Wireless Networks (2020) 26:4705–4721
https://doi.org/10.1007/s11276-020-02369-1

Game-theory-based lifetime maximization of multi-channel cooperative spectrum sensing in wireless sensor networks

Asma Bagheri[1] · Ataollah Ebrahimzadeh[1] · Maryam Najimi[2]

Published online: 22 May 2020

Abstract

Accurate and efficient detection of the radio-frequency spectrum is a challenging issue in wireless sensor networks (WSNs), which are used for multi-channel cooperative spectrum sensing (MCSS). Due to the limited battery power of sensors, lifetime maximization of a WSN is an important issue further sensing quality requirements. The issue is more complex if the low-cost sensors cannot sense more than one channel simultaneously, because they do not have high-speed Analogue-to-Digital-Convertors which need high-power batteries. This paper proposes a novel game-theoretic sensor selection algorithm for MCSS that extends the network lifetime assuming the quality of sensing and the limited ability of sensors. To this end, an optimization problem is formulated using the "max–min" method, in which the minimum remaining energy of sensors is maximized to keep energy balancing in the WSN. This paper proposes a coalition game to solve the problem, in which sensors act as game players and decide to make disjoint coalitions for MCSS. Each coalition senses one of the channels. Other nodes, that decide to sense none of the channels, turn off their sensing module to reserve energy. First, a novel utility function for the coalitions is proposed based on the remaining energy and consumption energy of sensors besides their detection quality. Then, an algorithm is designed to reach a Nash-Equilibrium (NE) coalition structure. The existence of at least one NE, converging toward one of the NEs, and the computational complexity of the proposed algorithm are discussed. Finally, simulations are presented to demonstrate the ability of the proposed algorithm, assuming the systems using *IEEE802.15.4/Zigbee* and *IEEE802.11af*.

Keywords Coalition formation game · Game theory · Multi-channel spectrum sensing · Wireless sensor network

1 Introduction

Spectrum sensing is a functionality that is done to monitor the spectrum activities of primary users (PUs), i.e. detecting the frequency channels that are utilized by the PUs [1]. This functionality is performed by battery-equipped low-cost frequency sensors in a wireless sensor network (WSN) [2]. Since one sensor might not be able to reliably detect weak signals due to fading or shadowing effects, cooperative spectrum sensing (CSS) is proposed based on the combination of the detection results, from spatially distributed multiple sensors [1].

The limited energy of sensors due to their battery size and weight limitations makes the network lifetime as an important issue in WSN, especially for CSS application. Several methods are reported in the literature to address this issue in the CSS application [3]. An efficient way for extending a WSN lifetime is reducing the number of cooperative sensing nodes meanwhile guaranteeing the sensing quality. This solution can be performed through node selection [4], censoring [5], or voting [6] schemes. Most of the works have been done on energy-efficient node selection for CSS, focus on energy conservation or lifetime maximization in CSS for sensing only one channel [4, 7]. In multi-channel cooperative spectrum sensing (MCSS)

✉ Ataollah Ebrahimzadeh
abrahamzadeh@gmail.com

Asma Bagheri
bagheri.asma84@yahoo.com

Maryam Najimi
maryam_najimi1361@yahoo.com

[1] Department of Electrical and Computer Engineering, Babol Noshirvani University of Technology, Babol, Iran

[2] Department of Electrical Engineering, University of Science and Technology of Mazandaran, Behshahr, Iran

scenario, where the under monitor bandwidth is composed of many non-adjacent channels, the node selection becomes more complicated, because the sensors cannot sense more than one channel in a sensing duration. This challenge is caused by the sensors' hardware limitations, i.e. the low-cost sensors cannot sense more than one channel simultaneously, because they do not have high-speed Analogue-to-Digital-Convertors (ADCs) which need high-power batteries [8, 9]. In [10], a survey was provided on various aspects of cognitive multi-channel wireless sensor networks. A practical method for the MCSS is monitoring the channels separately [9]. In this method, sensors cooperate to sense all the channels. However, most of the works have been done on node selection for MCSS focus on detection quality or throughput maximization [11–13]. For instance, an algorithm for scheduling the sensing nodes into different groups to perform MCSS was proposed in [11], but in their algorithm, the main goal was throughput maximization, therefore, all the sensing nodes do sensing in each of the network lifetime durations. Clustering is another method for selecting the sensing nodes for MCSS, e.g. [12] introduced a clustering scheme for energy management in WSNs. It has been shown that if a cluster head and its member nodes collaboratively perform channel sensing, it provides more accurate channel sensing and needs less energy than CSS in a non-clustered WSN. This scheme saves energy because of reducing the transmission energy of sensors (they send their sensing results to the cluster head behalf sending to a fusion center (FC)). However, in the proposed scheme, in each of the network lifetime durations, all the nodes perform sensing which leads to higher false alarms and more energy. Another energy-efficient MCSS scheme was proposed in [13], in which first, sensors are allocated to different channels based on their detection probability, and then the thresholds of energy detectors are justified to conserve energy. In this scheme, all the nodes perform sensing in each of the network lifetime durations, too. However, in WSN, the lifetime extending problem needs to balance the energy consumption between sensors besides energy conservation and to maintain the detection quality. Therefore, solving the problem by channel and sensor selection is an NP-complete problem, in a WSN performing MCSS while assuming the limited ability of sensors in sensing more than one channel. In [14], for solving the problem, a sensor selection algorithm was proposed. However, the method needs to determine the priority of channels for assigning cooperative sensors, which makes the solution very complex. In this paper, we propose an efficient tool to solve the sensor assignment problem with lower computational complexity.

Game Theory (GT) is a powerful method, in allocation issues, for modeling the interaction and cooperation between players, especially in multi-user scenarios [15, 16]. Making coalitions using GT, i.e. coalition formation game (CFG), for cooperation in the multi-user networks has been a high-interest topic. CFG is exactly good at making the game players (which are the sensors in the paper) cooperate to obtain the maximum utility [17]. Several works have been done addressing its application for solving problems in MCSS scenarios, too. For instance, in [18], CFG was used to solve the problem of throughput maximization by making overlapping coalitions of the sensing nodes. In their proposed scheme for MCSS, it was assumed that every sensing node has multiple EDs and can sense more than one channel, simultaneously, which is costly and impossible for some limited applications. It has also been considered that nodes sense none of the channels if they have no data for transmission; however other nodes should sense all the channels simultaneously; hence this scheme is not energy efficient. Another scheme for MCSS using CFG was proposed in [19] which investigated the problem of throughput maximization. In the proposed scheme, sensing nodes make coalitions and every coalition cooperatively senses a part of the spectrum; if it was detected idle, in the coalition, a randomly-selected node sends data on the idle detected part of the spectrum. Although this scheme assumed the limitation of nodes in sensing more than one channel and used the CFG for making disjoint coalitions, the energy of nodes was not considered and all the sensors perform sensing in each of the network lifetime durations. In [20], CFG was used for clustering and determining cluster heads and fusion rules for combining the detection results of cooperative nodes to reduce the detection errors. Also, [21] is another recent work that has used CFG to improve the detection performance of MCSS, where the sensing nodes select PUs, form groups, and share their sensing results to provide the maximum communication opportunity for themselves. In [22], a scheme for MCSS combining CFG with a genetic algorithm was proposed to select the optimal coalition heads to improve the detection performance. In this scheme, the consumption energy decreases because only the coalition head sends the combined detection results to the FC, on behalf of all the coalition members. In [23], using CFG, a distributed cooperative spectrum sensing and accessing scheme was proposed which takes into account both the sensing accuracy and energy efficiency in the utility of coalitions. The scheme needs selecting a decision node for every coalition and sharing the received SNR of the nodes in each coalition. However, the utility of each coalition was defined as the ratio of the opportunistic usage of the channel to the energy consumption for sensing the channel. In fact, in their proposed scheme, sensing nodes are partitioned into groups that provide higher throughput for them, and also all the nodes do sensing simultaneously.

However, in [22, 23], extending the network lifetime was not the main goal and all the nodes do sensing in each of the network lifetime durations, that it is not necessary for some applications, because it increases the false alarm probability.

In [24], to enhance the network lifetime, a framework combining hierarchical clustering and CFG was introduced. In the proposed framework, first, sensors are classified into different clusters and a cluster head is selected for each cluster. Then, in each cluster, sensors make coalitions using CFG. In each coalition, only one node senses the channel. In the framework, the coalition formation is based on the node's probability of accessing the channels, although the proposed framework reduces the number of sensing nodes and thereupon it extends the network lifetime. Also, in [24], each cluster selects a channel for sensing based on the probability that the selected node in the coalition can access the channel, i.e. the consumption energy, remaining energy and detection quality of nodes are not considered, which is completely different from our problem. The proposed framework in [24] is also introduced for a large WSN with the capability of frequency re-use, which is completely different from our assumed system model. In [25], CFG was used for selecting sensing and transmission nodes in a multi-channel cognitive radio network. In the proposed scheme, the nodes, that sense a particular channel with non-identical sensing times, form a coalition in which a coalition head is selected to determine whether or not the channel is idle. The coalition head transmits over the sensed channel if the channel is idle. The main goal of the scheme is improving the average throughput meanwhile reducing interference with PUs and energy efficiency. However, in the scheme, all the nodes perform sensing; thus, the energy efficiency is achieved due to determining variable sensing times, not at reducing the number of sensing nodes.

