cup \{i\}) \).
   9. SU \( i \) calculates \( u(\Pi_{\text{Join}}) \) and \( p_i(\Pi_{\text{Join}}) \).
   10. \( \Pi_{\text{Switch}} := (\Pi \setminus \{S_j, S_k\}) \cup (S_j \cup \{i\}) \cup (S_k \setminus \{i\}) \).
   11. SU \( i \) calculates \( u(\Pi_{\text{Switch}}) \) and \( p_i(\Pi_{\text{Switch}}) \).
   12. **if** \( \Pi_{\text{Quit}} \succ_i \Pi \) **then**
   13. \( \Pi_i := \Pi_{\text{Quit}} \); \( u(\Pi_i) := u(\Pi_{\text{Quit}}) \).
   14. **else if** \( \Pi_{\text{Join}} \succ_i \Pi \) **then**
   15. \( \Pi_i := \Pi_{\text{Join}} \); \( u(\Pi_i) := u(\Pi_{\text{Join}}) \).
   16. **else if** \( \Pi_{\text{Switch}} \succ_i \Pi \) **then**
   17. \( \Pi_i := \Pi_{\text{Switch}} \); \( u(\Pi_i) := u(\Pi_{\text{Switch}}) \).
   18. **end if**
   19. SU \( i \) executes \( i.\text{broadcast}(i, \Pi, u(\Pi)) \).
   20. SU \( i \) executes \( i.\text{receive}(\ell, i, u(\Pi(\ell)) \) for each SU \( \ell \in \mathcal{N} \setminus \{i\} \).
   21. \( T_{\text{Info}} := \{1, \ldots, N\} \).
   22. \( U := \arg \max_{\Pi_i \in T_{\text{Info}}} u(\Pi_i) \).
   23. **until** \( i \in \mathcal{N} \), \( \forall k \in A_i \cup \{M + 1\} \) \( j \in \mathcal{M} \setminus A_i \), \( \Pi_{\text{Quit}} \not\in \Pi \), \( \Pi_{\text{Join}} \not\in \Pi \), and \( \Pi_{\text{Switch}} \not\in \Pi \).
   24. SU \( i \) executes \( i.\text{sense}(\Pi) \).
   25. **if** SU \( i \) is assigned channel \( j \) **then**
   26. SU \( i \) calculates \( W_{i,t,k} \) and transmits data with power \( W_{i,t,k}^{\text{opt}} \).
   27. **end if**

According to the definition, SU \( i \) switches from one of its coalitions to a new coalition when the new coalition structure is preferred over the current one. The switch rule balances the size of different coalitions and improves the spectrum efficiency. During the coalition structure formation, some channels may be chosen by many SUs while some other channels are selected by few ones. When an SU notices that its payoff can be improved by switching from the coalition with many members to another coalition with few members, it performs the switch move. In this way, SUs autonomously distribute their contributions to different coalitions and the available channels will be equally utilized.

We now define the stability of an overlapping coalition structure as follows:

**Definition 7.** An overlapping coalition structure \( \Pi \) over a set of players \( \mathcal{N} \) is stable if any SU \( i \in \mathcal{N} \) such that \( i \in S_j \) and \( i \notin S_k \) for coalitions \( S_j, S_k \in \Pi \). SU \( i \) will not deviate from coalition \( S_j \) or join coalition \( S_k \).

According to Definition 7, for each SU \( i \in \mathcal{N} \) in a stable coalition structure, it will not leave any of its coalitions or join any new coalition. Therefore, all the SUs would stay in their current coalitions and do not make any change.

**1) Overlapping Coalition Formation (OCF) Algorithm:** To reach a stable coalition structure, we propose an OCF algorithm shown in Algorithm 1. This algorithm is a distributed algorithm, which is executed by each SU \( i \in \mathcal{N} \). In Algorithm 1, we initialize the coalitions by letting all SUs join quit sensing coalition \( S_{M+1} \) (Line 1). Then SU \( i \) communicates the traffic demand information with other SUs (Lines 2 - 3). This is accomplished by invoking functions \( i.\text{broadcast}() \) and
We assume that SUs exchange the information via a control channel, which is orthogonal to the licensed channels of PUs [29]. After that, SU $i$ makes coalition formation moves according to the three move rules. At the beginning of each iteration, SU $i$ randomly selects a coalition that it currently belongs to and a new coalition it does not belong to, which are denoted by $S_k$ and $S_j$, respectively (Lines 5 - 10). SU $i$ assumes that it leaves coalition $S_k$ and a new coalition structure $\Pi'_{\text{Quit}}$ is formed (Line 6). SU $i$ calculates its resulted payoff and the value of coalition structure $\Pi'_{\text{Quit}}$ according to equations (13) - (15) (Line 7). Similarly, it considers its payoffs in potential new coalition structures resulting from join move and switch move, and calculates the values of coalition structures $\Pi'_{\text{Join}}$ and $\Pi'_{\text{Switch}}$ by equations (13) - (15) (Lines 9 - 11), respectively. If the resulted coalition structure by quit move is preferred over the current one by SU $i$, the coalition structure is updated (Line 13). If the quit move does not improve the coalition structure, SU $i$ considers join move (Line 14) and switch move (Line 16). SU $i$ updates the information of coalition structure and its value, and communicates the updated information with other SUs (Lines 19 - 20). All the updated coalition structure information form a set $T_{\text{info}}$ (Line 21). The coalition structure with the greatest value among $T_{\text{info}}$ is selected as the new coalition structure (Line 22). SUs repeat the coalition formation process until for all SUs they will not deviate from their current coalitions or join other new coalitions (Line 23). In other words, the process converges to a stable coalition structure. After the coalition formation process, SU $i$ cooperatively senses the channels with other SUs in corresponding coalitions according to the coalition structure $\Pi$. This is accomplished by invoking function $i$\_sense(II). During the spectrum access stage, if SU $i$ is allocated a channel, it will calculate the optimal transmission power $W_{i,t,j}^{\text{opt}}$ by solving problems (17) and (21), which will be discussed in the next section. SU $i$ then sets its transmission power to the optimal value to transmit data (Line 26).

