Optimum Sensor Selection Based on Energy Constraints in Cooperative Spectrum Sensing for Cognitive Radio Sensor Networks

Maryam Monemian*, Mehdi Mahdavi*, and Mohammad Javad Omidi*
*Electrical and Computer Engineering Department
Isfahan University of Technology, Isfahan, Iran
m.monemian@ec.iut.ac.ir, m.mahdavi@cc.iut.ac.ir, omidi@cc.iut.ac.ir

Abstract—Cooperative Spectrum Sensing (CSS) has been proposed to improve the reliability of decisions made about the presence of Primary Users (PUs) in Cognitive Radio Networks (CRNs). However, new solutions are required to face various challenges in the implementation of CSS. Efficiency in energy consumption for CSS is a challenging issue which should be solved through the effective management of sensors for CSS. In this paper, all the subsets of sensors which cooperatively satisfy the desired sensing accuracy are formulated as an optimization problem. Then, a heuristic algorithm is proposed to solve such an optimization problem. To reduce the computational complexity of the heuristic algorithm, a novel sub-optimal algorithm which chooses the subset with minimum average energy consumption for CSS is proposed. In addition, a sub-optimal algorithm is proposed to reduce the computational complexity of the heuristic algorithm. The simulation results confirm the advantages of the proposed algorithms in terms of energy-efficient sensor selection for CSS and low complexity compared to other state-of-the-art methods.

Index Terms—Energy consumption, Battery life, Sensor selection, Cognitive radio.

I. INTRODUCTION

WITH respect to the significant growth in requests for wireless services and the scarcity of the spectrum resources, Cognitive Radio Networks (CRNs) have been proposed where unlicensed Secondary Users (SUs) opportunistically use the bandwidth dedicated to some licensed Primary Users (PUs) [1]. In such networks, SUs may use the spectrum resources that belong to PUs, without interfering with their operation [1,2]. Cognitive Radio Sensor Networks (CRSNs), as a notable kind of CRNs, are wireless sensor networks where sensors are implemented with cognitive radio capability. In CRSNs the problem of resource scarcity is solved by opportunistic access to the available spectrum resources [3,4]. CRNs are classified into three categories based on the reactions that SUs show upon the identification of PU on the frequency spectrum [5]. In interweave networks, sensors completely vacate the frequency spectrum upon the presence of PU on the spectrum. Underlay and overlay networks are those where SUs and PUs can simultaneously transmit on the frequency spectrum [5]. In interweave networks, to monitor the presence of PUs on the desired frequency band, SUs should periodically sense the spectrum. In order to improve the reliability of spectrum sensing, Cooperative Spectrum Sensing (CSS) has been extensively used in the literature [6-43]. In CSS, several SUs independently sense the spectrum and jointly decide about the presence of PU on the frequency spectrum. The accuracy of CSS is determined based on two parameters which are detection and false alarm probabilities. The former represents the probability of correct identification of PU on the frequency spectrum when it is actually present. The latter is the probability of false detection of PU when it is actually absent. Although CSS provides significant advantages in terms of reliability, there are important challenges in its implementation. Each SU engaged in CSS should listen to spectrum and collect sufficient samples from PU’s signals. Then, it measures the energy of received samples and reports its sensing result to a Fusion Center (FC) to make final decision according to a special rule. Non-negligible energy consumption for performing CSS and the overhead made in FC are some of the challenges that should be properly managed.

In [6-11] the focus is on the maximization of CRN throughput while the sensing accuracy is satisfied. The optimum value of sensing duration is derived in [7]. The optimum sensing time and transmission powers are attained in [8]. In [9] sensing times, sensing thresholds and transmission powers are derived such that SUs’ throughputs are maximized. The order of channels to be sensed is obtained in [10] such that the average throughput of CRN is maximized. Spectral correlation is used as a tool to predict the presence of PU on the channels and reduce sensing time and overhead [11]. The SUs’ decisions are combined with different weights in [12]. The aim of such a study is to find the optimum weights such that the detection probability is maximized in a CRN with PU emulation attack. The resource allocation and throughput maximization in CRNs are discussed in [13]. In [14] the aim is to provide trade-off between sensing overhead and throughput in CRNs. The throughput enhancement for both primary and secondary networks is discussed in [15]. The optimal medium access probabilities are derived such that the SU’s throughput is maximized while the PU’s QoS requirements are provided. In [16] a robust CSS framework is proposed to mitigate the negative effects of malicious user presence in CRNs. However, none of the aforementioned references ([6-16]) has focused on the important subject of energy consumption of CSS.

In this paper, a CRSN is considered which includes sensors with different values of received SNR from PU. First, all the subsets of sensors that can satisfy the desired accuracy of CSS are formed based on the clustering phase of the proposed method. Then, the average energy consumption for CSS is computed for all the formed subsets. Then, the problem of sensor selection for CSS is formulated as an optimization problem by which the average energy consumption for CSS is minimized while the energy constraints of sensors are considered. Then, a heuristic algorithm is proposed to solve such an optimization problem. To reduce the computational complexity of the proposed heuristic algorithm, a novel sub-optimal algorithm with approximately same energy-efficiency performance is proposed. Such a sub-optimal algorithm leads to the reduced computational complexity.

The main contributions of the paper are explained as follows.

In this paper, a clustering method is proposed to form all the subsets of sensors which can cooperatively provide the network with desired detection and false alarm probabilities. In this method, the sensors with less detection probabilities will be grouped with the sensors with high detection probabilities. In fact, high detection probability is not a prioritization metric for participation in CSS. Therefore, by
choosing the appropriate group for CSS, it is possible to engage the sensors with less detection probabilities in CSS. As a consequence, the sensors with high detection probabilities do not encounter rapid battery drain.

- We formulate the sensor selection for CSS as an optimization problem. The aim of such an optimization problem is to minimize the average energy consumption for CSS while the energy constraints of the chosen sensors for CSS are carefully considered. Then, a heuristic algorithm called MECEC (Minimum Energy Consumption with Energy Constraints) is proposed to periodically select an appropriate set of sensors for CSS with respect to the energy constraints of its members.

- To reduce the computational complexity of MECEC algorithm, a sub-optimal algorithm is proposed which its performance is similar to MECEC algorithm from energy-efficiency point of view. However, the required computations of the sub-optimal algorithm is significantly lower than MECEC and other existing algorithms. Therefore, the improved performance of sub-optimal algorithm is not only limited to energy-efficient sensor selection but it also leads to reduced complexity.

- The unnecessary reports of sensors which do not affect the final decision of FC are removed. Therefore, energy consumption for reporting the sensing results and CSS process is reduced.

The structure of this paper is as follows. The related works are explained in section II. System model and main assumptions are described in section III. The problem formulation has been described in section IV. The novel heuristic algorithms for solving the optimization problems are presented in section V. Numerical results and comparisons are explained in section VI. Finally, Section VII consists of the concluding remarks.

II. RELATED WORKS

In order to reduce energy consumption for CSS, several research works have been proposed [23-36]. In [23] a censoring and sleeping method is proposed where some sleep during the sensing phase of CSS and some other sensors that sense the spectrum will censor their results and avoid transmitting them to FC. The sleeping and censoring rates are calculated such that the energy consumption for CSS is minimized. However, all sensors are considered with the same values of received SNR from PU which is not a practical assumption.

In [24] another energy efficient method for CSS is proposed. The reporting channels between sensors and FC are considered to be error-prone in this research. At first, an algorithm is proposed to determine the maximum acceptable error probability on the reporting channels. Then, an algorithm is proposed to dynamically select appropriate sensors for CSS considering their energy constraints. Such a method helps to keep more sensors live. In [25] a sensor selection method has been proposed for performing CSS in a CRSN. The sensors of CRSN receive approximately same SNRs from PU which is not a practical assumption.