Because of the efficient results of using CFG in cooperative problems, in this paper, the method is investigated. We propose a novel sensor selection algorithm based on CFG for lifetime maximization of WSNs performing MCSS when the sensing quality and limited ability of sensors in multi-channel sensing are considered, which is different from other existing MCSS problems. The main contribution of this paper is:

1. In this paper, we notice the energy consumption balancing in WSN besides energy conservation, while most of the works focused on reducing the energy or increasing the throughput in these networks, at the time this paper is being written. A novel and low-complexity algorithm based on CFG is proposed for sensor assignment in a WSN, while some constraints on the sensing quality are satisfied. In the proposed algorithm, the sensors play a coalition game and calculate the reward of joining every coalition for sensing the channels. Then, they decide to join a coalition for sensing a channel or to sense none of the channels for energy conservation. However, the assumed network model and the objective problem, in this paper are completely different from the existing works which have used CFG [18–25], and also from other existing studies on MCSS issues [11–13]. The network model and the objective problem in [14] is similar to this paper, but the convex method was used to solve the problem in [14]; the method needs to determine the priority of channels for assigning cooperative sensors, which makes the solution very complex. Here, CFG is used to solve the problem and the computational complexity of the proposed algorithm is discussed and its efficiency is compared through simulations.
2. Designing a proper utility function is a key challenge in a CFG. Another novelty of this paper is finding a proper utility for describing the coalition's values for convergence of the proposed algorithm toward an efficient Nash-equilibrium (NE). The next important issue in CFGs is how to divide the benefits of the coalition between the members. In this paper, we use Shapley value and define a reward function for each player to move them toward selecting a coalition with the most total benefit for the system.
3. However, showing that the proposed game has a NE and also analyzing the convergence of the proposed game toward a stable NE is not easy to prove because of the high number of players and choices for them. Therefore, another novelty of the paper is that we model the proposed CFG to the ordinal potential game (OPG), with defining a proper potential function for the proposed game, to show that the proposed CFG reaches a stable NE. Since OPG is an exhaustively analyzed game, we can use its property to prove that the proposed algorithm adjusts the behavior of sensors toward Nash-stable coalitions with balancing energy consumption in the network, a reasonable detection probability, and a low false alarm probability. Also, the computational complexity order of the proposed algorithm is discussed. Simulations are conducted for performance evaluation, which demonstrates the efficiency of the proposed algorithm in lifetime extension in comparison with the similar existing works [13, 14]. We assume two notable applications for evaluating the proposed algorithm: *IEEE802.15.4/Zigbee* and *IEEE802.11af*. Power-saving and extending the network lifetime are from the main issues of these applications.

The proposed algorithm in the paper can be used for extending lifetime of a WSN performing MCSS with applications that the recharge of sensors is difficult or impossible, such as military networks or out of rural-area WSNs, e.g. a WSN performing CSS for sensing TV-white space spectrum for wireless regional area network (WRAN). The rest of this paper is organized as follows: In Sect. 2, the system model is expressed. Section 3 formulates the lifetime maximization problem and describes the problem as a coalition game. In Sect. 4, the GT-based algorithm is proposed. Then, the convergence of game to the Nash stability is proved and the complexity of the proposed algorithm is discussed. The simulation results are presented in Sect. 5. Finally the conclusions are presented in Sect. 6.

2 System model

This paper focuses on the centralized MCSS, i.e. a common kind of cooperation in which there is a fusion center (FC) that receives the local sensing results from spatially distributed nodes and makes the final decision about the status of the channels [26]. We consider a WSN composed of N frequency sensors and an FC as shown in Fig. 1. The under monitor frequency spectrum is composed of M channels with the same bandwidth, which are used by M PUs. It is assumed that every PU can use only one of the channels.

It is assumed that each sensor cannot sense more than one channel at the same time, but a sensor can sense different channels in different sensing times. For this scheme, every sensor is equipped with a receiver circuit which composed of a synthesizer, a narrowband filter, and an energy detector. The simple and low-cost receiver circuit is plotted in Fig. 2. In this receiver, the synthesizer tunes frequency based on the decision of the sensor to the center frequency of the selected channel (for instance the m-th channel is selected for sensing by sensor n), such that a sensor senses only one channel in every sensing period. If a sensor selects none of the channels for sensing, it turns off its sensing module at the sensing period. Therefore, for MCSS, a scheme is proposed in which the WSN can simultaneously sense all the channels. Energy detector (ED) is used in the receiver of sensors due to the detector low complexity.[1] The signal energy in the m-th channel is measured by the n-th sensor as: $T_{m,n} = \frac{1}{K}\sum_{k=1}^{K}\left|x_{m,n}(k)\right|^2$, in which $x_{m,n}(k)$ denotes the k-th sample of the received signal from the m-th channel that is observed by the n-th sensor, and K is the number of samples which is calculated as f_s, where f_s is the Nyquist sampling rate of the ED according to the under-monitor channel bandwidth, and δ is the sensing time. Then, the signal energy is compared with a threshold, γ, to generate one decision bit, i.e. $D_{m,n}$. This bit shows the detected status of the channel by the n-th sensor, as follows:

$$\begin{cases} D_{m,n} = 1; & if\ T_{m,n} \geq \gamma \\ D_{m,n} = 0; & if\ T_{m,n} < \gamma \end{cases} \tag{1}$$

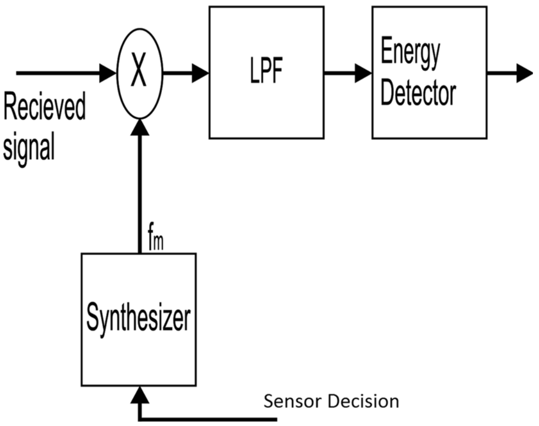

Fig. 1 A sample of system model

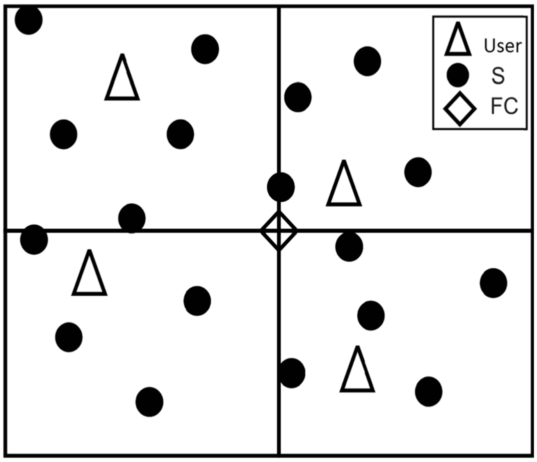

Fig. 2 The receiver circuit of every sensor

When this decision bit is *one*, it states that the channel is busy, and when the decision bit is *zero*, it means that the channel is idle. Two hypotheses for the state of every channel are defined. The first, i.e. $H_{0,m}$, says that the m-th channel is free, and the second, i.e. $H_{1,m}$, says that the channel is busy. Hence; the $x_{m,n}(k)$ is written as:

$$\begin{cases} H_{1,m}; & x_{m,n}(k) = g_{m,n} \cdot s_m(k) + v_{m,n}(k) \\ H_{0,m}; & x_{m,n}(k) = v_{m,n}(k) \end{cases} \tag{2}$$

in which the k-th sample of the PU signal on the m-th channel is denoted by $s_m(k)$, that is assumed to be an i.i.d

[1] This assumption does not have an effect on the proposed algorithm, and it can be extended to the other existing proper detectors, easily.

random Gaussian process with *zero* mean and variance $\sigma_{S_m}^2$. The additive white Gaussian noise with *zero*-mean and variance σ_0^2 is denoted by $v_{m,n}$. It is assumed that s_m and $v_{m,n}$ are independent. The channel gain between the m-th PU and the n-th sensor is denoted by a random variable $g_{m,n}$. We consider the path loss, Rayleigh fading and shadowing effects to model the PU-sensor channel [14]. Hence; under the assumed channel model, the n-th sensor received signals to noise ratio (SNR) from the PU which uses the m-th channel, i.e. $SNR_{m,n}$, is obtained as [7]:

$$SNR_{m,n} = \frac{p_t |g_{m,n}|^2 \sigma_{S_m}^2}{\sigma_0^2} \tag{3}$$

in which p_t denotes the transmission power of PUs. It is assumed that every sensor knows the instantaneous received SNR from all the channels.[2]

There are two important metrics for the spectrum sensing quality of a sensor which are called false alarm probability (FP) and detection probability (DP). The FP states the probability that a sensor decides a channel is busy when the channel is free. The DP states the probability that a sensor detects a PU signal if the PU is transmitting actually. Therefore, the local FP and DP of a sensor about the status of the m-th channel are, respectively as [7]:

$$P_{f_{m,n}} = P(D_{m,n} = 1|H_{0,m}) = Q\left(\left(\frac{\gamma}{\sigma_0^2} - 1\right)\sqrt{\delta f_s}\right) \tag{4}$$

$$\begin{aligned} P_{d_{m,n}} &= P(D_{m,n} = 1|H_{1,m}) \\ &= Q\left(\left(\frac{\gamma}{\sigma_0^2} - SNR_{m,n} - 1\right)\sqrt{\frac{\delta f_s}{2SNR_{m,n} + 1}}\right) \end{aligned} \tag{5}$$

Here $Q(.)$ denotes the complementary distribution function.

As mentioned, due to fading or shadowing effects and also limited sensing range of sensors, selecting only a single sensor for sensing every channel may fail to detect the PU signals correctly [9]. To improve the sensing quality, we assume a centralized CSS scheme, in which all sensing nodes send their decision bits to an FC, and the FC uses the logic OR rule to fuse the decisions of sensors to make the final decision about busy or idle status of the channels (For a comparison on the CSS methods and the fusion rules for CSS see [26]). Because of hardware limitations of sensors in sensing more than one channel, in every sensing duration, all the nodes should be classified into disjoint groups and every group of sensors senses only one channel. In this paper, the cooperative sensors for sensing the m-th channel are inserted in a set J_m, where the sets are non-overlapping for $m = 1,\ldots,M$. After sensing duration, the selected cooperative nodes send their decision bits to the FC. Thus, the global detection probability (GDP) (P_{d_m}) and the global false alarm probability (GFP) (P_{f_m}) for the m-th channel are respectively calculated as:

$$P_{d_m} = 1 - \prod_{n \in J_m} \left(1 - P_{d_{m,n}}\right) \tag{6}$$

$$P_{f_m} = 1 - \prod_{n \in J_m} \left(1 - P_{f_{m,n}}\right) \tag{7}$$

The Eqs. (6) and (7) shows that when the number of cooperative nodes increases, both the GDP and GFP increases. The higher GDP leads to reduce the data loss of the occupied channels, but the higher GFP leads to lower ability to detect empty channels. Therefore, if all sensors participate in sensing, it leads to high energy consumption and more GFP about the channels without increasing significant GDP [1]. In this paper, sensor assignment is done until reasonable GDP is provided for sensing the channels; hence the sensors which decide to sense none of the channels are inserted in a set J_0.