The convergence of the proposed OCF algorithm is guaranteed by the following theorem:

**Theorem 1:** The proposed OCF algorithm converges to a stable overlapping coalition structure after a finite number of iterations.

**Proof:** Given that the number of channels $M$ and the number of players $N$ are finite, the number of possible overlapping coalition structures is $2^{MN}$. When implementing the OCF algorithm, the coalition formation process involves a sequence of moves of SUs, which results in a sequence of coalition structures $\{\Pi^{(0)}', \Pi^{(1)}', \ldots, \Pi^{(r)}'\}$, where $r$ is the total number of moves made by SUs. According to Definitions 3 - 6, after each move of an SU, a new coalition structure with a higher value is formed. In addition, the number of possible coalition structures is finite. Therefore, $r$ is a finite number. We now use the proof by contradiction approach to show that if $\Pi^{(r)}'$ is a final coalition structure after the last move of SUs, then it must be stable. Assume that $\Pi^{(r)}'$ is not stable, according to Definition 7, there exists SU $i \in N$ such that SU $i$ will deviate from one of its current coalitions or join a new coalition. Thus, according to the proposed OCF algorithm, SU $i$ will make a join, quit, or switch move, and coalition structure $\Pi^{(r)}'$ will change into a new one. This contradicts the fact that $\Pi^{(r)}'$ is the final coalition structure. Thus, $\Pi^{(r)}'$ is a stable coalition structure. Therefore, after a finite number of iterations, the proposed OCF algorithm returns SUs a stable overlapping coalition structure.

During the process of coalition formation in Algorithm 1, each SU seeks to improve its individual utility while increasing the value of the coalition structure. The moves of SUs lead to a new coalition structure $\Pi$ after each iteration. Thus, a larger total payoff for SUs is obtained every time when the coalition structure $\Pi$ changes. Moreover, our proposed algorithm is adaptive to the changes in the CRN (e.g., number of SUs, channel states). When new SUs join in the network, more channels become available, the traffic demand changes, or the channel state varies, the SUs can adaptively change their cooperation strategies and form a stable coalition structure.

2) **Sequential Coalition Formation (SCF) Algorithm:** Although the convergence of Algorithm 1 is guaranteed, the number of iterations required to reach a stable coalition structure may grow exponentially with the number of SUs. Therefore, we propose the SCF algorithm, which has a lower computational complexity and requires less information to be exchanged among SUs to form an overlapping coalition structure.

The idea in the SCF algorithm is originated from the concept introduced in [30]. In a sequential game of coalition formation, which is defined by the rule of order $\rho$, the coalition structure is formed step by step. In each step, only one player can update the coalition structure. Players make moves one by one according to the rule of order $\rho$. Once a player has joined a coalition, it has to remain in the coalition. In our proposed SCF algorithm, the rule of order $\rho$ is determined according to the traffic demands of SUs. Specifically, the SU that has the highest traffic demand is the first one to make a move. The active SU makes a decision to join multiple coalitions based on the current coalition structure, and remains in these coalitions once it has joined them.

The proposed SCF algorithm is shown in Algorithm 2. In this algorithm, SUs make distributed coalition formation decision. However, their behaviours are coordinated by a central coordinator. First, each SU reports its traffic demand information to the central coordinator (Line 2). This is accomplished by invoking function $i$\_broadcast(). The traffic demand information from different SUs forms an information vector $D$ (Line 4). $H(\mathcal{X})$ is a sorting function that maps a vector $\mathcal{X}$ to a $|\mathcal{X}|$-dimensional vector. It returns a vector with each element representing the sorted index of each $\mathcal{X}$'s element in descending order. For example, for $Y = H(\mathcal{X})$, where $\mathcal{X} = (x_1, x_2, x_3, x_4)$ and $x_2 \geq x_3 \geq x_1 \geq x_4$. $Y = (2, 3, 1, 4)$, which means $x_2$ ranks first, $x_3$ ranks second, $x_1$ ranks third, and $x_4$ ranks fourth in the sequence. The coordinator calculates the rule of order $\rho$ through sorting $D$ with function $H(D)$. The information of $\rho$ is broadcast to all SUs (Line 5). Then, SUs make coalition formation decision one by one according to $\rho$. For example, the SU with the highest traffic demand (e.g., SU $\rho[1]$) makes the first choice. It initializes the coalition structure by setting all SUs in quit sensing coalition $S_{i,t+1}$ (Line 8). For other SUs, the active SU
Algorithm 2 The sequential coalition formation (SCF) algorithm in CRN.