In [26] another energy efficient method for CSS is proposed to minimize the energy consumption for CSS. The proposed method called Modified Energy Efficient Sensor Selection (MEESS) gives priority to sensors based on their detection probabilities and the energy amounts they consume for CSS. However, such a selection method leads to the unfair fast battery drain of the sensors with higher priorities and imperfect coverage of the network.

Network Lifetime Improvement Sensor Selection (NLISS) method has been proposed in [27]. Such a method periodically calculates a function for all sensors to dynamically determine their priorities for participation in CSS. The function considers the remaining energy of sensors and their detection probabilities. NLISS method keeps almost all sensors live until the last time frame of network lifetime. However, the implementation of such a method requires a considerable volume of computations which should be periodically performed. A CSS method has been proposed in [28] which clusters sensors based on the values of received SNR from PU. Then, one of the sensors having the most reliable sensing data becomes the cluster head and sends its result to the FC. However, all sensors should listen to the spectrum for a contention time interval and only the one who wins the contention time is selected as a cluster head. Thus, all sensors periodically consume energy for sensing the spectrum. A CRN with multiple SUs and PUs is considered in [29]. Each channel which belongs to one PU should be sensed by at least a number of SUs at the beginning of each frame. One part of each frame is considered for spectrum sensing. During the spectrum sensing phase of each frame, each SU may sense one or more channels. The aim of this study is to determine the SU's which sense each channel such that the total energy consumed for CSS is minimized. However, no point about energy constraints of SUs has been considered in this study. The optimization of combination rule in a cognitive sensor networks with power-limited sensors is proposed in [30]. The authors use k-out-of-N rule to combine the hard decisions of sensors and find the optimum value of k maximizing the network throughput. Such an optimum value for k will also guarantee that the energy consumption of each sensor for CSS and data transmission does not exceed a maximum pre-defined value. However, all sensors sense the frequency spectrum in the sensing phase and the sensor selection for CSS which significantly affect the network lifetime [24-27] is not addressed in [30]. Furthermore, it is assumed that all sensors receive same SNRs from PU which is not a practical assumption. In [31] a censored truncated spectrum sensing method is proposed where sensors sense the spectrum during the truncation interval. To decide about the sensing results, two thresholds are considered. If the sensing results of sensors are more than the higher threshold or less than another one, they are sent to FC. Otherwise, they are censored. The aim is to find the optimum values for the thresholds such that the maximum value of energy consumption for CSS is minimized. In contrast with [31], we consider the current energy constraints of sensors as a metric for sensor selection for CSS, while no point about the sensor selection is addressed in [31].

Note that the important effect of sensor selection for CSS in the provisioning of energy efficiency has been previously proved in [24-27]. Two energy efficient CSS methods called Time Saving Energy Efficient One Bit-CSS (TSEE Cob CSS) are proposed in [32]. In the second method which is more efficient than the first one, all sensors sense the spectrum for a time interval which is a part of sensing phase. Also, two thresholds are considered for comparison with sensors’ results. If the results are more than the higher threshold or less than the lower one, they are sent to FC. If the FC can make decision based on the received results, the sensing phase finishes. Otherwise, only the sensors which their results fall between the two thresholds sense the spectrum again. Note that it is be possible to provide the desired CSS accuracy by the participation of a number of sensors and not all of them. This makes our proposed method more efficient in term of energy consumption. A combined censoring and sleeping method for energy-efficient CSS is proposed in [33]. Some sensors sleep during the sensing phase. The sensing results obtained by the awake sensors are compared with two thresholds and if they are located in the interval between thresholds, they are not sent to FC. The aim is to find the optimum values of sleeping rate and two thresholds such that the maximum average energy consumption per sensor for CSS is minimized. The optimum values are separately obtained for the cases where OR and AND rules are used in FC. However, no point about the engagement of sensors in CSS...
with respect to their dynamic energy constraints is considered in [33]. Joint optimization of required number of sensors for CSS and sensing scheduling has been proposed in [34]. By sensing scheduling, the authors mean that it is possible for FC to maintain one or more sensing phases to make the decision about the presence of PU more reliable. Also, the minimum number of sensors cooperatively satisfying the desired sensing accuracy is calculated. Moreover, the authors find an optimal sensing time such that they make a trade-off between sensing overhead and accuracy. In contrast with our proposed algorithms, the dynamic sensor selection for CSS regarding the battery-limited nature of sensors is not addressed in [34]. In [35] the optimal number of cooperating relays is derived to minimize the total energy consumption for cooperative sensing and transmission in CRNs while the desired sensing accuracy is satisfied. The relays help a secondary transmitter to sense the spectrum and then transmit data to a secondary destination. The spectrum is divided to several narrow bands. However, the responsible relays for CSS are considered with same missed detection probabilities which is not a practical assumption. In [36] the optimum sensing threshold and sensing nodes are determined such that energy consumption for CSS per frame is minimized. In order to determine the set of sensing nodes, a priority metric is calculated for each sensor. In fact, a sensor with more detection probability and less energy consumption for CSS receives higher priority for engagement in CSS. However, to periodically choose the sensors with high priority leads to the unfair fast battery drain of them.

### III. SYSTEM MODEL

In the following the system model is described. To facilitate the follow of this section and next sections, the descriptions of the most used parameters are presented in Table. 1.

#### A. Network layout

Consider a CRSN including $N$ sensors. Let us denote the $j$th sensor by $s_j$. The sensors sense some environmental parameters and transmit the information obtained to FC. Herein, we consider time-driven applications in Wireless Sensor Networks (WSNs) where sensors sense the environment with a certain period [44-45]. They opportunistically use the bandwidth dedicated to one Primary Base Station. The network model is shown in Fig. 1. The secondary network is considered to use interweave model for spectrum access [5]. Thus, to recognize the presence of PU on the frequency spectrum, the sensors periodically perform spectrum sensing. If the presence of PU is detected on the frequency spectrum, the sensors avoid data transmission. The sensors use energy detection method for spectrum sensing, since such a method does not require extra information about PU’s signal and is simple to implement.

To improve the accuracy of the spectrum sensing, CSS is used in which a number of sensors satisfying the desired detection and false alarm probabilities are chosen to sense the spectrum and determine the presence of PU on the frequency spectrum. Consider a time slotted channel in which time is divided in equal frames. Duration of each frame is equal to $T$ second (see Fig. 2). Each frame consists of five fields called Environmental Sensing, Sensor Selection, Spectrum Sensing, Reporting and Data Transmission. In the environmental sensing phase, sensors sense the pre-defined environmental parameter. The sensors can transmit the obtained data in data transmission phase only if the PU is detected to be absent. The sensor selection phase is the phase during which the proper sensors for CSS are chosen based on the proposed algorithm. The chosen sensors for CSS measure the energy of received signals from PU in the spectrum sensing phase. Based on the measurements obtained in the spectrum sensing phase, the sensors make hard decisions about the presence of PU. They can report their one-bit results to FC in the reporting phase. Let $M$ denote the maximum number of sensors that are engaged in CSS per frame and can satisfy the desired detection and false alarm probabilities. Moreover, let $T_r$ denote the required time for reporting one bit result of spectrum sensing to FC by a sensor. For the sake of simplicity we consider the case where the duration of reporting phase is equal to $T_r$. More details about the value of $M$ can be found in section II.