The energy consumption of a sensor in a WSN mainly depends on two factors: the sensing energy, and the transmission energy. Therefore, the energy consumption of the n-th sensor is calculated as [7]:

$$EC_n = E_s + E_{t_n} \tag{8}$$

The sensing energy, i.e. E_s, is the amount of energy that a sensor consumes to sense a channel and to make its decision about the status of the channel (for synthesizer and detector circuit). It is assumed that E_s is constant, and it is the same for all sensors (It is assumed that all sensors, which decides to participate in CSS, pick samples with a similar sampling rate in a fixed sensing duration, although the results can be extended to the different assumption such as different sensing times or different bandwidth for different channels). The E_{t_n} denotes transmission energy which is the energy to send the decision bit to the FC reliably, and it is calculated as [7]:

$$E_{t_n} = E_{t-elec} + e_{amp} \cdot d_{0,n}^2, \tag{9}$$

in which E_{t-elec} is the energy used for the electronic circuits of a sensor transmitter, e_{amp} is the amplifying coefficient, and $d_{0,n}$ is the distance between the n-th sensor and the FC.

In the next section, with the aim of lifetime maximization, a sensor selection problem is raised for MCSS under GDP and GFP constraints.

[2] There are studies on the SNR estimation of nodes in spectrum sensing networks, but it is not the goal of this paper. Although the assumption seems unrealistic for some scenarios, it does not affect the proposed algorithm, and the algorithm steps can be done based on the average SNR or Packet Reception Ratio, similarly [26].

3 The GT-based problem

This paper investigates lifetime maximization for a WSN performing MCSS. There are different definitions for the lifetime of a WSN based on the network application [27, 28]. One of the most used definitions of a WSN lifetime is the time at which a group of sensors runs out of energy [14]. We here define the network lifetime as:

Definition 1 (The WSN lifetime) *The lifetime of the frequency sensor network is the time until the number of live sensors drops below* $\mathcal{L} \cdot \mathrm{N}$ *where* $0 < \mathcal{L} \leq 1$, [27].

The reason behind this definition is that when only some of the nodes have enough energy, the duty of spectrum sensing can be performed. In this paper, a lifetime maximization scheme is proposed based on sensor selection. This scheme selects sensors for sensing different channels in different sensing durations, under some detection quality constraints. The reason behind such constraints is that the purpose of this network is to monitor the spectrum activities. Therefore, the GDP and GFP are limiting the sensor selection problem. A high GDP must be achieved to reduce the amount of interference with the PUs. Also, when a false alarm occurs, it is decided that a PU is transmitting, while the channel is free, therefore a high GFP is highly undesirable. In an ideal CSS, it should be GDP = 1 and GFP = 0 for every channel, but such perfect CSS is impossible in actual systems because, from (6) and (7), it is concluded that increasing the GDP leads to increase the GFP. Therefore, we define an upper-bound for GFP and a lower-bound for GDP denoted by α and β, respectively. The lifetime maximization problem is presented as:

$$problem\,1 \quad \max_{\mathrm{J_m}} lifetime \quad \forall m \in \{0, \ldots, M\} \qquad (10-1)$$

subject to;

$$1. \quad P_{f_m} \leq \alpha \quad \forall m \in \{1, \ldots, M\} \qquad (10-2)$$

$$2. \quad P_{d_m} \geq \beta \quad \forall m \in \{1, \ldots, M\} \qquad (10-3)$$

$$3. \quad J_m \cap J_{m'} = \emptyset, \bigcup_{m=0}^{M} J_m = \{1, \ldots, N\} \quad \forall m, m' \in \{0, \ldots, M\},\ m \neq m' \qquad (10-4)$$

The third constraint states that one sensor cannot sense more than one channel in a sensing duration. We use the "max–min" method for optimizing the network lifetime [28]. In this method, the minimum remaining energy of sensors is maximized. Thus, the sensors that have the lower remaining energy are not being selected for sensing. It leads to the remaining energy level of sensors keep balanced, and consequently to extend the network lifetime. The remaining energy of sensors and the minimum of remaining energy of sensors are denoted by E_n and E_{th}, respectively. Now, the problem is written as:

$$problem\,2 \quad \max_{J_m}\left\{\min_{n}\{E_n\} = E_{th}\right\} \quad \forall m, n \qquad (11-1)$$

subject to;

$$1. \quad P_{f_m} \leq \alpha \quad \forall m \in \{1, \ldots, M\} \qquad (11-2)$$

$$2. \quad P_{d_m} \geq \beta \quad \forall m \in \{1, \ldots, M\} \qquad (11-3)$$

$$3. \quad J_m \cap J_{m'} = \emptyset, \bigcup_{m=0}^{M} J_m = \{1, \ldots, N\} \quad \forall m, m' \in \{0, \ldots, M\},\ m \neq m' \qquad (11-4)$$

It is noted that this is a non-convex and NP-complete problem, and finding the optimal solution needs an exhaustive search with high complexity. In [14], a method for solving this problem was proposed. However, it needs to determine the priority of channels for assigning cooperative sensors, which makes the solution very complex. We seek a more efficient method that will solve the problem with lower computational complexity. GT is useful in analyzing the mutual interactions among multi-users [17]. Because of the efficient results of using GT in multi-channel problems [15, 25], the method is investigated in this paper. Since the assumed problem is a cooperative problem and a sensor may not sense a channel with enough detection probability, therefore, we use a kind of cooperative game. CFG is a good cooperative game at making the game players (which are the sensors in the paper) cooperate to obtain the maximum utility for the system [17]. This paper proposes that sensors play a CFG and based on their reward decide to join a coalition to cooperatively sense a channel or do not participate in the MCSS. The game is denoted as:

$$\mathcal{g} = \left\{\mathcal{N}, \{\mathcal{A}_n\}_{n\in\mathcal{N}}, \{u(J_m)\}_{J_m \subset \mathcal{N}}\right\} \qquad (12)$$

in which $\mathcal{N} = \{1, \ldots, N\}$ denotes the players of the game which are the sensors. $\mathcal{A}_n = \{a_{n0}, a_{n1}, \ldots, a_{nM}\} = \{0, 1, \ldots, M\}$ is the set of available actions of player n ($a_{n0} = 0$ means sensing no channel and $a_{nm} = m$ means participating in CSS of the m-th channel). An action profile $(a_1, \ldots, a_N)$ shows the selected actions by the N players, when $a_n \in \mathcal{A}_n$. In every CFG, there is a utility function that describes the value of every possible coalition. The value of a coalition J_m in sensing the channel m is denoted by $u(J_m)$. In the game, we want to select sensors for sensing M channels such that the detection quality of sensing satisfies the constraints (11-2) and (11-3). For reducing the energy consumption and the false alarm probability, some sensors are selected and others turn off their sensing module for a sensing-time-duration. Therefore, the number of coalitions is fixed and equal to $M + 1$: M coalitions for sensing M

channels and one coalition for the sensors which sense none of the channels (coalition *zero*). Therefore, the number of coalitions is fixed and equal to $M+1$. Also, from the constraint (11-4), the coalitions are mutually disjoint. In this paper, $u(J_m)$ must be an increasing function of GDP and E_{th} within the coalition J_m and a decreasing function of the GFP and the energy consumption of sensors in the coalition. Also, it is assumed that if a sensor decided to sense no channels, its utility would be *zero*, because the sensors that select actions a_{n0} (we name it as coalition *zero*), receive no gain meanwhile spend no cost. A suitable utility function for a coalition J_m for cooperatively sensing channel m is proposed as (See "Appendix 1"):

$$u(J_m)=\begin{cases} if\ P_{d_m}(J_m)\geq\beta,\ P_{f_m}(J_m)\leq\alpha & \begin{cases}1+E_{th}; & if\ E_{th}>E_{th,0}\\ \dfrac{max(E)-E(J_m)}{max(E)-min(E)}; & if\ E_{th}=E_{th,0}\\ \vartheta\dfrac{max(E)-E(J_m)}{max(E)-min(E)}; & if\ E_{th}<E_{th,0}\end{cases}\\ if\ P_{d_m}(J_m)<\beta\ or\ P_{f_m}(J_m)>\alpha & \begin{cases}\dfrac{-min(E)}{max(E)}\dfrac{E_0-E_{th}}{E_0}; & if\ E_{th}>E_{th,0}\\ \dfrac{-E(J_m)}{max(E)}; & if\ E_{th}=E_{th,0}\\ -1-\dfrac{E_0-E_{th}}{E_0}-E(J_m); & if\ E_{th}<E_{th,0}\end{cases}\end{cases} \quad (13)$$

in which, $E(J_m)$ denotes the sum of energy consumption of sensors in the coalition J_m, $min(E)$ denotes the minimum energy consumption for sensing a channel which is equal the energy consumption of only one node with the lowest energy consumption. The maximum energy consumption is for the case that all the sensors sense a channel, which is denoted by $max(E)$. The initial energy and the minimum remaining energy of sensors in the previous step of the game are denoted by E_0 and $E_{th,0}$, respectively. The coefficient $\vartheta=\frac{0.5min(E)}{max(E)-min(E)}$.

In this paper, it is assumed that each sensor n can select a coalition in two possible ways: The first is adding to the coalition when all the other nodes don't change their coalitions. The second is adding to the coalition when a node n_0 removes from the coalition, it means that the n_0 is replaced by the player n. In this way, the removed sensor goes to coalition *zero*. In both ways, the player n receives a reward or penalty for selecting the coalition J_m. We determine the reward based on the Shapely value [29]. This reward value determines how much benefit results in a coalition value if the player adds to the coalition. For the first possible way of the assumed game, the reward is calculated as:

$$Re_n(J_m)=u(J_m+\{n\})-u(J_m) \quad (14)$$

in which $u(J_m)$ is the utility of the coalition J_m before the n-th sensor is added to, and $u(J_m+\{n\})$ denotes the utility of the coalition after the n-th sensor is added to. For the second possible way of the assumed game, the reward is calculated, similarly, as:

$$Re_n(J_m)=u(J_m+\{n\}-\{n_0\})-u(J_m) \quad (15)$$

when $u(J_m+\{n\}-\{n_0\})$ denotes the utility value of the coalition when a sensor n_0 in the coalition is replaced by the sensor n. If the sensor decision increases the coalition value, it gains a positive reward. If the sensor decision decreases the coalition value, it gains a negative reward which means penalty.