1: for each $i \in \mathcal{N}$ do
2: SU $i$ executes $i.$broadcast($i, D_i$).
3: end for
4: $\mathcal{D} := (D_1, D_2, \ldots, D_N)$
5: Coordinator calculates $\rho := H(\mathcal{D})$ and broadcasts $\rho$ to all SUs.
6: for $i = 1$ to $N$ do
7: if $i = 1$ then
8: SU $\rho(i)$ initializes $S_{M+1} := \mathcal{N}$; $S_j := \emptyset$, $\forall j \in \mathcal{M}$, and $\Pi := \{S_1, S_2, \ldots, S_{M+1}\}$.
9: else
10: SU $i$ executes $i.$receive($\rho(i-1), \Pi$).
11: end if
12: for each $j \in \mathcal{M}$ do
13: $\Pi_{Join} := \{\Pi \setminus S_j\} \cup \{S_j \cup \{\rho(i)\}\}$.
14: SU $\rho(i)$ calculates $u(\Pi_{Join})$ and $p_{\rho(i)}(\Pi_{Join})$.
15: SU $\rho(i)$ randomly selects $k \in A_{\rho(i)} \cup \{M+1\}$.
16: $\Pi_{Switch} := \{\Pi \setminus \{S_j, S_k\}\} \cup \{\{S_j \cup \{\rho(i)\}\} \cup \{S_k \setminus \{\rho(i)\}\}\}$.
17: SU $\rho(i)$ calculates $u(\Pi_{Switch})$ and $p_{\rho(i)}(\Pi_{Switch})$.
18: if $\Pi_{Join} \succ_{\rho(i)} \Pi$ then
19: $\Pi := \Pi_{Join}$.
20: else if $\Pi_{Switch} \succ_{\rho(i)} \Pi$ then
21: $\Pi := \Pi_{Switch}$.
22: end if
23: end for
24: if $i = N$ then
25: SU $\rho(i)$ executes $\rho(i).$broadcast($\rho(i), \Pi$).
26: else
27: SU $\rho(i)$ executes $\rho(i).$send($\rho(i), \rho(i+1), \Pi$).
28: end if
29: end for
30: for each $i \in \mathcal{N} \setminus \{\rho(N)\}$ do
31: SU $i$ executes $i.$receive($\rho(N), \Pi$).
32: end for
33: for each $i \in \mathcal{N}$ do
34: SU $i$ executes $i.$sense($\Pi$).
35: if SU $i$ is assigned channel $j$ then
36: SU $i$ calculates $W^*_{t,i,j}$ and transmits data with power $W^*_{t,i,j}$.
37: end if
38: end for

receiving the updated information of $\Pi$ from previously active SU (Line 10). This is accomplished by invoking function $i.$receive(). During the coalition formation process, each SU checks all the channels one by one to find potential new coalition when it is active. For a new channel $j$, if SU prefers to sense channel $j$, SU can join coalition $j$ or switch to coalition $j$, which means quit move is not considered in this algorithm. Specifically, the active SU considers the potential new coalition structure by assuming that it joins coalition $S_j$ (Line 13). The SU calculates its new payoff and the value of the newly resulted coalition structure from equations (13) - (15) (Line 14). Moreover, the active SU randomly selects a coalition it belongs to (Line 15), and assumes that it switches from this selected coalition to coalition $S_j$ (Line 16). Also, the value of the potential new coalition structure resulting from switch move and SU’s new payoff are calculated (Line 17). If the resulted coalition structure by join move $\Pi_{Join}$ is preferred over the current one $\Pi$, the coalition structure is updated (Line 19). If the join move cannot improve the coalition structure, the resulted structure by switch move $\Pi_{Switch}$ is considered (Line 20). After checking all the new coalitions, the currently active SU $\rho(i)$ sends the updated information of $\Pi$ to the next active SU $\rho(i+1)$ (Line 27). This is accomplished by invoking function $i.$send(). For the last active SU, it broadcasts the final coalition structure to other SUs (Line 25). After the coalition formation process, SUs perform sensing cooperatively within each coalition that they belong to according to the final coalition structure. During the transmission stage, SUs perform data transmission with the optimal transmission power (Lines 34 - 36).

Although the SCF algorithm does not guarantee the stability of the coalition structure obtained, its performance is as good as the OCF algorithm in terms of the aggregate throughput, which is the sum of throughput of all SUs. This will be shown in the simulation results in Section V. Moreover, the SCF algorithm has a lower computational complexity than the OCF algorithm. In the SCF algorithm, each SU makes one move when it is active. The active SU checks each potential channel only once and proposes its coalition structure. Therefore, there are at most $MN$ iterations when running the SCF algorithm. On the other hand, in the OCF algorithm, the formation of a coalition structure takes place every time when a new coalition structure is preferred over the current one. The number of the possible overlapping coalition structures in the OCF algorithm is $2^{MN}$, which is much more than the number of iterations required in the SCF algorithm. Furthermore, the SCF algorithm requires less information to be exchanged than that in the OCF algorithm. In the SCF algorithm, each SU only needs to send the updated coalition structure to the next active SU. In the OCF algorithm, each SU has to exchange the information of the updated coalition structures with others after each iteration, which consumes much more energy.

IV. ADAPTIVE TRANSMISSION POWER CONTROL

In this section, we present the transmission power control scheme given that an SU has been assigned a channel. We prove that this adaptive transmission power control scheme achieves the optimal transmission power for an SU, which minimizes the energy consumption spent on data transmission under the constraint that the maximum throughput of the SU is achieved.