#### Table 1. The description of notations used in the paper.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_j$</td>
<td>The $j$th sensor in the network, $j=1,2,...,N$</td>
</tr>
<tr>
<td>$p_{d,j}$</td>
<td>Detection and false alarm probabilities of $s_j$</td>
</tr>
<tr>
<td>$p_{e,j}$</td>
<td>Error probability on the reporting channel between $s_j$ and FC</td>
</tr>
<tr>
<td>$p_{f,ij}, p_{f,ij}^{FC}$</td>
<td>Detection and false alarm probabilities of $s_i$ seen at FC</td>
</tr>
<tr>
<td>$P(n)$</td>
<td>Set of sensors engaged in CSS at the beginning of $n$th frame</td>
</tr>
<tr>
<td>$P_d(n)$</td>
<td>Detection probability obtained through cooperation between sensors in the $n$th frame</td>
</tr>
<tr>
<td>$P_f(n)$</td>
<td>False alarm probability obtained through cooperation between sensors in the $n$th frame</td>
</tr>
<tr>
<td>$p_{(a)s,i}$</td>
<td>Probability of sensing the environment by $s_i$ at the beginning of $n$th frame</td>
</tr>
<tr>
<td>$p_{(p)s,i}$</td>
<td>PU identification probability on the frequency spectrum at the beginning of $n$th frame</td>
</tr>
<tr>
<td>$p_{s,i}$</td>
<td>The probability of producing the result of “1” by $s_i$ in the sensing phase</td>
</tr>
<tr>
<td>$p_{(i)f,s}$</td>
<td>The probability of participation in CSS by $s_j$ at the beginning of $n$th frame</td>
</tr>
<tr>
<td>$f_k$</td>
<td>A set of $k$ sensors satisfying the desired detection and false alarm probabilities</td>
</tr>
<tr>
<td>$C^k$</td>
<td>The set of all $i_k (k=1,2,...)$</td>
</tr>
<tr>
<td>$g_i$</td>
<td>The $i$th member of $C^k$</td>
</tr>
<tr>
<td>$\Delta_{i,f}, \Delta_{a,s}, \Delta_{f,s}$</td>
<td>The amount of energy consumed for the sensing phase, environmental sensing, reporting the result of sensing by $s_i$, data transmission by $s_j$ to FC.</td>
</tr>
<tr>
<td>$E_{i,f}$</td>
<td>The remaining energy level of $s_i$ at the beginning of $n$th frame</td>
</tr>
<tr>
<td>$e_{i}^k$</td>
<td>The average energy consumption for CSS by the members of $g_i$</td>
</tr>
<tr>
<td>$E_{i,min}$</td>
<td>The minimum energy value of $s_i$ such that the sensor can be considered alive</td>
</tr>
<tr>
<td>$E_{\text{avg}}(n)$</td>
<td>The average energy consumption for CSS at the beginning of the $n$th frame</td>
</tr>
<tr>
<td>$\lambda_{th}$</td>
<td>Energy threshold value</td>
</tr>
<tr>
<td>$S(n)$</td>
<td>A subset of $k$-sets that the energy levels of their members are more than $\lambda_{th}$ at the beginning of $n$th frame</td>
</tr>
<tr>
<td>$T(n)$</td>
<td>A subset of $k$-sets that their members are alive at the beginning of $n$th frame</td>
</tr>
<tr>
<td>$S(n)$</td>
<td>The set of all candidate sets whose members energy levels are more than $\lambda_{th}$ at the beginning of $n$th frame</td>
</tr>
<tr>
<td>$T(n)$</td>
<td>The set of all candidate sets that their members are alive at the beginning of $n$th frame</td>
</tr>
</tbody>
</table>

![Fig. 1. The structure of system model.](image-url)
consider the energy constraints of sensors in the sensor selection for CSS as will be explained in section IV.B.

Another energy threshold called minimum energy value is defined for each sensor. The minimum energy value of \(s_j\), denoted by \(E_{j,\text{min}}\), is the amount of energy such that the sensor can be considered alive at a given frame if its current energy level is more than or equal to \(E_{j,\text{min}}\). It is apparent that \(\lambda_{j,h} \geq E_{j,\text{min}}\). For the proposed algorithm, it is sufficient for FC to know whether or not the energy level of a sensor (e.g. the \(j^{th}\) sensor, \(s_j\)) is more than \(\lambda_{j,h}\) and/or \(E_{j,\text{min}}\). In order to do so, FC has a lookup table in which for each \(s_j\) two bits are considered. The value of each corresponding bit for \(s_j\) is “0” in FC. If the energy level of a sensor (e.g. \(s_j\)) reaches to \(\lambda_{j,h}\), such sensor sets one bit to “1” and sends it to FC. Then, FC sets the first bit corresponds to \(s_j\) to “1”. After a while when the energy level of \(s_j\) becomes less than \(E_{j,\text{min}}\), it sends again the aforementioned bit (i.e. “1”) to FC. To facilitate the implementation of such a method we use a control channel with a data rate of 250kbit/s similar to [23,30]. The sensors send their identification number together with one bit information of their remaining energy values over the control channel during the data transmission phase. Since a sensor sends such information only two times during the network lifetime, negligible amount of energy is consumed. Furthermore, no heavy overhead is made at FC.

B. CSS formulation

At the beginning of each frame, the sensors engaged in CSS collect sufficient samples during the sensing phase. In the reporting phase, each sensor that participates in CSS is allowed to transmit its result about sensing the presence of PU by one bit. The value of such a bit is “1” for the presence of PU and “0” for the absence of PU. The data transmission phase is the interval used by sensors, only if the presence of PU has not been identified during the sensing phase. Let \(p_{dj}(j = 1, \ldots, N)\) and \(p_{fj}\) denote the detection probability of \(s_j\) and the false alarm probability of each sensor, respectively. We note that the duration of sensing phase is the same for all sensors and we assume that the sensing threshold is the same for all sensors. Therefore, the false alarm probabilities of all sensors are equal [6,46]. Also, energy detection technique is used for spectrum sensing due to its simple implementation [47]. Assume that the reporting channels between sensors and FC are error-prone. Therefore, the detection and false alarm probabilities of sensors at the location of FC differ from their original values i.e. \(p_{dj}(j = 1, \ldots, N)\) and \(p_{fj}\). Let \(p_{ej}\), \(p_{dj}^{FC}\) and \(p_{fj}^{FC}\) denote the error probability on the reporting channel between \(s_j\) and FC, the detection and false alarm probabilities of \(s_j\) seen at FC, respectively. Having known \(p_{ej}\), \(p_{dj}\) and \(p_{fj}\), \(p_{dj}^{FC}\) and \(p_{fj}^{FC}\) can be obtained as follows.

\[
p_{dj}^{FC} = p_{dj}(1 - p_{ej}) + p_{ej}(1 - p_{dj}) \tag{1}
\]

\[
p_{fj}^{FC} = p_{fj}(1 - p_{ej}) + p_{ej}(1 - p_{fj}) \tag{2}
\]

Similar to most previous studies [23], [26-27] and [29] OR rule is used in FC to combine the received results. Let \(p_{dj}^{a}\) and \(p_{fj}^{a}\) denote the detection and false alarm probabilities obtained after cooperation between sensors in the \(n^{th}\) frame. For OR rule, \(p_{dj}^{a}\) and \(p_{fj}^{a}\) are as follows.

\[
p_{dj}^{a} = 1 - \prod_{s_j \in P(n)} (1 - p_{dj}^{FC}) \tag{3}
\]

\[
p_{fj}^{a} = 1 - \prod_{s_j \in P(n)} (1 - p_{fj}^{FC}) \tag{4}
\]