4 The GT-based sensor assignment algorithm

In this section, an algorithm is proposed for the CFG, with the goal of lifetime extension of MCSS in the WSN. Also, the main properties of the algorithm are discussed.

4.1 The GT-based coalition formation algorithm

In each period of network lifetime: first live sensors select channels for sensing or take action a_{n0}, randomly. Then, the live sensors start to play the game such that: every node calculates its reward from joining in any of the coalitions and also it calculates its reward from replacing each sensor participating in the coalitions. If there is a coalition which its utility is more than the current coalition the sensor changes its decision because this change makes a positive reward for the sensor, based on (14) or (15). Playing the game continues to reach a stable state. In the stable state, no sensor changes its decision. After the game reaches a

Fig. 3 The GT-based algorithm

```
While    (live sensors number >= 0.25*N)% lifetime loop
Step1: live sensors randomly select channels.
Step2: while (at least one of the latest $N$ players have changed their decision) % This is
the stop criteria.
    1. a live sensor $n$ is selected randomly
    2. The sensor $n$ calculates his reward from adding to any coalition, from (14).
    3. The sensor $n$ calculates his reward from replacing any of the sensors in the current
       coalitions, from (15), from coalition1 to M.
    4. The sensor $n$ compares the rewards and decides to sense or not, and if it senses, it
       joins which of the coalitions (in this way, it selects the highest reward).
    5. Update the $E_{th,0}$ .
    end
Step3: Update the remaining energy of sensors, the live sensors and life numbers.
end % for lifetime loop
```

stable state, the remaining energy of sensors is updated, and the FC orders the selected nodes to sense their allocated channels. If the algorithm converges to an acceptable answer that satisfies all the constraints, we calculate the iteration as a successful period of the lifetime. Then, the numbers of off sensors are calculated. The program continues until the off-sensor-number is higher than $(1-\mathcal{L})N$. The Fig. 3 presents a pseudo-code for the algorithm.

It is concluded that assuming the algorithm, sequences of improvement steps of the game do not run into cycles, but reach a Nash equilibrium (NE) after a finite number of steps [17]. An action profile $(a_1, \ldots, a_N)$ is a strategy NE if and only if no player can improve its utility by deviating unilaterally [30]. In the following subsection, we describe the concept of converging toward NE for the proposed algorithm.

4.2 Proof of converging toward NE

For the assumed problem, the straightforward analysis of the algorithm convergence toward NE is complicated; therefore, we use the ordinal potential game (OPG), to show that the proposed CFG reaches a stable NE, in “Appendix 2”. Here, we use a potential approach for analyzing the game. The intuition behind the approach is that it tracks the changes in the rewards when some player deviates, without taking into account which one. Also, there are some other reasons which make interest in the potential games: (1) in the finite potential games, all executions of the players' dynamics terminate; (2) at each step of the players' dynamics the potential strictly increases; (3) each finite potential game has NE [31]. To this end, in this paper, a potential function for the game is defined and the properties of the potential game are used. First, the definition of a special type of potential game, ordinal potential game (OPG), is given. Then, it is proved that the game is an OPG. Finally, we prove the proposed game converges to a NE.

Definition 2 (Ordinal potential game) *A game* $\mathcal{G} = \left\{\mathcal{N}, \{\mathcal{A}_n\}_{n\in\mathcal{N}}, \{u(J_m)\}_{J_m\subset\mathcal{N}}\right\}$ *is an OPG if there is a function* Φ *such that every move of the players to increase their selfish benefit increases the potential function, too* [32].

Theorem 1 *The proposed coalition game is an OPG.*

Proof See “Appendix 2”. □

Theorem 2 *The proposed coalition game has at least one NE and converges to a NE.*

Proof Here, we use the following properties of OPGs: (1) at each step of the players' dynamics the potential of OPGs strictly increases; (2) OPGs do not take a repeating circle and stop at one point, certainly; (3) if the potential function of OPGs has a limited upper bound, these games converge to a NE in a finite number of steps. The proof of the mentioned properties of OPGs was exhaustively presented in [31, 33]. To this end, in “Appendix 1“, in (22), we define the potential function of the proposed game as the sum of the utilities of all the coalitions. Since the utility function, defined in (13), has a finite upper bound (the limited measure of the utility function is discussed in “Appendix 1“), the defined potential function has a limited upper bound. Therefore, according to the above-mentioned properties, the CFG proposed in the paper has at least one NE, also, it is concluded that the proposed game converges to a NE in a finite number of steps. In our model, Nash equilibria are the fixed points of the dynamics defined by improvement steps of sensors. □

4.3 Analysis of the stability concept of the proposed game

In CFGs, first, a proper utility function for each coalition should be defined to represent the amount of profit that each coalition receives. This function must be defined so that the players tend to make coalitions [17]. We defined the utility function of the game in (13) more discussion on the proposed function is presented in "Appendix 1". The next important issue is how to divide the benefits of the coalition between the members [17]. For the issue, there are many solutions, such as core, Shapley value, and nucleus, etc. [17]. The core solution for the assumed problem of the paper is very complicated and one does not necessarily find a rational allocation of sensors. The nucleus solution is also much more sophisticated than Shapley, and it is difficult to deal with at a high number of players and choices. Therefore, the Shapley solution is used in this paper for dividing the benefits of the coalitions between the members. Shapley value should take into account the player's contribution to the benefits of their coalition. In this paper, based on Shapley value, in (14) and (15), we defined a reward function for each player to move them towards selecting a coalition with the most total benefit for the system. Since the straightforward analysis of the proposed CFG for the assumed problem is complicated, we defined a proper potential function in (22) and used the OPG properties to show that the reward function (which was based on the Shapley value) moves the game to a stable NE, in "Appendix 2".

4.4 Analysis of computational complexity of the proposed algorithm

Now the computational complexity of the proposed algorithm, in each period of network lifetime, is described. In an iteration of the game, a sensor is selected for playing when other nodes remain on their last decision. The selected sensor calculates the following metrics:

1. The sensor calculates his reward from adding to any coalition, from (14): To this end, first, the sensor calculates the utility of the M current coalitions, without itself [$u(J_m)$ is the utility of the coalition J_m before the n-th sensor is added to, and it is calculated from (13) for M coalitions, with a computational complexity of $O(M)$]. Next, it calculates the utility of the M coalitions, when it joins them [$u(J_m + \{n\})$ denotes the utility of the coalition J_m after the n-th sensor is added to, and it is similarly calculated from (13) for M coalitions, with a computational complexity of $O(M)$]. Then, it calculates its reward from joining all the M coalitions, from (14), with a computational complexity of $O(M)$. Therefore, this step has a computational complexity of $O(3M)$.
2. The sensor calculates its reward from joining each of the coalitions from (15), when another sensor is removed: Here, the number of calculations is equal to the number of sensors that are in coalitions for sensing a channel, in the last step of the game. Because the number of sensors is not a fixed number in different steps of the game, we consider the highest possible limit for the number of computations. If no sensor is in coalition *zero*, the numbers of sensors that can be replaced with the player will be $(N-1)$. Also, it is noted that the number of live sensors reduces from N to $\mathcal{L}\ N$ during the network lifetime. Therefore, in every step of the game, at most, $(N-1)$ times the reward function from (15) is calculated. Since calculating the reward function from (15), for a sensor replacing with the player is $O(2)$, this step has a computational complexity of $O(2(N-1))$.
3. The player selects the best strategy (which provides the highest reward) from the all possible strategy space, with a computational complexity $O(1)$.

Therefore, the computational complexity for each player is $O(max(2(N-1),3M))$. Because in WSN the number of the sensors is more than the number of channels, the computational complexity is $O(2(N-1))$, for each player in each period of network lifetime.

5 Simulation results

This section shows the simulation results using MATLAB. Monte-Carlo method is used with 1000 iterations. We designed simulation environments with fixed and variable parameters. The parameters that do not affect the performance of the algorithm are assumed fixed, mainly including the PU's signal parameters such as the signal modulation, sampling frequency of the signal (it is equal to the Nyquist frequency) and the threshold level of the detectors (it is assumed as a coefficient of the noise power). The variable parameters are those that affect the algorithm performance of such as the number of sensors or channels. Here, the network lifetime is defined as the number of iterations, in which more than 25% of sensors are kept alive, multiplied by the duration between the sensors selections (because the duration is assumed fixed for all iterations, we just plot the iterations number).

The performance of the proposed GT-based algorithm (we call it as GT in simulation plots) is compared with a random sensor selection algorithm, a detection based sensor selection algorithm, and a lifetime-based sensor selection algorithm that has been proposed in [14]. The reasons behind these comparisons are: The random algorithm has the lowest complexity order; The detection based algorithm finds acceptable solution, if there is, meanwhile the complexity order of the algorithm is low, also many other algorithms for sensor selection or sensor scheduling for CSS have been proposed on this basis such as [13] (We call it as DBSS[3]); The proposed algorithm in [14] has computational complexity of order of the GT-based algorithm when the order of channels for sensor allocation to them is not determined optimally. The algorithm has a higher order of complexity than the GT-based algorithm if the optimal channel ordering is done (We call it as CONVEX because the algorithm uses the convex optimization method). We provide simulations assuming the following cases for evaluating these algorithms:

5.1 Case study1: IEEE 802.15.4/Zigbee

A square region with a length of 200 m is assumed in which an FC is located in the center, N sensors and M PUs are distributed identically, in which the number of sensors varies from ten to eighty and the number of channels varies from two to eight. The *IEEE 802.15.4/Zigbee* is used for the cognitive sensors [34]. The first part of sensing energy is used in the synthesizer circuit which is equal to 86 nJ [35]. Because the typical circuit power consumption of *ZigBee* is approximately 40 mW, the energy consumed for listening is approximately 40 nJ, for sensing time $\delta = 1\,\mu\text{s}$. The processing energy related to the signal processing part in the decision bit transmit mode, for a data rate of 250 kb/s, a voltage of 2.1 V, and current of 17.4 mA is approximately 150 nJ/bit. Since every sensor sends one bit per decision to the FC, the sensing energy is $E_s = 276\,\text{nJ}$. Assuming a data rate of 250 kb/s and a transmit power of 20 mW, $E_{t-elec} = 80\,\text{nJ}$. The amplifier gain to satisfy a receiver sensitivity of -90 dBm is $e_{amp} = 40.4\ \text{pJ/m}^2$. Also, the threshold level for GDP and GFP are set as $\beta = 0.99,\ \alpha = 0.01$. These parameters and the other used parameters for simulations are presented in Table 1.