From equations (6), (8), and (9), we notice that the throughput $R_{i,j}$ that SU $i \in \mathcal{N}$ can achieve over channel $j \in \mathcal{M}$ depends on the transmission power $P_{t,i,j}$ and the number of information bits $D_i$ in its buffer. When the amount of data in the buffer of SU $i$ is not large and can be completely transmitted in a time slot with a low data rate, increasing the transmission power $P_{t,i,j}$ may consume more energy in vain since SU $i$ cannot increase its throughput further. Therefore, it is reasonable for SU $i$ to adaptively change its transmission power to balance its transmission data rate and the energy consumption. There are two steps to determine the optimal transmission power for an SU. First, we find the set of transmission power that an SU can achieve the maximum throughput. Second, from this set of values, we determine the one that minimizes the energy consumption for the SU.

Without loss of generality, we consider SU $i$ has been assigned channel $j \in \mathcal{M}$ for data transmission. Since the
bandwidth of channel \( j \) is fixed, the transmission rate \( R_{t,i,j} \) is a function of \( W_{t,i,j} \) given by (6). For SU \( i \) in a time slot, the values of \( T, \tau, \) and \( D_t \) are constant. Thus, both the throughput and the energy consumption during data transmission only depend on the transmission power \( W_{t,i,j} \). Let \( W_{t_{\min}} \) and \( W_{t_{\max}} \) denote the minimum and maximum transmission power of SU \( i \), respectively. Thus, the transmission power \( W_{t,i,j} \) can vary from \( W_{t_{\min}} \) to \( W_{t_{\max}} \). We first consider the throughput maximization problem as follows:

\[
\text{maximize}_{W_{t,i,j}} U_{i,j}(W_{t,i,j})
\]

subject to \( W_{t_{\min}} \leq W_{t,i,j} \leq W_{t_{\max}}. \tag{17}\]

The objective function in problem (17) is a piecewise function. We can rewrite \( U_{i,j}(W_{t,i,j}) \) by substituting (6) in (9) to obtain the objective function as follows:

\[
U_{i,j}(W_{t,i,j}) = \begin{cases} 
B_j \log_2 \left( \frac{1 + |g_{t,i,j}|^2}{\sigma_n^2} \right) (T - \tau), & \text{if } W_{t,i,j} \leq W^h_{t,i,j}, \\
B_j D_t, & \text{otherwise},
\end{cases}
\]

where \( W^h_{t,i,j} > 0 \) is a threshold of transmission power that satisfies

\[
B_j \log_2 \left( 1 + \frac{2 |W^h_{t,i,j}|}{\sigma_n^2} \right) (T - \tau) = D_t. \tag{19}\]

From the objective function of problem (17) given in (18), we find that the range of transmission power \( [W_{t_{\min}}, W_{t_{\max}}] \), from which SU \( i \) selects its transmission power \( W_{t,i,j} \) to transmit data over channel \( j \), affects the optimal solutions to problem (17). Let \( \mathcal{W}_{t_{ij}}^* \) denote the set of optimal solutions to problem (17). We have

\[
\mathcal{W}_{t_{ij}}^* = \begin{cases} 
\{W_{t_{\max}}\}, & \text{if } W_{t_{\min}} \leq W^h_{t,i,j}, \\
\{W_{t_{ij}} \mid W^h_{t,i,j} \leq W_{t_{ij}} \leq W_{t_{\max}}\}, & \text{if } W_{t_{\min}} \leq W_{t_{\max}} < W_{t_{ij}}, \\
\{W_{t_{ij}} \mid W_{t_{\min}} \leq W_{t_{ij}} \leq W_{t_{\max}}\}, & \text{otherwise}.
\end{cases} \tag{20}\]

According to the set in (20), problem (17) has either a unique solution or multiple solutions depending on the threshold \( W^h_{t,i,j} \) and the range of the possible transmission power of SU \( i \). In other words, an SU can set its transmission power to be equal to a certain value or any value within an interval to obtain its maximum throughput. For example, when \( W_{t_{\min}} \leq W_{t_{ij}} \leq W_{t_{\max}} \), the optimal solution to problem (17) can be any value between \( W^h_{t,i,j} \) and \( W_{t_{\max}} \). In order to save energy spent on data transmission while maintaining the maximum throughput, we consider an optimization problem that minimizes the energy consumption of SU \( i \) for data transmission under the constraint that the maximum throughput is obtained. This optimization problem is formulated as follows:

\[
\text{minimize}_{W_{t,i,j}} E^t_{i,j}(W_{t,i,j})
\]

subject to \( U_{i,j}(W_{t,i,j}) \geq r_{t_{ij}}^{\max}, \tag{21}\]

where \( r_{t_{ij}}^{\max} \) is the maximum value of \( U_{i,j}(W_{t,i,j}) \), which can be achieved only if the transmission power \( W_{t,i,j} \) is an optimal solution to problem (17) (i.e., \( W_{t_{ij}} \in \mathcal{W}_{t_{ij}}^* \)). In other words, the solution to problem (17) serves as the constraint in problem (21). Thus, the requirement for a maximum throughput on SU \( i \) can be guaranteed when we minimize the energy consumption for the data transmission. In summary, depending on the values of the parameters \( W^h_{t,i,j}, W_{t_{\min}} \), and \( W_{t_{\max}} \), the solutions to problem (17) may not be unique. Therefore, the optimal solution to problem (21), which is denoted by \( W_{t_{ij}}^{\text{opt}} \), can be obtained based on the following three cases.

In Case 1, where \( W_{t_{\min}} \leq W^h_{t,i,j} \), set \( W_{t_{ij}}^{\text{opt}} = W^h_{t,i,j} \), is a singleton set. The unique solution to problem (17) is also the optimal solution to problem (21). That is, we have \( W_{t_{ij}}^{\text{opt}} = W_{t_{ij}}^h \). It means that SU \( i \) has to perform data transmission over channel \( j \) with its maximum transmission power in order to achieve the maximum throughput. In this case, the minimum energy consumption for data transmission during the time slot (i.e., the optimal value of the objective function in problem (21)) is \( W_{t_{\max}}(T - \tau) \).