Where in (3-4) \(P(n)\) denotes the set of sensors engaged in CSS at the beginning of \(n^{th}\) frame. To continue, we introduce \(\Delta_{j,f}, \Delta_{j,e}, \Delta_{j,r}, \Delta_{ij}\) and \(E_{dj}^{FC}, \Delta_{sf}\) represents the amount of energy consumption for collecting sufficient samples in the sensing phase by each sensor engaged in CSS. Note that the value of \(\Delta_{j,f}\) is the same for all sensors, since the duration of sensing phase is equal for all of them. \(\Delta_{j,e}\) indicates the energy amount consumed by each sensor for sensing the environmental parameter. \(\Delta_{j,r}\) indicates the amount of energy consumption to report one bit result by \(s_j\). Note that each sensor is located in a different distance from FC. Thus, to report the one bit sensing result to FC, each sensor adjusts its transmission power according to its distance to FC. As mentioned before the value of such a bit is “1” for the presence of PU and “0” for the absence of PU. When a sensor’s decision is “absence of PU” or “0”, its one bit report has no effect on the result of an OR operation. Thus, because of OR rule in FC, only those sensors whose decisions are “1” send their decisions to FC and the rest of sensors that are engaged in CSS, avoid transmission of their results. The amount of energy consumption of \(s_j\) for data transmission in the transmission phase is indicated by \(\Delta_{j,t}\). Let \(E_{j}\) denote the initial energy level of all sensors. The remaining energy level of \(s_j\) at the beginning of \(n^{th}\) frame is denoted by \(E_{j}^{t} \leq \prod_{j} (1 - \lambda_{j,h})\). Based on the above explanations, we can write.

\[
E_{j}^{t} = E_{j}^{t-1} - 1(\Delta_{j,e}) - 1(\Delta_{j,r}) - 1(\Delta_{j,f}) = E_{j} - \alpha_{j,h} - \beta_{j} - \gamma_{j} - \delta_{j} - \epsilon_{j} \leq a, b, c, d, e \leq n \tag{5}
\]

Where \(1(\Delta_{j,e}) = \Delta_{j,e}\) if \(s_j\) participates in sensing environment. Otherwise, \(1(\Delta_{j,e}) = 0\). Description of \(1(\Delta_{j,f}), 1(\Delta_{j,r})\) and \(1(\Delta_{j,t})\) are the same as \(1(\Delta_{j,e})\) and is not repeated. Note that \(a \leq b \leq c \leq d \leq e\). The proof is simple and therefore is omitted. Equation (5) reveals that \(E_{j}^{n}\) depends only on \(E_{j}^{t-1}\). This means that \(E_{j}^{n}\) is a Discrete Time Markov Chain (DTMC). To follow this section, we introduce \(p_{n,j}^{(se)}, p_{n,j}^{(pu)}\) and \(p_{n,j}^{(sp)}\). As mentioned earlier, we consider the time-driven applications where sensors sense the environment with a certain period denoted by \(r\) frames. The probability of sensing the environment by \(s_j\) at the beginning of \(n^{th}\) frame is denoted by \(p_{n,j}^{(se)}\). Note that \(s_j\) starts to sense environment from the frame the number of which is equal to the remnant of dividing \(j\) by \(r\). For the time-driven applications, we can write.

\[
p_{n,j}^{(se)}(n = 1, \ldots, r, k = 0, \ldots, n) = \begin{cases} 
1, & \text{if } n = kr + j \text{ mode } r, k = 0, \ldots, n, j = 1, \ldots, N \\
0, & \text{otherwise} \end{cases} \tag{6}
\]

\[
p_{n,j}^{(pu)}\) denotes the probability of identification of PU on the frequency spectrum at the beginning of \(n^{th}\) frame. We can write, \(p_{n,j}^{(pu)} = P(H_0)p_{n}^{(pu)} + P(H_1)p_{n}^{(pu)}\) \(n\) where in (7) \(P(H_0)\) and \(P(H_1)\) denote the probabilities of absence and presence of PU on the spectrum. Let \(p_{ij}\) denote the probability of producing the result of "1" by \(s_j\) in the sensing phase. In fact, we have, \(p_{ij} = P(H_0)p_{ij}^{(se)} + P(H_1)p_{ij}^{(se)}\)

The probability of participation in CSS by \(s_j\) at the beginning of \(n^{th}\) frame is denoted by \(p_{n,j}^{(sp)}\). The value of \(p_{n,j}^{(sp)}\) depends on the type of algorithm used for sensor selection for CSS and is
computed in section IV.D. Finally, using equations (5)–(8) we can easily find $P(E_{n}^{1}E_{n-1}^{1})n \geq 1$ as follows.

$$P(E_{n}^{1} = E_{n-1}^{1}) = a_{n-1} - b_{n-1} - c_{n-1} + b_{n-1} + c_{n-1} - d_{n-1}$$

$$= (\frac{p_n}{1-p_n})^{d_{1}}(\frac{c_n}{1-c_n})^{(a-1)}(\frac{p_n}{1-p_n})^{(c-d)}$$

$$= (\frac{1-p_n}{p_n})^{d_{1}}(\frac{1-p_n}{1-p_n})^{(c-d)}$$

With respect to this point that $P(E_{n}^{1} = E_{n-1}^{1}) = 1$ for all $s_i (i=1,...,N)$, we can find $P(E_{n}^{1})$ recursively using $P(E_{n}^{1}E_{n-1}^{1})$ as follows.

$$P(E_{n}^{1}) = \sum_{s_n} P(E_{n}^{1}E_{n-1}^{1})P(E_{n-1}^{1})$$

(10)

Note that $P(E_{n}^{1})$ helps us to obtain the network life time described in section V.B.

IV. PROBLEM FORMULATION

In this section, the problem of selecting possible appropriate sensors for CSS, is formulated.

A. Clustering of Sensors

In this section, the aim is to find different sets of sensors satisfying the desired detection and false alarm probabilities. The set that contains all the sensors is denoted by $S_N$. Let $f_k \subseteq S_N$ denote a possible set of $k$ sensors which can satisfy the desired detection and false alarm probabilities that is

$$f_k = \begin{cases} 1 - \prod_{i=1}^{j-1} (1 - p_{j-1}^{f_{j-1}}) \geq \delta_1 \\ 1 - \prod_{i=1}^{j-1} (1 - p_{j-1}^{f_{j-1}}) \leq \delta_2 \\ 1 \leq l \leq k, s_i \in S_N \\ \end{cases}$$

(11)

Where in (11) $\delta_1$ and $\delta_2$ denote the desired detection and false alarm probabilities, respectively. To facilitate the analysis, hereafter we call $f_k$ as $k$-set. With respect to the definition of $M$ in section IIIA, one can see that $M$ is the maximum value that $k$ can take. Such a value is also equal to the maximum number of sensors that can be engaged in CSS per frame. Thus, the range of $k$ in $k$-sets is $1 \leq k \leq M$. Given $k$, we define $C_k$ as the set of all possible $k$-sets that is

$$C_k = \{ f_k, 1 \leq k \leq M \}$$

(12)

In the following, we explain the steps we have used to determine all $f_k \in C_k (1 \leq k \leq M)$ defined in (11) without performing exhaustive search. Note that in constructing $f_k$, permutation is not considered. The steps are based on equation (11).

- Sorting: We sort the detection probabilities of sensors in a descending order i.e. $p_{j-1}^{f_{j-1}} < p_{j-1}^{f_{j-2}}$ for $1, .., N - 1$.

- Form $C_k$. To form $C_k$, the detection probabilities of the sensors are checked in a descending order. While the detection probabilities of the sensors are more than or equal to $\delta_1$, each of them is considered as 1-set i.e. $f_1 \subseteq C_k$. However, when detection probability of a sensor becomes less than $\delta_1$, it is not necessary to check the rest of sensors with less detection probabilities.