Table 1 The values of simulation parameters [34]

$f_c = 2.45$ GHz	$\sigma_z^2 = 3$ db	$\sigma_0^2 = 10^{-11}$ W	$p_t = 20$ mW
$E_0 = 0.2$ mJ	$E_s = 276$ nJ	$E_{t-elec} = 80$ nJ	$e_{amp} = 40.4\ \text{pJ/m}^2$

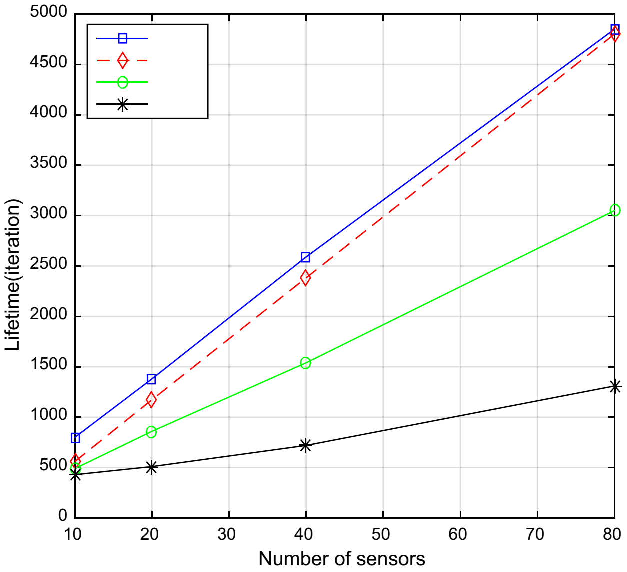

Fig. 4 The average lifetime versus the total number of sensors (M = 4)

In Fig. 4, the network lifetime is plotted versus the total number of sensors. It shows that the GT algorithm extends the WSN lifetime more than the others. In a lower number of sensors, it provides (about 60%) more iterations than the convex algorithm (for N = 10, the lifetime of the GT and the convex algorithms are about 800 and 500 iterations, respectively). When the total number of sensors increases: the network lifetime of all the algorithms increases, because the number of sensors with high DP increases; the number of lifetime iterations for the convex and GT-based algorithms are closer because the difference between consumption energy of sensors decreases and also the ratio of sensors number to the channel number decreases; the number of lifetime iterations for the GT algorithm is much higher than the random and the DBSS algorithms because the algorithms ignore remaining energy and consumption energy of sensors.

For the lifetime maximization, we maximized the minimum remaining energy of the nodes. In Fig. 5, the average E_{th} is shown during the lifetime iterations. The GT and the convex algorithms have the highest E_{th} in similar iteration numbers, because in these algorithms, the remaining energy of nodes is considered in sensor selection. It means that in these algorithms, there is equilibrium between the remaining energies of the sensors while the other algorithms have less E_{th} due to not considering this parameter in sensor selection. It is clear that DBSS has the least E_{th}, because in this algorithm, the sensing nodes with highest DP are selected in sequential iterations of the network lifetime, which leads to fast decreasing in E_{th}. The GT algorithm selects sensors such that the E_{th} of this algorithm is higher than the convex algorithm, since the convex algorithm needs to select the priority of channel for assigning sensors, which is done in an efficient but non-optimal way [14].

Since the aim of the WSN is monitoring the spectral activities of the PUs, so the proper algorithm must extend lifetime while the answer leads to acceptable GDP and

[3] Detection based sensor selection.

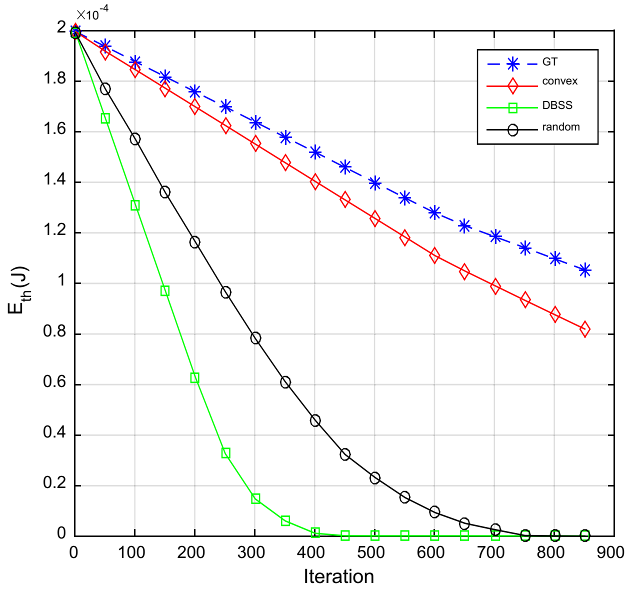

Fig. 5 The average minimum remaining energy of sensors during lifetime iterations (N = 40, M = 4)

GFP. We show that although the proposed algorithm focuses on the energy metrics, it does not ignore the detection accuracy. Therefore, a success rate metric is defined to show the ability of the algorithms in satisfying the GDP and GFP constraints. Figures 6 and 7 show the success rate of the proposed algorithm respect with the convex algorithm, at the different number of sensors and number of channels, respectively. The success rate is calculated as the ratio number of iterations that an algorithm finds an acceptable answer to the total number of iterations that the algorithm provides. Both plots show that the success rate of the GT algorithm is higher than the convex algorithm. At a fixed number of channels, increasing the total number of sensors improves the success rate of both the algorithms that is because of increasing the number of sensors with higher DP for being selected. The similar changes are also seen when the total number of sensors increases, at a fixed number of channels. Also, the difference between the success rates of the algorithms is greater when the total number of sensors decreases at a fixed number of channels, or when the number of channels increases at a fixed number of sensing nodes. The reason for this behavior is that the number of sensors with higher DP increases, hence the probability of existing an acceptable solution is more; therefore, an algorithm, that can extend the lifetime more, would be more successful.

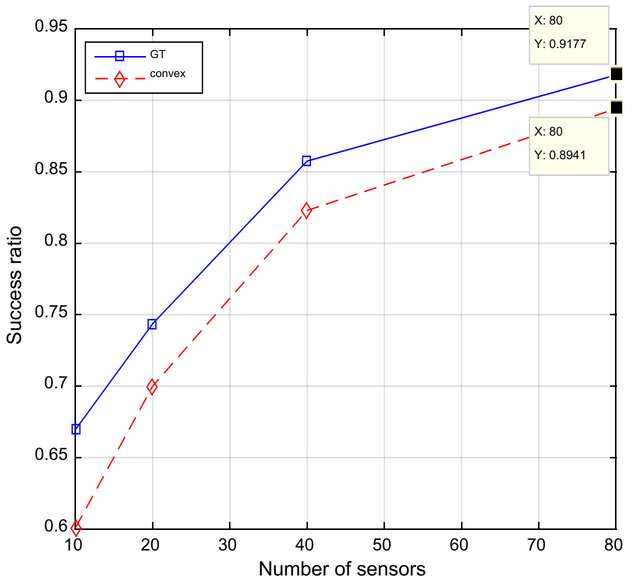

Fig. 6 The success rate versus the number of sensors (M = 4)

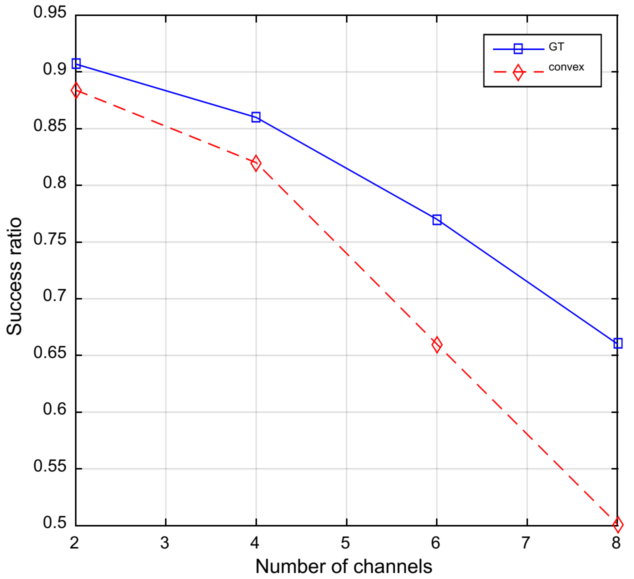

Fig. 7 The success rate versus the number of channels (N = 40)

In Fig. 8 the average success percentage of finding acceptable answers for the algorithms is plotted, respect to the number of iterations. The number of total nodes is considered 10 and 20 for sensing four channels. Since DBSS selects the sensors with maximum DP for spectrum sensing (i.e. if the problem has an answer, DBSS can find it), the algorithm has the highest success percentage in initial iterations. When the number of iterations increases, the remaining live nodes have less DP, or most of the sensors run out of energy; hence, DBSS cannot satisfy the detection performance while GT and convex algorithms select the sensors according to their remaining energy and DP. It means that the GT and convex algorithms increase the network lifetime and more nodes are alive; hence, the remaining live nodes can maintain the GDP constraint for more iterations, i.e. the GT algorithm has a higher success percentage in average. Also, the plot shows that in initial

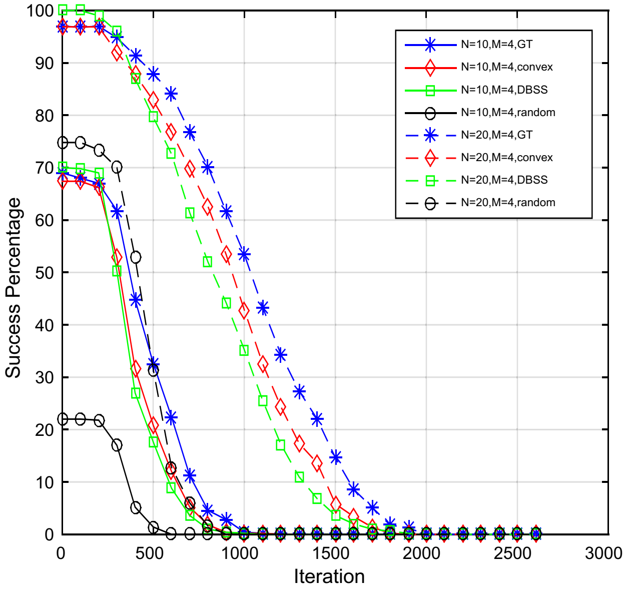

Fig. 8 The percentage of finding acceptable answer at different number of iterations