In Case 2, we have \( W_{t_{\min}} \leq W^h_{t,i,j} \leq W_{t_{\max}} \). The optimal solution to problem (17) can be any element in the set \( \mathcal{W}_{t_{ij}}^* = \{W_{t_{ij}} \mid W^h_{t,i,j} \leq W_{t_{ij}} \leq W_{t_{\max}}\} \). Note that \( \mathcal{W}_{t_{ij}}^* \) is also the set of transmission power which is feasible to the constraint in problem (21). By considering equations (6) and (8), and noting that \( E^t_{i,j}(W_{t,i,j}) = E^t_{i,j}(W_{t,i,j}^{\text{opt}}) \), the objective function of problem (21) is given by

\[
E^t_{i,j}(W_{t,i,j}) = \frac{W_{t,i,j} D_t}{B_j \log_2 \left( 1 + \frac{|g_{t,i,j}|^2}{\sigma_n^2} \right)}. \tag{22}\]

We now have the following proposition:

**Proposition 1:** Function \( E^t_{i,j}(W_{t,i,j}) \) in (22) is a monotonically increasing function with respect to \( W_{t,i,j} \) > 0.

**Proof:** We take the first derivative of function \( E^t_{i,j}(W_{t,i,j}) \) with respect to \( W_{t,i,j} \). We have

\[
E^t_{i,j}'(W_{t,i,j}) = \frac{\ln 2 D_t \sigma_n^2}{B_j |g_{t,i,j}|^2} \left( \ln (1 + |g_{t,i,j}|^2 W^t_{t,i,j}) \right)^2 \times \left( \ln (1 + |g_{t,i,j}|^2 W^t_{t,i,j}) - \frac{|g_{t,i,j}|^2 W^t_{t,i,j}}{1 + |g_{t,i,j}|^2 W^t_{t,i,j}} \right). \tag{23}\]

Since \( \frac{\ln 2 D_t \sigma_n^2}{B_j |g_{t,i,j}|^2} \left( \ln (1 + |g_{t,i,j}|^2 W^t_{t,i,j}) \right)^2 > 0 \), to show the monotonicity of function \( E^t_{i,j}(W_{t,i,j}) \), we need to prove that \( \ln (1 + |g_{t,i,j}|^2 W^t_{t,i,j}) - \left( 1 + |g_{t,i,j}|^2 W^t_{t,i,j} \right)^{-1} \left( |g_{t,i,j}|^2 W^t_{t,i,j} \right) > 0 \) for \( W_{t,i,j} > 0 \). We denote \( y(W_{t,i,j}) = \ln (1 + |g_{t,i,j}|^2 W^t_{t,i,j}) - \left( 1 + |g_{t,i,j}|^2 W^t_{t,i,j} \right)^{-1} \left( |g_{t,i,j}|^2 W^t_{t,i,j} \right) \). We determine the first derivative of \( y(W_{t,i,j}) \) with respect to \( W_{t,i,j} \), which is given by

\[
y'(W_{t,i,j}) = \frac{|g_{t,i,j}|^2}{\sigma_n^2} \left( \frac{1}{1 + |g_{t,i,j}|^2 W^t_{t,i,j}} - \frac{1}{(1 + |g_{t,i,j}|^2 W^t_{t,i,j})^2} \right) = \frac{|g_{t,i,j}|^2 W^t_{t,i,j}}{(1 + |g_{t,i,j}|^2 W^t_{t,i,j})^2}. \tag{24}\]

That is, we have \( y'(W_{t,i,j}) > 0 \) for \( W_{t,i,j} > 0 \). Thus, \( y(W_{t,i,j}) \) is monotonically increasing with \( W_{t,i,j} \), and \( y(W_{t,i,j}) > y(0) = 0 \) for \( W_{t,i,j} > 0 \). Therefore, \( E^t_{i,j}'(W_{t,i,j}) > 0 \) and
According to Proposition 1, the value of $E_{t,i,j}^i(W_{t,i,j})$ increases with $W_{t,i,j}$ when $W_{t,i,j}^h \leq W_{t,i,j} \leq W_{t,i,j}^{max}$. Thus, the optimal solution to problem (17) is $W_{t,i,j}^{opt} = W_{t,i,j}^h$.

In Case 3, we have $W_{t,i,j}^h \leq W_{t,i,j}^{min}$, the solutions to problem (17) are given by the set $W_{t,i,j}^* = \{W_{t,i,j}^{min}, W_{t,i,j} \leq W_{t,i,j}^{min} \leq W_{t,i,j}^{max} \}$. The objective function is the same with that in Case 2. According to Proposition 1, the optimal solution to problem (21) is $W_{t,i,j}^{opt} = W_{t,i,j}^{min}$.