- Form $C_k$. Here we have two phases. In the first phase, the detection probabilities of the sensors are checked in descending order. While $p_{j-1}^{f_{j-1}} \geq \delta_1, 1 \leq j \leq N$, such a sensor (i.e. $s_i$) with each of the rest of sensors is considered as 2-set (i.e. $f_2$) regardless of their detection probabilities. This is because inequality (13) is true that is

$$(1 - p_{j-1}^{f_{j-1}}) \geq \delta_1, \forall j \neq 1$$

(13)

We note that $p_{j-1}^{f_{j-2}} \geq \delta_1$ means $(1 - p_{j-1}^{f_{j-1}}) \leq 1 - \delta_1$. Since for all $j = 2, 3,.., N$ we have $1 - p_{j-1}^{f_{j-2}} \leq 1$, then it can be readily seen that the above inequality is true. The second phase starts when we reach to the first sensor say $s_j$ whose detection probability is less than $\delta_1$, we check each sensor $s_{j_i}, j = 1, ..., N$. While $(1 - p_{j-1}^{f_{j-1}}) \geq (1 - \delta_1), s_i$ and $s_j$ are considered as 2-set (i.e. $f_2$). In this way, when we reach to the first sensor $s_j$ which cannot make a 2-set with $s_j$, it is not necessary to check the rest of the sensors i.e. $s_{r(j}>j>)}$. This is because the detection probability of $s_{r(j)>j)}$ is less than $s_j$. Thus, it is possible to form all $f_2 \in C^2$ without checking all (2)-combinations of sensors.

- Form $C_k^2 (2 \leq k \leq M)$. First, we note that if $C^3$ is not empty, all $f_3 \subseteq C^3$ can make $k$-set i.e. $f_k$ with all ($k$)-combinations of sensors. Also, if $C^3$ is not empty all $f_2 \subseteq C^3$ can make $f_k$ with all ($k$)-combinations of sensors and so on. Hence, if $C^k=1$ is not empty, all $f_{k-1} \subseteq C^{k-1}$ can be grouped with all ($k$)-combinations of sensors. However, if $C^k=1$ is empty, we check inequality (14) for $k$ sensors with the highest detection probabilities.

$$\Pi_{j=1}^{k}(1 - p_{j-1}^{f_{j-1}}) \leq 1 - \delta_1$$

(14)

If the above inequality is true, then $s_1, s_2, ..., s_{k-1}$ and $s_k$ can make $f_k$.

Next, we start with $s_j, k+1 \leq j \leq N$. The detection probability obtained through cooperation between $s_1, s_2, ..., s_{k-1}$ and $s_j$ is computed. If such a probability is more than or equal to $\delta_1$, such sensors can also make $f_k$. This trend continues until all $C^k$ are formed. However, inequality (14) is not true for $k$ sensors with the highest detection probabilities, there is no need to check the inequality for other ($k$)-combinations of sensors. The reason is that if the first selected sensors with highest detection probabilities cannot establish a $f_k$, the other sensors with lower detection probabilities are disable to establish such a group. Thus, since it is not necessary to check all possible ($k$)-combinations of sensors, no exhaustive search is performed.

Each member of $C^k$ can be considered as a candidate group for CSS. Without loss of the generality the members of $C^k$ (i.e. the $k$-sets for CSS) are numbered from 1 to $|C^k|$ where $|C^k|$ is the number of $k$-sets in $C^k$. For the sake of the simplicity, we denote the $i^{th}$ member of $C^k (i=1, ..., |C^k|)$ of member of $C^k (k=1, ..., M)$ by $g^k_i$ that is

$$g^k_i = \{s_{i_1}, s_{i_2}, ..., s_{i_k}\} \subseteq S_N, i = 1, ..., k$$

Let $e^k_i$ denote the average energy consumption for CSS when the members of $g^k_i$ are chosen for CSS. We can write,

$$e^k_i = kDA_f + \sum_{s_j \in g^k_i} P_{s_j} \lambda_{s_j}$$

(15)

Where $kDA_f$ indicates the total sensing energy consumption of all $k$ members of $g^k_i$ since they participate in CSS. Also, the second term in (15) indicates the average reporting energy consumption for the members of $g^k_i$.

B. Problem Definition

Our aim is to minimize the average energy consumption for CSS in each frame. Therefore, the optimization problem is as follows.

$$\min_{g^k_i} |E_{ave}(n)|$$

s.t. \(1 - \prod_{j\in g^k_i}(1 - p_{j-1}^{f_{j-1}}) \geq \delta_1 \)

$$1 - \prod_{j\in g^k_i}(1 - p_{j-1}^{f_{j-1}}) \leq \delta_2$$

$$E_{j}^{k} \geq \lambda_{w}, \forall s_j \in g^k_i$$

(16-1)

(16-2)

(16-3)

Where as explained in section IIIA $\lambda_{w}$ is an energy threshold which is used for sensor selection in CSS. Using $\lambda_{w}$ it is possible to categorize the $k$-sets formed by (12) based on the remaining energy levels of sensors. Also, based on the definition of $C^k$, we can be sure that constraints (16-1) and (16-2) are satisfied for all $g^k_i \in C^k, i = 1, ..., |C^k|, k = 1, ..., M$ (see equations (11-12)).
Thus, P1 can be converted to the following optimization problem.

\[
P_2: \min_{g_l^k} |E_{\text{ave}}(n)|
\]  
\[
\text{s.t. } E_n^l \geq \lambda_{th}, \forall s_j \in g_l^k
\]

(17-1)

Based on the optimization problem defined in (17), our aim is to find an appropriate candidate \(k\)-set, \(g_l^k\) for CSS in each frame which can minimize the average energy consumption for CSS. It is possible that the constraint (17-1) is not valid for any \(k\)-set, \(k = 1, \ldots, M\) after some frames, that is \(P2\) cannot be solved. In such a case it is reasonable to change the constraint of (17-1) to the constraint of

\[
E_n^l \geq E_{\text{thin}}^l, \forall s_j \in g_l^k
\]

This means that our aim is to minimize the average energy consumption for CSS during each frame based on the new constraint. Thus, the optimization problem \(P2\) is changed to the optimization problem \(P3\) as follows.

\[
P_3: \min_{g_l^k} |E_{\text{ave}}(n)|
\]

\[
\text{s.t. } E_n^l \geq E_{\text{thin}}^l, \forall s_j \in g_l^k
\]

(18-1)

Where in (18-1) \(E_{\text{thin}}^l\) denotes the minimum energy value of \(s_j\) such that the sensor can be considered alive at the beginning of \(n\)-th frame, if \(E_n^l \geq E_{\text{thin}}^l\).

V. HEURISTIC ALGORITHM

In this section, a heuristic algorithm is proposed to solve the optimization problems defined in the previous section. Using such a heuristic algorithm, \(P_{n_{x,j}}^{(x)}\) can be found. To describe the proposed heuristic algorithm, we introduce two subsets \(S(n)\) and \(T(n)\) of \(C^k\). We denote a subset of \(k\)-sets by \(S(n)\) where the energy levels of the members of such \(k\)-sets are more than \(\lambda_{th}\) at the beginning of \(n\)-th frame. In other words

\[
S(n) = \{g_l^k | \forall s_j \in g_l^k, E_n^l \geq \lambda_{th}, 1 \leq i \leq |C^k|\}
\]

(19)

In order to appropriately engage live sensors in CSS, we denote a subset of \(k\)-sets by \(T(n)\) where the members of such \(k\)-sets are alive at the beginning of \(n\)-th frame. In fact, we have,

\[
T(n) = \{g_l^k | \forall s_j \in g_l^k, E_n^l \geq E_{\text{thin}}^l, 1 \leq i \leq |C^k|\}
\]

(20)

To solve the optimization problems \(P2\) and \(P3\), one heuristic algorithm is proposed in the following.