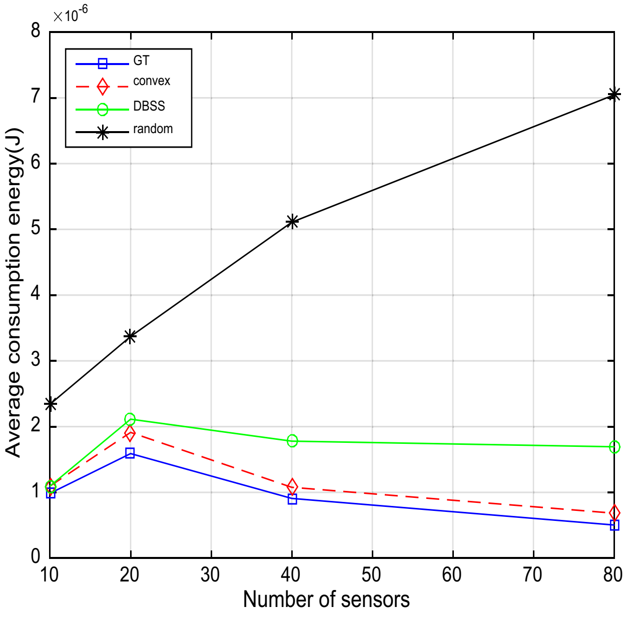

Fig. 9 The average energy consumption at different number of sensors (M = 4)

iterations, the GT and convex algorithms have similar success percentage, but when the number of iterations increases, the GT algorithm has higher success percentage in respect with the convex algorithm, because the convex algorithm assigns sensors with a sub-optimum order of channels and therefore when the number of live sensors decreases, this channel order gets more important.

In Fig. 9, the average energy consumption of sensors for MCSS, in an iteration of the network lifetime, is plotted at the different number of sensors. In this plot, the energy

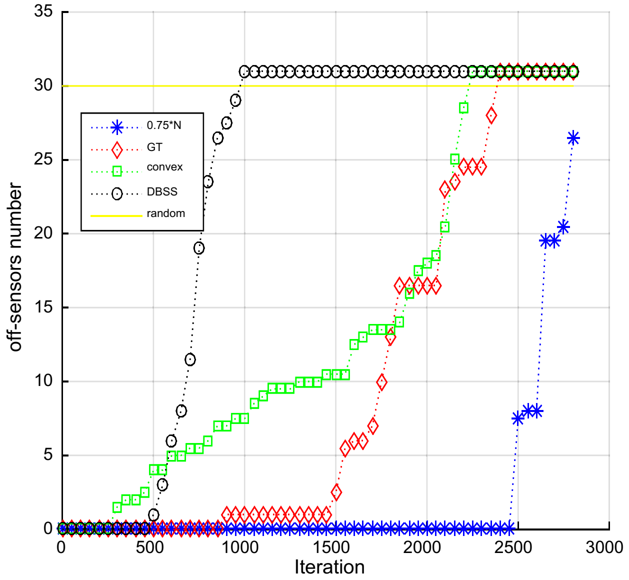

Fig. 10 The percentage of finding acceptable answer at different number of iterations

consumption is averaged over the iterations which satisfy all the problem constraints. We know that the random algorithm selects sensors completely random, therefore when the number of nodes increases, the average energy consumption for the random algorithm increases because it can select more sensors to satisfy GDP constraint. When the number of nodes is 10, the other algorithms have almost the same average energy consumption, because there is a lower number of possible selections. It is noted that the GT algorithm provides the highest lifetime and success ratio with similar average energy consumption. When the number of nodes increases from 10 to 20, there are more sensors for being selected to satisfy GDP constraint, hence, the average energy consumption for the other algorithms increases, but when the number of nodes increases from 20 to higher numbers, the average energy consumption for the algorithms decreases, because there are more sensors with higher DP, and with lower number of selected nodes the GDP constraints are satisfied. For the case that the number of sensors is 20, there is the greatest difference between the GT algorithm and the DBSS and convex algorithms. However, the proposed algorithm provides the highest lifetime and success ratio with the lowest average energy consumption, because it selects the sensors for MCSS according to their consumption and remaining energies and DP.

In Fig. 10, the number of deactivated sensors, i.e. the sensors which don’t have enough energy to sense none of the channels, is plotted during the lifetime iterations. This metric for the proposed GT algorithm is the lowest. It means that the GT algorithm selects sensors with higher remaining energy which further extends the network lifetime. The DBSS is the first algorithm, which its off-sensors number goes up, because the selected sensors may consume high energy while having the highest DP, but in continuing it has gentle slope respect to the other benchmark algorithms. However, the DBSS algorithm provides a lower lifetime than the convex and the GT algorithms, and it is shown that the number of off-sensors is higher than the 75% of sensors, in a lower iteration number than the GT and the convex algorithms.

5.2 Case study 2: IEEE 802.11af

This standard recently gets high attention in cognitive radio system studies. Since the *IEEE 802.11af* system operating the TV white spaces would use frequencies below 1 GHz, this would allow for greater distances to be achieved. However, power-saving is a major issue for *IEEE 802.11af* that will be used for many (Internet of Things) IoT applications. Many of the nodes will need to run using batteries and these need to be able to run for weeks. The proposed algorithm in this paper can be used for this technology, too. For this case, we assume a square region with a length of 1000 m in which an FC is located in the center, and N sensors, which their number varies from 50 to 200, are distributed identically. It is assumed that the $M = 7$ TV channels, with bandwidth 6 MHz regulatory domain, on the frequency spectrum $f = 470–512$ MHZ with the transmitter's power $p_t = 40$ mW are sensed by the sensors [36–38].

In Fig. 11, the network lifetime, assuming the second case, is plotted versus the total number of sensing nodes. In this case, the performance of the GT proposed algorithm is very close to the convex algorithm. However, the proposed algorithm extends the network lifetime more than the random and the DBSS algorithms because the algorithms ignore the remaining energy and consumption energy of sensors. The reason for the similarity of the performance of the GT and convex algorithms is that when the number of sensors goes up (respect with the number of channels) and also the distances between sensing nodes and the PUs is high, the detection quality of sensors and the energy of sensors is more similar. The possible choices of the algorithms are similar. Also, in this case, because the number of sensors increases more than the number of channels the effect of determining the priority of channels for the convex algorithm reduces. Therefore, the lifetime of the algorithms, which both of them consider the energy and the detection probability of sensors, are close.

In Fig. 12, the convergence of the GT algorithm is plotted. The two first plots show the changes in the minimum of remaining energy of all the sensors in the network, and the energy consumption of candidate sensors for the MCSS at different iterations of the game. The next four plots show the changes in the GDP of coalitionist sensors for CSS of different channels at different iterations of the game. It shows that the energy consumption of cooperative sensors during the game is a descending curve. Also, the GDP of all channels reaches a final measure higher than β.

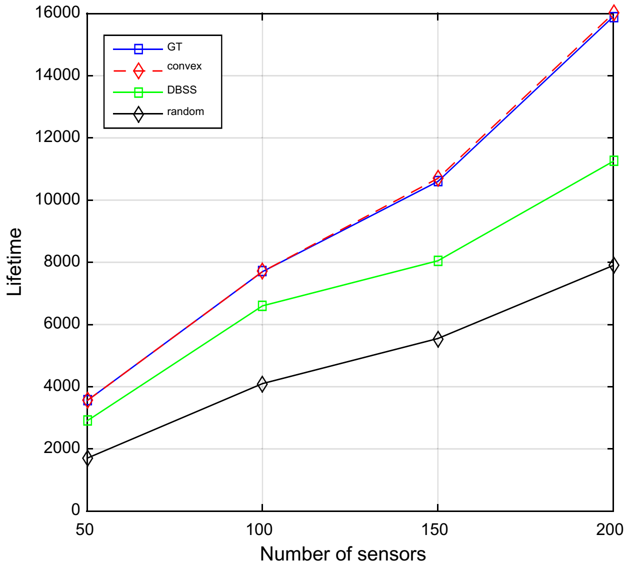

Fig. 11 The average lifetime versus the total number of nodes for case2

The simulations present the efficiency of the game theory in selecting appropriate sensors in a WSN, such that the network lifetime is maximized while the detection quality of selected sensors is acceptable. It is noted that the optimal exhaustive search finds optimal sensors for the problem with the computational complexity order of $O\left((M+1)^N\right)$ which takes a long time in each sensing period. The complexity order of the random and DBSS algorithms is $O(1)$, but their solution leads to a short lifetime. The computational complexity order of the GT algorithm is $O(2(N-1))$ which is much less than the exhaustive search. The convex algorithm finds answers to the problem with the computational complexity order of $O(N.M)$ [14]. Simulations showed that the GT algorithm finds more efficient answers for the problem with lower computational complexity order than the convex algorithm.