The optimal solution to problem (21) changes according to the relations between $W_{t,i,j}^{min}$ and the range of transmission power $[W_{t,i,j}^{min}, W_{t,i,j}^{max}]$. When the range of the transmission power of SU $i$ is fixed, the value of $W_{t,i,j}^{h}$ determines the solution set $W_{t,i,j}^*$ for problem (17), and thus affects the solution to problem (21). According to the definition of $W_{t,i,j}^{h}$, the value of $W_{t,i,j}^{h}$ depends on the traffic demand of SU $i$. When the number of information bits in the buffer of SU $i$ decreases (i.e., $D_i$ goes smaller), $W_{t,i,j}^{h}$ also decreases by the equality in (19). Eventually, Case 3 occurs when $W_{t,i,j}^{h}$ is smaller than $W_{t,i,j}^{min}$. Therefore, SU $i$ saves energy by setting its transmission power as $W_{t,i,j}^{min}$. On the contrary, when $D_i$ is very large, $W_{t,i,j}^{h}$ may be greater than $W_{t,i,j}^{max}$. In order to obtain the highest throughput, SU $i$ needs to perform its data transmission with a transmission power $W_{t,i,j}^{max}$. Thus, the transmission power of an SU on a channel is adaptively controlled according to its traffic demand. Meanwhile, the energy spent on data transmission is minimized under the constraint that the SU achieves its maximum throughput.

V. PERFORMANCE EVALUATION

In this section, we compare the performance of the OCF algorithm, SCF algorithm, and DCF algorithm from the perspective of aggregate throughput. In the DCF algorithm, each SU can only join at most one coalition, which is similar to the algorithm proposed in [12]. We consider a CRN with $N$ SUs and $M$ PUs (i.e., $M$ licensed channels). SUs are randomly placed in a $100 \times 100$ m square region. The base station is placed at the centre of the square region. We model the channel gain of the link of SU $i$ as $|g_i|^2 = 1/d_i^n$, where $d_i$ is the distance from SU $i$ to the base station, and $n$ is the path loss exponent. The probability that the channel is idle is randomly chosen between [0.5, 1]. The number of sensing samples during the sensing period in a time slot is $N_s = 5000$, which is similar to the parameter setting in [31]. The number of packets generated by each SU during a time slot follows the Poisson distribution. A list of simulation parameters is summarized in Table I.

Fig. 1 shows a snapshot of a stable overlapping coalition structure obtained by implementing the OCF algorithm. There are 3 PUs (i.e., 3 licensed channels), 7 SUs, and a base station (BS) in the CRN. The base station is located at the centre of the square area. The 7 SUs are randomly located in the area. All the SUs in a same ellipse form a coalition to sense and access a channel. Results in Fig. 1 show that SUs 1, 4 and 7 belong to coalition 1, which corresponds to channel 1. SUs 3, 4, 5 and 7 form a coalition to sense and access channel 2. SU 2 forms a singleton coalition to sense and access channel 3. SU 6 joins the coalition 4, which is the coalition of quitting sensing. Thus, the overlapping coalition structure is $\Pi = \{\{1, 4, 7\}_1, \{3, 4, 5, 7\}_2, \{2\}_3, \{6\}_4\}$. In this stable coalition structure, coalitions 1 and 2 are overlapped with each other. SUs 4 and 7 contribute to both coalitions at the same time. SU 6 chooses not to cooperate with other SUs due to the fact that it has no traffic demand in the time slot. Therefore, SU 6 is in the quit sensing coalition.

Fig. 2 shows the aggregate throughput of SUs when the number of SUs $N$ increases from 2 to 20. As shown in this figure, both OCF and SCF algorithms outperform the DCF algorithm in terms of aggregate throughput. This is because

TABLE I

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of SUs $N$</td>
<td>10</td>
</tr>
<tr>
<td>Number of PUs (licensed channels) $M$</td>
<td>6</td>
</tr>
<tr>
<td>Path loss exponent $n$</td>
<td>2</td>
</tr>
<tr>
<td>Bandwidth of channel $j B_j$</td>
<td>100 kHz</td>
</tr>
<tr>
<td>Probability that channel $j$ is being idle $P_{d,j}$</td>
<td>[0.5, 1]</td>
</tr>
<tr>
<td>Target detection probability $P_{d,i,j}$</td>
<td>0.99</td>
</tr>
<tr>
<td>Noise power $\sigma_n^2$</td>
<td>0.01 mW</td>
</tr>
<tr>
<td>Maximum transmission power of SU $i W_{t,i,j}^{max}$</td>
<td>150 mW</td>
</tr>
<tr>
<td>Minimum transmission power of SU $i W_{t,i,j}^{min}$</td>
<td>50 mW</td>
</tr>
<tr>
<td>Received SNR at each SU during the sensing stage $\gamma_{i,j}$</td>
<td>−15 dB</td>
</tr>
<tr>
<td>Sensing power of SU $i$ on channel $j W_{s,i,j}$</td>
<td>50 mW</td>
</tr>
<tr>
<td>Slot duration $T$</td>
<td>100 ms</td>
</tr>
<tr>
<td>Sensing duration $\tau$</td>
<td>5 ms</td>
</tr>
<tr>
<td>Number of sensing samples during the sensing stage $N_s$</td>
<td>5000</td>
</tr>
<tr>
<td>Average number of packets generated by SU during a time slot $\lambda$</td>
<td>0.5 packet per time slot</td>
</tr>
<tr>
<td>Packet size</td>
<td>20 kbit</td>
</tr>
<tr>
<td>Buffer size of each SU</td>
<td>200 kbit</td>
</tr>
<tr>
<td>Lower bound of the energy efficiency $\eta_{min}$</td>
<td>500 kbit/Joule</td>
</tr>
</tbody>
</table>

Fig. 1. An example of the stable overlapping coalition structure ($M = 3, N = 7$) by the OCF algorithm.
in OCF and SCF algorithms, each SU can join multiple coalitions, which increases its chance to be assigned a channel. However, in the DCF algorithm, an SU can only join one coalition and share one channel with other SUs. Therefore, its chance of being assigned a channel is limited. This result shows that the overlapping coalitional game strategy improves spectrum efficiency. Furthermore, the performance of SCF algorithm is similar to or slightly better than the OCF algorithm. This is because in the SCF algorithm, SUs with higher traffic demands are prioritized to join coalitions. These SUs have a higher chance to transmit data than those SUs with lower traffic demands. Therefore, the SCF algorithm achieves a slightly higher aggregate throughput than the OCF algorithm.