A. MECEC Algorithm

In order to solve the optimization problems \(P2\) and \(P3\), MECEC algorithm is proposed which is called Minimum Energy Consumption with Energy Constraints (MECEC) algorithm. In order to explain MECEC algorithm, let \(S(n)\) denote the set of all candidate sets that the energy levels of their members are more than \(\lambda_{th}\) at the beginning of \(n\)-th frame. In other words, we can write,

\[
S(n) = \bigcup_{k=1}^{M} S^k(n)
\]

(21)

Also, let \(T(n)\) denote the set of all candidate sets that their members are alive at the beginning of \(n\)-th frame. In other words, we can write,

\[
T(n) = \bigcup_{l=1}^{M} T^l(n)
\]

(22)

First, the average energy consumption, \(e^{l}_k\), for all \(g_l^k \in C^k, k = 1, \ldots, M\) is computed via (15) (line 2). First, \(S(n)\) and \(T(n)\) are formed based on (21) and (22) (line 3). Then, all \(g_l^k \in S(n) (k = 1, \ldots, M)\) are sorted based on the values of \(e^{l}_k\) in an ascending order (line 4). Then, a loop is executed by which the appropriate sensors are chosen for CSS at the beginning of each frame (lines 5 to 23). If FC received information about the remaining energy levels of sensors in the previous frame, it updates \(S(n)\) and \(T(n)\) (line 7).

Algorithm 1: MECEC algorithm

1: \(n = 0\);
2: Calculate \(e^{l}_k\) for all \(g_l^k, i = 1, \ldots, |C^k|, k = 1, \ldots, M\) using (15).
3: Form \(S(n)\) and \(T(n)\) based on (21) and (22).
4: Sort all \(g_l^k \in T(n) (k = 1, \ldots, M)\) based on \(e^{l}_k\) in an ascending order.
5: WHILE (TRUE)
6: \(P(n) = \emptyset\);
7: Update \(S(n)\) and \(T(n)\) (if necessary).
8: IF (|\(S(n)\)| \(\geq 1\))
9: Choose \(g_l^k \in S(n)\) with the least \(e^{l}_k\) from \(S(n)\) as \(P(n)\);
10: END IF
11: IF (\(P(n) = \emptyset\))
12: \(P(n) = \emptyset\);
13: Choose \(g_l^k \in T(n)\) with the least \(e^{l}_k\) from \(T(n)\) as \(P(n)\);
14: END IF
15: IF (\(P(n) \neq \emptyset\))
16: BREAK;
17: END IF
18: END WHILE
19: IF (\(P(n) \neq \emptyset\))
20: Update the energy levels of the members of \(P(n)\);
21: END IF
22: \(n = n + 1\);
23: END WHILE

If \(S(n)\) includes at least one candidate set, the candidate set \(g_l^k \in S(n)\) with the least value of \(e^{l}_k\) is chosen as the set of appropriate sensors for CSS (lines 8 to 10). However, if \(S(n)\) does not include any candidate set, the optimization problem \(P2\) cannot be solved and \(P(n)\) remains empty. In such a situation, we look for the solution for \(P3\). The proper sensors for CSS are selected from \(T(n)\) in this case. If \(T(n)\) includes at least one candidate set, the set \(g_l^k \in T(n)\) with the least value of \(e^{l}_k\) is selected as the proper set of sensors for CSS (lines 12 to 14). If \(T(n)\) does not include any candidate set, it is not possible to choose sufficient sensors for CSS and \(P(n)\) remains empty. In such a situation, the algorithm stops (lines 15 to 17). If it is possible to choose sufficient sensors for CSS, their energy levels are updated according to their energy consumptions for CSS (lines 19 to 21).

B. Determination of \(P_{x,j}^{(x)}\)

Herein, the probability of participation in CSS by \(s_j\) at the beginning of \(n\)-th frame is computed. Found having such a probability, \(P(E^l_{n_{x-1}}|E^l_{n-1})\) can be found (see section III.B)

Let \(X^l\) denote the set of all candidate groups such that \(s_j\) belongs to them. In other words,

\[
X^l = \{g_l^k \in C^k | s_j \in g_l^k, 1 \leq j \leq N\}
\]

(23)

We can write,

\[
p_{x,j}^{(x)} = \sum_{g_l^k \in X^l} p_n(g_l^k)
\]

(24)

Where in (24) \(p_n(g_l^k)\) denotes the probability of performing CSS by \(g_l^k\) at the beginning of the \(n\)-th frame. For MECEC algorithm, \(p_n(g_l^k)\) can be obtained as follows.

\[
|g_l^k \in S(n), g_l^k \in T(n)\|
\]

\[
p_n(g_l^k) = \begin{cases} 1, & \text{if } e^{l}_k = \min_{g_l^k \in S(n), g_l^k \in T(n)} e^{l}_k \\ 0, & \text{if } e^{l}_k \neq \min_{g_l^k \in S(n), g_l^k \in T(n)} e^{l}_k \end{cases}
\]

(25)

\[
|g_l^k \notin S(n), g_l^k \in T(n)\|
\]

\[
p_n(g_l^k) = \begin{cases} 0, & \text{if } |S(n)| \geq 1 \\ 1, & \text{if } |S(n)| = 0, e^{l}_k = \min_{g_l^k \in S(n), g_l^k \in T(n)} e^{l}_k \\ 0, & \text{if } |S(n)| = 0, e^{l}_k \neq \min_{g_l^k \in S(n), g_l^k \in T(n)} e^{l}_k \end{cases}
\]

(26)

C. Sub-optimal Algorithm

In order to execute MECEC algorithm, it is necessary to compute the values of \(e^{l}_k\) for all \(g_l^k, i = 1, \ldots, |C^k|, k = 1, \ldots, M\). Therefore, it is necessary to perform a considerable number of additions which increase the complexity of the MECEC algorithm. In order to reduce the complexity of MECEC
algorithm, a sub-optimal algorithm called Simplified MECEC (SMECEC) algorithm is proposed. As can be seen in the next section, SMECEC algorithm has the same performance from energy-efficiency point of view. However, the average number of required computations in SMECEC is significantly lower than other algorithms. The pseudo-code of SMECEC is presented in Table 1.

**Algorithm 2: SMECEC algorithm**

1. Form all clusters using the steps explained in section IV.A.
2. Compute \( p_k \Delta_{j} \) for all \( j = 1 \ldots N \).
3. Sort \( s_j \) \((j = 1 \ldots N)\) based on the values of \( p_k \Delta_{j} \) in an ascending order \((p_k \Delta_{j} \leq p_{k+1} \Delta_{j+1})\).
4. Let \( Pr = 0, k = 1, n = 0, a_0 = 0 \).
5. While \( (k \leq M) \)
   
   //for each \( k \) \( k \) nested loops are executed
   
   6. FOR \( a_{k-1} = 1 \) \( \rightarrow \) \( k \rightarrow 1 \)
   
   7. FOR \( a_{k-1} = 1 \) \( \rightarrow \) \( k \rightarrow 2 \)
   
   8. \( \rightarrow \) \( a_{k-1} = a_{k-1} + 1 \) \( \rightarrow \) \( N \rightarrow 1 \)
   
   9. \( Pr = Pr + 1 \).
   