6 Conclusion

In this paper, to lifetime extending, a novel algorithm for cooperative node selection in a WSN was proposed, in which some sensors cooperatively sense spectrum activities in multiple channels while others go into a sleep mode to save energy. This algorithm can be used for applications that the recharge of sensors is difficult or impossible. Also, it was assumed that the sensors cannot sense more than one

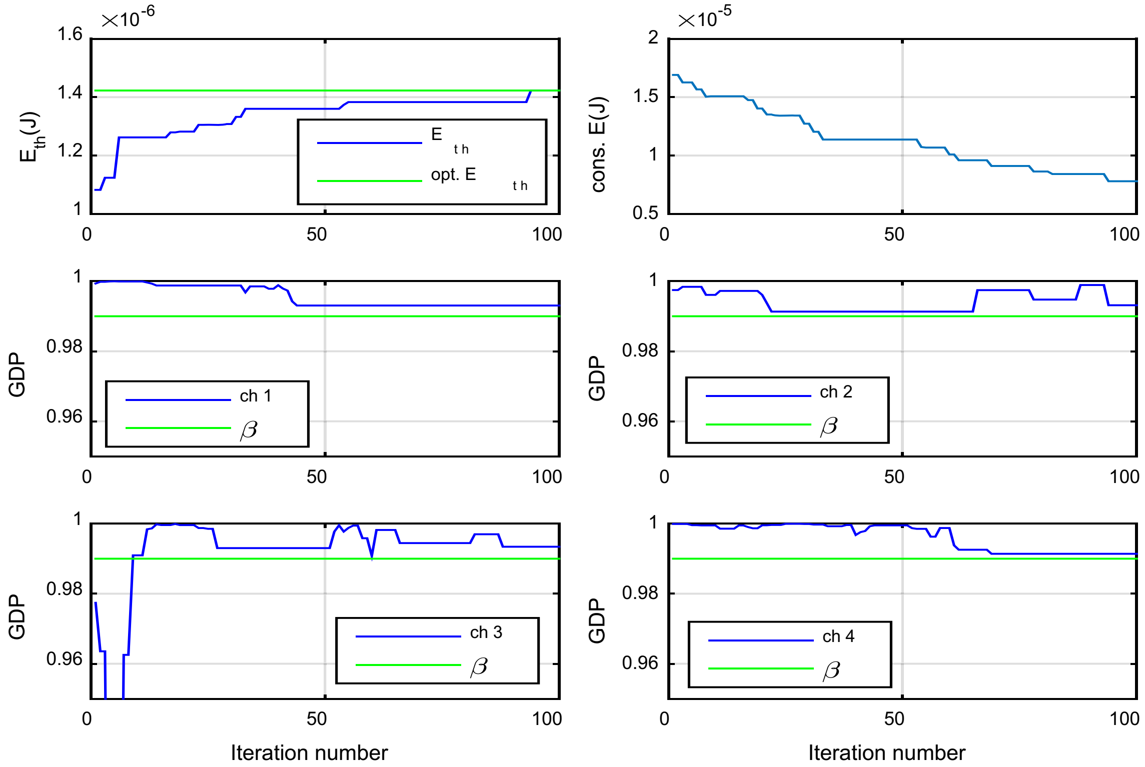

Fig. 12 Convergence of the proposed algorithm (in an iteration of network lifetime, N = 40, M = 4)

channel in a sensing duration. Based on the network duty and the sensor limitations, the lifetime maximization problem was constrained, to guarantee the necessary detection probability and low false alarm probability. The optimal solution for the problem is an exhaustive search with a high order of computational complexity, in each sensing period. In this paper, for solving the problem a CFG was proposed in which sensors act as game players and decide to make coalitions to sense the channels or not while guaranteeing the detection quality constraints. The selected sensors are activated to sense the channels and others put into a sleep mode to save energy. A novel efficient utility function was designed to make sure that the game is converged to astable NE. The properties of the used game and its convergence proof were discussed. Then, an efficient algorithm with a computational complexity order of $O(2(N - 1))$, to reach a proper Nash-stable coalition structure was concluded. The proposed algorithm was compared with other comparable algorithms of lifetime extending in such a network. Through simulations, it was demonstrated that the proposed algorithm extends the network lifetime more, e.g. up to four times of the random algorithm and up to 1.6 times of the detection based algorithm in a scenario assuming four channels. Also, the lifetime of the GT algorithm is *60%* more than the lifetime of the convex algorithm, in a network with a low number of sensors. These results are compared when the success rate of the GT algorithm is more than the success rate of the convex algorithm (up to 30% for $N = 40$, $M = 8$), which means that the proposed algorithm extends the network lifetime while it provides sufficient detection quality.

Appendix 1

We want to define a suitable value function for coalitions in the proposed game such that it conducts the game to a proper NE, which maximizes the E_{th}, while it must capture the tradeoff between the GDP, the GFP, the E_{th} and energy consumption. Based on the [31], for a finite OPG, the rank of a strategy is a proper candidate for the value of the strategy. This rank of a strategy is measured by counting the number of other strategies that have lower benefits than the strategy.

Now we use this idea to define a coalition utility for the proposed game. We define *zero* value for the coalition number of *zero*, i.e. the coalition of sensors which sense none of the channels. The result of selecting this *zero* value is that if sensors spend no cost, they receive no gain. Also, if a coalition does not satisfy at least one of the constraints ($P_{d_m}(J_m) < \beta$ or $P_{f_m}(J_m) > \alpha$ or both), its value is defined negatively because forming the coalition causes to consume energy but it cannot provide adequate detection quality. If a coalition satisfies all the constraints

($P_{d_m}(J_m) \geq \beta$ and $P_{f_m}(J_m) \leq \alpha$), its value is defined positively.

Based on the lifetime maximization goal of the problem, the value of a coalition must be an increasing function of E_{th} and a decreasing function of the energy consumption of sensors in the coalition. Between coalitions with positive values, the lower $E(J_m)$ and higher E_{th}, the more utility. So when a player wants to make a decision, it first determines the effect of its decision on the E_{th}. If its decision increases the E_{th} in respect to the previous round of the game (the minimum of remaining energy of sensors after the previous sensor has played is denoted by $E_{th,0}$), selecting the decision causes more utility. If its decision does not change the $\mathrm{E_{th}}$ level of sensors, the utility of selecting the decision only depends on the $E(J_m)$; hence, the decision with the lower $E(J_m)$ has more utility. Of course, these decisions have a lower utility than the decision which increases the E_{th}. Therefore, we have defined the utility of the coalition when the sensor decision provides $E_{th} > E_{th,0}$ as $1 + E_{th}$, which has a measure greater than one. Then, the utility of the coalition when the sensor decision provides $E_{th} = E_{th,0}$ as $\frac{max(E)-E(J_m)}{max(E)-min(E)}$, which is:

$$\frac{min(E)}{max(E)-min(E)} < \frac{max(E)-E(J_m)}{max(E)-min(E)} < 1 \tag{16}$$

The strategies that a sensor decision reduces the E_{th} and causes to $E_{th} < E_{th,0}$ has the lower utility, hence we define its value as $\frac{0.5min(E)}{max(E)-min(E)} \frac{max(E)-E(J_m)}{max(E)-min(E)}$, because:

$$\begin{aligned} 0 &< \frac{0.5min(E)}{max(E)-min(E)} \frac{max(E)-E(J_m)}{max(E)-min(E)} \\ &< \frac{0.5min(E)}{max(E)-min(E)} < \frac{max(E)-E(J_m)}{max(E)-min(E)} < 1 \end{aligned} \tag{17}$$

A similar ordering of coalition values is done for the strategies that have a negative value. If a sensor decision increases the E_{th} (i.e. $E_{th} > E_{th,0}$), selecting the decision has more utility. Therefore, we have defined the utility of the coalition as $\frac{-min(E)}{max(E)} \frac{E_0-E_{th}}{E_0}$, which has a measure lower than *zero*, in which $\mathrm{E_0}$ denotes the initial energy of sensors. The upper and lower bounds of the utility value are:

$$-1 < \frac{-min(E)}{max(E)} < \frac{-min(E)}{max(E)} \frac{E_0 - E_{th}}{E_0} < 0 \tag{18}$$

Then, the utility when the sensor decision provides $E_{th} = E_{th,0}$ as $\frac{-E(J_m)}{max(E)}$, which is:

$$-1 < \frac{-E(J_m)}{max(E)} < \frac{-min(E)}{max(E)} \tag{19}$$

The strategies that a sensor decision reduces the E_{th} and causes to $E_{th} < E_{th,0}$ has the lower utility, hence we define its value as $-1 - \frac{E_0-E_{th}}{E_0} - E(J_m)$, because:

$$-3 < -1 - \frac{E_0 - E_{th}}{E_0} - \frac{E(J_m)}{max(E)} < -1 \tag{20}$$

Finally, the utility value of every coalition is defined as:

$$u(J_m) = \begin{cases} if\ P_{d_m}(J_m) \geq \beta,\ P_{f_m}(J_m) \leq \alpha & \begin{cases} 1 + E_{th}; & if\ E_{th} > E_{th,0} \\ \dfrac{max(E) - E(J_m)}{max(E) - min(E)}; & if\ E_{th} = E_{th,0} \\ \dfrac{0.5min(E)}{max(E) - min(E)} \dfrac{max(E) - E(J_m)}{max(E) - min(E)} & if\ E_{th} < E_{th,0} \end{cases} \\ if\ P_{d_m}(J_m) < \beta\ or\ P_{f_m}(J_m) > \alpha & \begin{cases} \dfrac{-min(E)}{max(E)} \dfrac{E_0 - E_{th}}{E_0}; & if\ E_{th} > E_{th,0} \\ \dfrac{-E(J_m)}{max(E)}; & if\ E_{th} = E_{th,0} \\ -1 - \dfrac{E_0 - E_{th}}{E_0} - E(J_m); & if\ E_{th} < E_{th,0} \end{cases} \end{cases} \tag{21}$$

Appendix 2

We define the following potential function for the proposed game:

$$\Phi(J_0, \ldots, J_M) = \sum_{l=1}^{M} u(J_l) \tag{22}$$

in which J_l is the current coalition of sensors that cooperatively sense channel l. Given any coalition J_m, an improvement step of player n is a change of its strategy from $a_n = m$ to $a_n = m'$, such that the reward utility of player n increases. This move is performed in two possible ways. The first way is adding to the coalition m' when other nodes in the coalition remain. In this ways, the player n does the move when he receives more reward, i.e.:

$$\begin{aligned} &Re_n(J_{m'} + \{n\}) > Re_n(J_m + \{n\}) \\ &\quad \rightarrow u(J_{m'} + \{n\}) - u(J_{m'}) > u(J_m + \{n\}) - u(J_m) \end{aligned} \tag{23}$$

Now the potential function of the game before and after the move of player n are called as $\varPhi_1$ and $\varPhi_2$, respectively. The measures of potential function are calculated as:

$$\varPhi_1 = \sum_{l=1}^{M} u(J_l) = u(\mathrm{J}_{m'}) + u(J_m + \{\mathrm{n}\}) + \sum_{\substack{l=1 \\ l \neq m, m'}}^{M} u(J_l) \tag{24}$$

$$\varPhi_2 = \sum_{l=1}^{M} u(J_l) = u(\mathrm{J}_{m'} + \{\mathrm{n}\}) + u(J_m) + \sum_{\substack{l=1 \\ l \neq m, m'}}^{M} u(J_l) \tag{25}$$

Therefore, the change in the potential function of the game is calculated as:

$$\begin{aligned} \Delta\varPhi &= \varPhi_2 - \varPhi_1 \\ &= u(\mathrm{J}_{m'} + \{\mathrm{n}\}) + u(J_m) - u(\mathrm{J}_{m'}) - u(J_m + \{\mathrm{n}\}) > 0 \end{aligned} \tag{26}$$

The second way for the player n is adding to the coalition m' when a node n_0 in the coalition removes. This move is done if:

$$\begin{aligned} &Re_n(\mathrm{J}_{m'} + \{n\} - \{n_0\}) \geq Re_n(J_m + \{\mathrm{n}\}) \\ &\quad \rightarrow u(J_{m'} + \{n\} - \{n_0\}) - u(J_{m'}) > u(J_m + \{n\}) - u(J_m) \end{aligned} \tag{27}$$

Now the measure of $\varPhi_2$ is calculated as:

$$\begin{aligned} \varPhi_2 &= \sum_{l=1}^{M} u(J_l) \\ &= u(J_{m'} + \{n\} - \{n_0\}) + u(J_m) + \sum_{\substack{l=1 \\ l \neq m, m'}}^{M} u(J_l) \end{aligned} \tag{28}$$

Therefore, the change in the potential function of the game is calculated as:

$$\begin{aligned} \Delta\varPhi &= \varPhi_2 - \varPhi_1 \\ &= u(J_{m'} + \{n\} - \{n_0\}) + u(J_m) - u(\mathrm{J}_{m'}) \\ &\quad - u(J_m + \{\mathrm{n}\}) > 0 \end{aligned} \tag{29}$$

Hence; an improvement step of an individual player increases also the potential function, in both possible ways. This is concluded that the proposed game is an OPG.