Fig. 3 shows the aggregate throughput of SUs when the number of PUs $M$ (i.e., the number of channels) increases from 1 to 10. We find that the SCF algorithm obtains a slightly higher aggregate throughput than the OCF algorithm. This is because SUs with high traffic demands have priority to choose coalitions in the SCF algorithm, which enables them to have a higher chance to access the channels. When $M$ is small, the performance gap from the DCF algorithm to our proposed algorithms is small. The gap of performance becomes larger when $M$ increases. When more channels become available, SUs in our OCF or SCF algorithms obtain higher throughput by joining multiple coalitions. However, SUs running the DCF algorithm obtain limited improvement of throughput due to the constraint that it can sense and access at most one channel only.

Fig. 4 shows the aggregate throughput when increasing the packet arrival rate $\lambda$. Both the OCF algorithm and SCF algorithm outperform the DCF algorithm in terms of the aggregate throughput when $\lambda$ changes from 0.1 to 1. The throughput increases with the traffic demands for all algorithms when $\lambda$ is small. Larger value of $\lambda$ means more data packets are generated at each SU during each time slot, which encourages SUs to join coalitions to obtain a higher throughput. When $\lambda \geq 0.7$, the aggregate throughput does not increase significantly with the traffic demand. This is because the increase of throughput is constrained by the spectrum resources. In the OCF algorithm, each SU in the same coalition has an equal chance to access an available channel. Some available channels may be used by SUs with low traffic demands, which can lead to a relatively low utilization of spectrum resources. SUs with higher traffic demands in the SCF algorithm are prioritized to be assigned an available channel. Thus, the SCF algorithm achieves a slightly higher aggregate throughput than the OCF algorithm.

We also perform simulations to determine the channel access delay of SUs who have different traffic demands in a CRN. The channel access delay is defined as the average duration from the time that a packet is generated by an SU to the time the packet is being transmitted. In this set of simulations, the packet arrival rate $\lambda$ of SU $i = 1, 2, \ldots, 10$ is 0.03, 0.06, $\ldots$, 0.3, respectively. These SUs have the same distance to the base station, which is 60 m. We set the transmission power of SU $i$ over channel $j$ as a constant by letting $W_{i,j}^{\text{min}} = W_{i,j}^{\text{max}} = 100$ mW. The lower bound of the energy efficiency $\eta_{\text{min}}$ is 1500 kbit/Joule. We have 1000 time slots in each simulation run, where the slot duration $T$ is equal to 0.5 sec. The average simulation results of the DCF, SCF, and OCF algorithms are given in Fig. 5. An SU does not have any incentive to join coalitions until a certain amount of information bits have been accumulated. An SU with a lower traffic demand needs to wait a longer time to have enough data in its buffer. Thus in Fig. 5, the SU who has a lower traffic demand has a longer channel access delay than the SU who has a higher traffic demand. In general, our proposed SCF and OCF algorithms obtain smaller channel access delay than the DCF algorithm because an SU can join multiple coalitions in
the SCF and OCF algorithms to increase its chance of being assigned a channel. We find the channel access delay of SU 
\( i = 1 \) in the OCF algorithm is greater than its channel access delay in the SCF and DCF algorithms. This can be explained
as follows. An SU in the OCF algorithm can form a new coalition structure in each round by increasing its payoff and
the social welfare. The SU with a higher traffic demand has more potential to form such a coalition structure which can
benefit itself and improve the social welfare at the same time. That is, the converged coalitions in the OCF algorithm can
better benefit the SUs with higher traffic demands. Thus, the OCF algorithm results in the longest channel access delay for
SU \( i = 1 \), whose packet arrival rate is \( \lambda = 0.03 \). This also explains why the SU \( i = 10 \) with \( \lambda = 0.3 \) obtains the shortest
channel access delay in the OCF algorithm compared with the SCF and DCF algorithms.

In Fig. 6, we present the traffic demand satisfaction ratio of each SU with the same simulation settings as those used for
Fig. 5. Let \( \gamma_i \) denote the traffic demand satisfaction ratio of SU \( i \), which is defined as the number of packets transmitted
by SU \( i \) and successfully received by the base station over the number of packets generated by SU \( i \). From Fig. 6, we find that the DCF, SCF, and OCF algorithms maintain a good fairness for SUs with different traffic demands, since these SUs can obtain almost the same traffic demand satisfaction ratio. Specifically, each SU \( i = 1, \ldots, 10 \) can obtain the traffic demand satisfaction ratio \( \gamma_i > 0.991 \) in these algorithms, which means each SU can successfully transmit almost all of
its generated packets to the base station. The traffic demand satisfaction ratio is not equal to 1 because some packets are
dropped when we terminate each simulation run after 1000 time slots. The high traffic demand satisfaction ratio is because
the channel capacity available for SUs is greater than their traffic demand. In our simulations, we also determine the
traffic demand to available channel capacity ratio for the SUs, which is defined as the ratio between the total traffic
demand of SUs and the total channel capacity available for their data transmission. With the simulation settings used in
Figs. 5-6, this ratio is equal to 0.24. Note that when the traffic demand to available channel capacity ratio is greater than
one, there will be competition between SUs. In that case, admission control can be used at the BS to keep this ratio to
be less than one and to maintain fairness among the SUs.