   10. Assign \( Pr \) to \( \{a_0, \ldots, a_k\} \).
   
   11. END FOR
   
12. END FOR
13. END FOR
14. \( k = k + 1 \).
15. END WHILE
16. Correspond the obtained priorities to the clusters formed in line 1.
17. Sort \( S(n) \) and \( T(n) \) based on (21) and (22).
18. WHILE(TRUE)
19. Let \( P(n) = \emptyset \).
20. Update \( S(n) \) and \( T(n) \) (if necessary).
21. IF \((|S(n)| \geq 1)\)
22. Choose \( s_j \) \( \in S(n) \) with the least \( Pr \) from \( S(n) \) as \( P(n) \);
23. END IF
24. IF \((P(n) = \emptyset)\)
25. IF \((T(n) \geq 1)\)
26. Choose \( s_j \) \( \in T(n) \) with the least \( Pr \) from \( T(n) \) as \( P(n) \);
27. END IF
28. IF \((P(n) = \emptyset)\)
29. BREAK;
30. END IF
31. END IF
32. IF \((P(n) = \emptyset)\)
33. Update the energy levels of the members of \( P(n) \);
34. END IF
35. \( n = n + 1 \).
36. END WHILE

In SMECEC algorithm, we assign a priority for participation in CSS to all the \((k)\)-combinations of sensors \((1 \leq k \leq M)\). At the beginning of the algorithm, all clusters are formed using the steps explained in section IV.A. The prioritization of all groups is performed in lines 2 to 15. It should be noted that all \((k)\)-combinations of sensors and not only the clusters formed in line 1 receive priority in the prioritization process. To do so, the value of \( p_k \Delta_{j} \) is computed for \( s_j \) \( (j = 1 \ldots N) \) (line 2). Then, all sensors are sorted based on the values of \( p_k \Delta_{j} \) in an ascending order (line 3). Then, a loop is executed by which the proper priorities are assigned to all the \((k)\)-combinations of sensors \((1 \leq k \leq M)\) (lines 5 to 15). As can be seen in lines 6 to 13, the \((k)\)-combination of sensors which include the sensors with the least values of \( p_k \Delta_{j} \) receive the lowest values of \( Pr \). In other words, such a group of sensors has a higher priority for participation in CSS. Also, the \((k)\)-combination of sensors which include the sensors with the highest values of \( p_k \Delta_{j} \) receive the highest values of \( Pr \). In fact, such a group of sensors has a lower priority for participation in CSS. Since the sorting process has been performed in line 3, we can be sure that the first \((k)\)-combination of sensors made in the loop (lines 6 to 13) has the least average energy consumption for CSS among all \((k)\)-combinations of sensors. In line 16, the clusters of sensors which satisfy the desired detection accuracy, receive their priorities for participation in CSS. In line 17, \( S(n) \) and \( T(n) \) are formed using (21) and (22). Then, a loop is executed where the proper set of sensors is chosen for CSS at the start of each frame (lines 18 to 36). If \( S(n) \) is not empty, the candidate group with the least \( Pr \) which belongs to \( S(n) \) is chosen (lines 21 to 23). Otherwise, the candidate group with the least \( Pr \) which belongs to \( T(n) \) is selected for CSS (lines 24 to 27). Finally, if \( T(n) \) becomes empty, \( P(n) \) also becomes empty and it is not possible to perform CSS and the algorithm stops (lines 28 to 30).

VI. NUMERICAL RESULTS

A. Performance Metrics

In this section, the required performance metrics are introduced by which the performances of our proposed algorithms are evaluated.

Let \( F(\alpha) \) denote the maximum lifetime of network. As mentioned in previous literature [48-49], there are different definitions for lifetime. Such definitions can be addressed as in the following equation:

\[
F(\alpha) = \arg \max_{n} \{|N(n)| \geq \alpha N\}
\]

Where in (27) \( \alpha \) is a multiplier between 0 and 1, \( N(n) = \{s_j | E^j_n \geq E_{min}\}, j = 1 \ldots N \) denotes the set of live sensors at the beginning of \( n \)th frame and \( |N(n)| \) denotes the number of members of \( N(n) \). Using equation (10) we can find \( N(n) \). Given \( \alpha, F(\alpha) \) indicates the time where by the end of this time, \( F(\alpha) \) percent of the nodes are still alive. Let \( E_{rep} \) denote the total energy amount consumed for reporting the sensing results to FC.

\[
E_{rep} = \sum_{n=0}^{\infty} N(n) \Delta_{j} + E_{rep}
\]

Where \( R(n) \) \((R(n) \subseteq P(n))\) denote a subset of cooperating sensors at the beginning of \( n \)th frame which report their results to FC. Also, \( F(\lambda) \) is the maximum frame number where by the start of this time, it is still possible to perform CSS. Let \( E \) denote the total energy amount consumed for CSS during the network lifetime. We can write,

\[
E = \sum_{n=0}^{\infty} L(\lambda) \Delta_{j} + E_{rep}
\]

Let \( E_{ave} \) denote the average energy amount consumed for CSS per frame during the network lifetime. In fact, we have,

\[
E_{ave} = \frac{E}{F}
\]

B. Requirements of the Performance Evaluation

In this section, the state-of-the-art research works used for comparison and also the required inputs for performance evaluations of the proposed algorithms are explained. In order to evaluate the performance of proposed algorithms, a Chipcon CC2420 transceiver based on IEEE 802.15.4/Zigbee is considered [50]. The sensors are uniformly distributed in a circular field with a radius of 100 m. FC is located at the center of the field. PU and FC have fixed locations in the CRSN. The values of parameters used in the simulations are presented in Table 2. The curves obtained in this section are the average of 100 different topologies. The simulations have been obtained via MATLAB R2011a.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>( \Delta_{j} )</th>
<th>( \Delta_{\alpha} )</th>
<th>( \Delta_{p} )</th>
<th>( \Delta_{s} )</th>
<th>( E )</th>
<th>( \mu_{d} )</th>
<th>( \delta_{1} )</th>
<th>( \delta_{2} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>0.2</td>
<td>0.1</td>
<td>0.07</td>
<td>0.48</td>
<td>250 [\mu]</td>
<td>0.42</td>
<td>0.08</td>
<td>0.9</td>
</tr>
</tbody>
</table>

The research works chosen for comparison with the proposed algorithm are as follows.

1-Modified Energy Efficient Sensor Selection (MEESS) [26].

In MEESS, the sensors with higher detection probabilities and less energy consumption for CSS receive the higher priorities for participation in CSS. The priority function for \( s_j \), \( cost(j) \), is presented as follows [26].
cost(j) = Δs_f + Δs_r - λpd^{RC}

(31)

Where in (31) \( \lambda \) is a multiplier weighting the effect of \( p d^{RC} \). To the best of our knowledge, this is one of the most recent works focusing on the minimization of energy consumption for CSS.

2-Network Lifetime Improvement Sensor Selection (NLISS) [27],

In NLISS, a function is periodically computed to dynamically determine the priorities of sensors for CSS. Such a function prioritizes sensors based on their remaining energy values and their detection probabilities. The priority function for \( s_j \), \( pri - func(j) \), is presented as follows [27].

\[
pri - func(j) = 0.5 \times \left( E_{j-1} - (\Delta s_f + e_{amp}d_j^2) \right) + \frac{\lambda}{2\pi}p d^{RC} - \frac{\eta_j}{2\pi}
\]

(32)

Where in (32) \( E_j \) and \( d_j \) denote the remaining energy value and the distance of \( s_j \) from FC, respectively. Also, \( e_{amp} \) is the required amplification to satisfy receiver sensitivity at FC. \( \lambda, \eta_j \), and \( \epsilon_j \) are the multipliers that should also be updated during each frame. NLISS is one of the most relevant works in the field of network lifetime improvement.

C. Performance Evaluation of the Proposed Algorithms

In this section the performances of the proposed MECEC and SMECEC along with MEESS and NLISS algorithms are presented (as can be seen in figures 4 to 8.) We also summarized the benefits of SMECEC and MECEC algorithms in comparison to MEESS and NLISS algorithms in Table 3.