References

1. Ali, A., & Hamouda, W. (2016). Advances on spectrum sensing for cognitive radio networks: Theory and applications. *IEEE Communications Surveys & Tutorials, 19*(2), 1277–1304.
2. Kobo, H., Abu-Mahfouz, A., & Hancke, G. (2017). A survey on software-defined wireless sensor networks: Challenges and design requirements. *IEEE Access, 5,* 1872–1899.
3. Cichoń, K., Kliks, A., & Bogucka, H. (2016). Energy-efficient cooperative spectrum sensing: A survey. *IEEE Communications Surveys & Tutorials, 18*(3), 1861–1886.
4. Deng, R., Chen, J., Yuen, C., Cheng, P., & Sun, Y. (2012). Energy-efficient cooperative spectrum sensing by optimal scheduling in sensor-aided cognitive radio networks. *IEEE Transactions on Vehicular Technology, 61*(2), 716–726.
5. Maleki, S., Chepuri, S., & Leus, G. (2013). Optimization of hard fusion based spectrum sensing for energy-constrained cognitive radio networks. *Physical Communication, 9,* 193–198.
6. Vien, Q.-T., Nguyen, H. X., & Nallanathan, A. (2015). Cooperative spectrum sensing with secondary user selection for cognitive radio networks over Nakagami-m fading channels. *IET Communications, 10*(1), 91–97.
7. Najimi, M., Ebrahimzadeh, A., Andargoli, S. M. H., & Fallahi, A. (2014). Lifetime maximization in cognitive sensor networks based on the node selection. *IEEE sensors Journal, 14*(7), 2376–2383.
8. Hattab, G. & Ibnkahla, M. (2014). Multiband spectrum sensing: Challenges and limitations. In *Proc. WiSense workshop, Ottawa*.
9. Quan, Z., Cui, S., Sayed, A. H., & Poor, H. V. (2009). Optimal multiband joint detection for spectrum sensing in cognitive radio network. *IEEE Transactions on Signal Processing, 57*(3), 1128–1140.
10. Wu, Y., & Cardei, M. (2016). Multi-channel and cognitive radio approaches for wireless sensor networks. *Computer Communications, 94,* 30–45.
11. Zheng, M., Chen, L., Liang, W., Yu, H., & Wu, J. (2017). Energy-efficiency maximization for cooperative spectrum sensing in cognitive sensor networks. *IEEE Transactions on Green Communications and Networking, 1*(1), 29–39.
12. Ozger, M., Alagoz, F., & Akan, O. (2018). Clustering in multi-channel cognitive radio ad hoc and sensor networks. *IEEE Communications Magazine, 56*(4), 156–162.
13. Kaligineedi, P., & Bhargava, V. (2011). Sensor allocation and quantization schemes for multi-band cognitive radio cooperative sensing system. *IEEE Transaction on Wireless Communications, 10*(1), 284–293.
14. Bagheri, A., Ebrahimzadeh, A., & Najimi, M. (2017). Sensor selection for extending lifetime of multi-channel cooperative sensing in cognitive sensor networks. *Physical Communication*. https://doi.org/10.1016/j.phycom.2017.11.003.
15. Asheralieva, A., Quek, T., & Niyato, D. (2018). An asymmetric evolutionary bayesian coalition formation game for distributed resource sharing in a multi-cell device-to-device enabled cellular network. *IEEE Transactions on Wireless Communications, 17*(6), 3752–3767.
16. Song, L., Li, Y., Ding, Z., & Poor, H. (2017). Resource management in non-orthogonal multiple access networks for 5G and beyond. *IEEE Network, 31*(4), 8–14.

17. Kim, S. (2014). *Game theory applications in network design*. Hershey: IGI Global.
18. Dai, Z., Wang, Z., & Wong, V. W. S. (2016). An overlapping coalitional game for cooperative spectrum sensing and access in cognitive radio networks. *IEEE Transactions on Vehicular Technology, 65*(10), 8400–8413.
19. Umar, R., & Mesbah, W. (2016). Coordinated coalition formation in throughput-efficient cognitive radio networks. *Wireless Communications and Mobile Computing, 16,* 912–928.
20. Olawole, A., Takawira, F., & Oyerinde, O. (2019). Fusion rule and cluster head selection scheme in cooperative spectrum sensing. *IET Communications, 13*(6), 758–765.
21. Sasabe, M., Nishida, T., & Kasahara, S. (2019). Collaborative spectrum sensing mechanism based on user incentive in cognitive radio networks. *Computer Communications, 147,* 1–13.
22. Rajendran, M., & Duraisamy, M. (2019). Distributed coalition formation game for enhancing cooperative spectrum sensing in cognitive radio ad hoc networks. *IET Networks, 9*(1), 12–22.
23. Hao, X., Cheung, M., Wong, V., & Leung, V. (2011). A coalition formation game for energy-efficient cooperative spectrum sensing in cognitive radio networks with multiple channels. In *GLOBECOM*.
24. Belghiti, I., Berrada, I., & El Kamili, M. (2019). A scalable framework for green large cognitive radio networks. *Cognitive Computation and Systems, 1*(3), 79–84.
25. Moualeu, J. M., Ngatched, T. M. N., Hamouda, W., & Takawira, F. (2014). Energy-efficient cooperative spectrum sensing and transmission in multi-channel cognitive radio networks. In *IEEE international conference on communications (ICC)*, Sydney.
26. Arora, N., & Mahajan, R. (2014). Cooperative spectrum sensing using hard decision fusion scheme. *International Journal of Engineering Research and General Science, 2*(4), 36–43.
27. Noori, M., & Ardakani, M. (2011). Lifetime analysis of random event-driven clustered wireless sensor networks. *IEEE Transactions on Mobile Computing, 10*(10), 1448–1458.
28. Li, P., Gua, S., & Cheng, Z. (2014). Max-min lifetime optimization for cooperative communications in cognitive radio networks. *IEEE Transactions on Parallel and Distributed Systems, 25*(6), 1533–1542.
29. Shapely, L. S. (1988). A value for n-person games. In A. E. Roth (Ed.), *The shapely value* (pp. 31–40). Cambridge: University of Cambridge Press.
30. Xu, Y., Wang, J., Wu, Q., Anpalagan, A., & Yao, Y. (2012). Opportunistic spectrum access in unknown dynamic environment: a game-theoretic stochastic learning solution. *IEEE Transaction on Wireless Communications, 11*(4), 1380–1390.
31. Lã, Q. D., Chew, Y. H., & Soong, B.-H. (2016). *Potential game theory: applications in radio resource allocation*. Berlin: Springer.
32. Mardan, J., Arslan, G., & Shamma, J. S. (2009). Cooperative control and potential games. *IEEE Transactions on Systems, Man and Cybernetics, 39*(6), 1393–1407.
33. Monderer, D., & Shapely, L. S. (1996). Potential games. *Games and Economic Behavior, 14,* 124–143.
34. Han, D., & Lim, J. H. (2010). Smart home energy management system. *IEEE Transactions on Consumer Electronics, 56*(3), 1403–1410.
35. Ismail, N., & Othman, M. (2009). Low power phase locked loop frequency synthesizer for 2.4 GHz band Zigbee. *American Journal of Engineering and Applied Sciences, 2*(2), 337–343.
36. Flores, A., Guerra, R., Knightly, E., Ecclesine, P., & Pandey, S. (2013). IEEE 802.11 af: A standard for TV white space spectrum sharing. *IEEE Communications Magazine, 51*(10), 92–100.
37. Banerji S. (2013). Upcoming standards in wireless local area networks. *Wireless & Mobile Technologies.* arXiv preprint arXiv:1307.7633.
38. Chiaravalloti, S., Idzikowski, F., & Budzisz, Ł. (2011). *Power consumption of WLAN network elements*. Berlin: Technische Universität Berlin.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Asma Bagheri received the B.Sc. and the M.Sc. degree in electrical engineering from Ferdowsi University of Mashhad, Iran, in 2010 and 2013, respectively, and the Ph.D. degree in communication from Babol University of Technology, Mazandaran, Iran, in 2018. Her current research interests include wireless communications and spectrum sensing in cognitive networks.

Ataollah Ebrahimzadeh received the Ph.D. degree in electrical engineering in 2007. He is currently a Professor with the Faculty of Electrical and Computer Engineering, Babol University of Technology. He has authored or co-authored more than 80 papers in international journals and conferences. His current research interests include signal processing and artificial intelligence. Prof. E. zadeh is a reviewer of international conferences and journals.

Maryam Najimi received her B.Sc. in electronics from Sistan and Baloochestan University, Zahedan, Iran in 2004 and her M.Sc. in telecommunication systems engineering from K.N.Toosi University of Technology, Tehran, Iran, and Ph.D. degree in communication from Babol University of Technology, Mazandaran, Iran, in 2008 and 2014, respectively. She is currently an assistant professor with the department of electrical and computer engineering, University of Science and Technology, Behshahr, Iran. Her interests include Spectrum sensing in wireless cognitive sensor networks.