Fig. 7 shows the aggregate throughput for various energy efficiency threshold \( \eta_{\text{min}} \). As the same transmission power
control scheme is used in both OCF and SCF algorithms, they have similar aggregate throughput and outperform the DCF
algorithm when the energy efficiency threshold varies from 0 kbit/J to 3500 kbit/J. When \( \eta_{\text{min}} \) is smaller than 1500 kbit/J, the
energy efficiency threshold has little effect on the number of coalitions that an SU can join. Therefore, SUs are able to
join enough coalitions to satisfy their traffic demand. In this case, an increase of \( \eta_{\text{min}} \) has little effect on SUs’ throughput.
However, when \( \eta_{\text{min}} \geq 1500 \) kbit/J, the number of coalitions that an SU is allowed to join is limited. SUs are refrained
to transmitting data in their buffer until the expected energy efficiency becomes greater than the threshold. This leads to the
decrease of throughput.

Fig. 8 shows the number of iterations when running the OCF and SCF algorithms as the number of SUs increases. When \( N \)
is small, the number of iterations for these two algorithms are similar. This is because when there are only a few SUs, the
cooperation possibilities are limited. The number of iterations that the OCF algorithm needs to converge is not very large.
However, when \( N > 10 \), the performance gap between these two algorithms becomes larger. In the SCF algorithm, each
SU checks each new coalition only once when it is active. However, in the OCF algorithm, all SUs try to form new
coalitions whenever there is a change in coalition structure.
VI. CONCLUSION

In this paper, we studied a traffic-demand based cooperation strategy in CRNs with multiple channels. We proposed a joint cooperative spectrum sensing and access scheme to enable energy-constrained SUs to obtain a high throughput while maintaining a high energy efficiency. An overlapping coalitional game is formulated to solve this problem, in which each SU makes its own decision to form overlapping coalitions with other SUs to sense and access multiple channels cooperatively. To reach a stable coalition structure, we proposed an OCF algorithm based on three move rules, which captures both individual payoff and social welfare. We proved that our proposed OCF algorithm converges to a stable coalition structure. We also proposed an SCF algorithm which has a lower computational complexity and requires less information exchange. Moreover, an adaptive transmission power control scheme is proposed. Simulation results show that the OCF and SCF algorithms have similar performance in terms of aggregate throughput. Both algorithms outperform the DCF algorithm proposed in [12]. For future work, we will extend our system model to a general setting, where an SU can adapt its sensing duration and sensing power according to the network settings.

REFERENCES

Zhiyu Dai received the B.Eng. degree in Control and System Engineering from Zhejiang University, Hangzhou, China, in 2012, and the M.A.Sc. degree in Electrical and Computer Engineering from the University of British Columbia (UBC), Vancouver, BC, Canada, in 2014. He is currently working as a software engineer in D&B Cloud Innovation Center, Vancouver, BC, Canada. His research interests include cognitive radio networks, data analysis, machine learning, and data mining.

Zehua Wang (S’11) received the B.Eng. degree in Software Engineering from Wuhan University, Wuhan, China, in 2009, and the M.Eng. degree in Electrical and Computer Engineering from Memorial University of Newfoundland, St Johns, NL, Canada, in 2011. He is currently a Ph.D. candidate at the University of British Columbia (UBC), Vancouver, BC, Canada. His research interests include machine-type communications, device-to-device communications, social networks, and routing and forwarding in mobile ad hoc networks. He has been the recipient of the Four Year Doctoral Fellowship (4YF) at UBC since 2012. He was also awarded the Graduate Support Initiative (GSI) Award from UBC. Mr. Wang served as technical program committee (TPC) members for several conferences including the IEEE International Conference on Communications (ICC) 2012, 2014–2016 and the IEEE Global Communications Conference (GLOBECOM) 2014–2015.

Vincent W.S. Wong (S’94, M’00, SM’07, F’16) received the B.Sc. degree from the University of Manitoba, Winnipeg, MB, Canada, in 1994, the M.A.Sc. degree from the University of Waterloo, Waterloo, ON, Canada, in 1996, and the Ph.D. degree from the University of British Columbia (UBC), Vancouver, BC, Canada, in 2000. From 2000 to 2001, he worked as a systems engineer at PMC-Sierra Inc. He joined the Department of Electrical and Computer Engineering at UBC in 2002 and is currently a Professor. His research areas include protocol design, optimization, and resource management of communication networks, with applications to wireless networks, smart grid, and the Internet. Dr. Wong is an Editor of the IEEE Transactions on Communications. He is a Guest Editor of IEEE Journal on Selected Areas in Communications, special issue on “Emerging Technologies” in 2016. He has served on the editorial boards of IEEE Transactions on Vehicular Technology and Journal of Communications and Networks. He has served as a Technical Program Co-chair of IEEE SmartGridComm’14, as well as a Symposium Co-chair of IEEE SmartGridComm’13 and IEEE Globecom’13. He is the Chair of the IEEE Communications Society Emerging Technical Sub-Committee on Smart Grid Communications and the IEEE Vancouver Joint Communications Chapter. Dr. Wong received the 2014 UBC Killam Faculty Research Fellowship. He is a Fellow of the IEEE.