It is important pointing out that from the energy point of the view both SMECEC and MECEC have almost the same performances as can be seen in figures 4 to 7. In the following the detail explanations of the performance evaluations of the aforementioned algorithms are presented. Fig. 3 presents the changes of maximum network lifetime versus \( \alpha \) for different values of \( \lambda_{th} \). Note that for each value of \( \lambda_{th} \), CSS is performed with those subsets of sensors that have less average energy consumptions. It is obvious that for a given \( \lambda_{th} \) increasing \( \alpha \) leads to decreasing in \( F(\alpha) \). To explain the reasons that lead to this curve trends for different values of \( \lambda_{th} \), the case of \( \alpha = 1 \) is considered. As can be seen in Fig. 3, the value of \( F(\alpha) \) for \( \lambda_{th} = 10 \) is less than that for other values of \( \lambda_{th} \). Note that the subsets involved in CSS are periodically chosen for CSS until the energy levels of some of their members reach to 10 which is closer to the minimum energy level i.e. \( E_{min} = 0.1 \) in here) when compared to other values of \( \lambda_{th} \). For this reason, the sensors of such subsets can be alive for few frames more after being exempted from performing CSS. As soon as one sensor of such subset is dead, we record \( F(\alpha) \). When the value of \( \lambda_{th} \) is increased, the number of frames spent by the subsets involved in CSS until their energy reach to \( \lambda_{th} \), decreases. This means: (A) the time duration that such subsets can be alive after being exempted from CSS, increases and (B) the time duration takes that other eligible subsets start CSS after exemption of the former subsets, decreases. From (A), we conclude that those subsets of sensors that finish their participation in CSS, can be alive longer when \( \lambda_{th} \) increases. This means that (A) helps increasing the value of \( F(\alpha) \). Conversely, (B) implies that the subsets of sensors are selected for CSS more frequently. This means that subsets consume their energy faster and therefore reduction in \( F(\alpha) \). In other words, while (A) tries to increase \( F(\alpha) \), (B) tries to decrease \( F(\alpha) \). From Fig. 3 we can find that for \( \alpha = 1 \), when \( \lambda_{th} \) increases from 10, (A) overcomes the effect of (B). Hence, \( F(\alpha) \) increases. At \( \lambda_{th} = 100 \), the value of \( F(\alpha) \) reaches to its maximum value. After that it is (B) that plays dominant role and therefore \( F(\alpha) \) starts decreasing.

Table 3. The features of SMECEC, MECEC, MEESS and NLISS algorithms.

<table>
<thead>
<tr>
<th>Feature</th>
<th>SMECEC</th>
<th>MECEC</th>
<th>MEESS</th>
<th>NLISS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exclusion of unnecessary reports to FC</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Low complexity</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Sensor selection with respect to dynamic energy constraints</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Considering all desired sensor subsets</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Error-prone reporting channels</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Performance evaluation for different lifetime definitions</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

Similar discussion can be done for other values of \( \alpha \). As can be seen in Fig. 3, for each value of \( \alpha \), the nonlinear effects of (A) and (B), result in different values of \( \lambda_{th} \), at which \( F(\alpha) \) reaches to its maximum value. As can be seen in the figure, for \( \lambda_{th} = 50 \) the MECEC algorithm provides higher values of maximum network lifetime for a wider range of \( \alpha \). Hence, we have considered \( \lambda_{th} = 50 \) for the rest of the simulations. Similar curves have been obtained for SMECEC algorithm. Further to that the best value for \( \lambda_{th} \) in SMECEC algorithm which provides higher values of maximum network lifetime for a wider range of \( \alpha \) is equal to 50, too. For the sake of convenient appearance, the results of SMECEC algorithm are not presented in Fig. 3. Fig. 4 presents the maximum network lifetime, \( F(\alpha) \), versus different values of \( \alpha \) in MECEC, SMECEC, MEESS and NLISS algorithms. The figure reveals that for the values of \( \alpha \) between 0.3 and 0.9, the maximum network lifetime in MECEC is significantly more than those of MEESS and NLISS. Nevertheless, for the values of \( \alpha \) between 0.9 and 1, the maximum network lifetime using NLISS is a few percent more than the same when MECEC is used. This is because NLISS periodically calculates the priority function introduced in (32) for all sensors considering their remaining energy levels. As can be observed, the network lifetime in NLISS has the same value for all values of \( \alpha \). The reason is that the priority function considers the remaining energy levels of sensors frame by frame. In fact, the sensors with lower remaining energy levels receive lower priorities for CSS. Therefore, the majority of sensors are kept alive until the end of network lifetime. However, such lifetimes for NLISS are obtained at the price of performing a significant volume of computations.

Fig. 5 presents the average energy consumption for CSS per frame, \( E_{ave} \), in SMECEC, MECEC, NLISS and MEESS algorithms. As can be seen in the figure, \( E_{ave} \) in MECEC algorithm has significantly lower value compared to NLISS and MEESS. As can also be seen, SMECEC and MECEC algorithms have nearly equal average energy consumptions for CSS per frame. Fig. 6 presents the total energy consumption for CSS during the network lifetime, \( E_{tot} \) versus the number of sensors in MEESS, NLISS and MECEC and SMECEC.
algorithms. As can be seen in the figure, the total energy consumption for CSS in MECEC is considerably less than those of MEESS and NLISS. The reason is that MECEC algorithm selects the sets with the lowest average energy consumption for CSS as much as possible. Also, as can be seen in Fig. 6, total energy consumption for CSS in MECEC and SMECEC algorithms is approximately equal. Fig. 7 presents the total energy amount required for reporting the sensing results to FC, $E_{rep}$, versus the number of sensors in SMECEC, MECEC, MEESS and NLISS algorithms. From this figure we can state that the value of $E_{rep}$ in MECEC is considerably lower than those of MEESS and NLISS. The reason is that a large number of unnecessary reports from sensors to FC have been removed in MECEC algorithm. In addition, as can be seen in Fig. 7, the total energy consumption for CSS in MECEC algorithm is almost same to that in SMECEC algorithm. Fig. 8 shows the average total number of additions and multiplications in MECEC, SMECEC, MEESS, and NLISS algorithms. As it is expected from the pseudo-code of SMECEC algorithm and can be seen in Fig. 8, the average total number of computations in SMECEC is much lower than other existing algorithms. It should be noted that the average total number of additions in SMECEC algorithm is in the range of [10,40] and negligible compared to other existing methods.

![Fig. 5. Average energy consumption for CSS per frame.](image)

![Fig. 6. $E_v$ versus number of sensors.](image)

![Fig. 7. $E_{rep}$ versus number of sensors.](image)

![Fig. 8. a) Average total number of additions, b) multiplications.](image)

### VII. CONCLUSIONS

In this paper a new method was proposed to form all the subsets of sensors which can cooperatively provide the network with desired false alarm and detection probabilities. For each subset of sensors with such conditions, the average energy consumption for CSS per frame was computed. Then, a new heuristic algorithm was proposed to select the subset minimizing the average energy consumption for CSS in each frame. The simulation results show that the proposed algorithms have better performances in terms of maximum network lifetime, energy consumption for reporting the results to FC and CSS compared to other existing methods. It can be seen that the average energy consumption for CSS per frame is reduced up to 35% in comparison to the state-of-the-art research works. In addition, the total reporting energy for CSS is reduced up to 67% compared to the existing methods. However, since the reporting energy is a part of energy consumption for CSS, the total energy consumption for CSS is decreased for at most 36%. Finally, the proposed algorithm increases the maximum network lifetime up to 39%. To reduce the complexity of the proposed algorithm, a sub-optimal algorithm was proposed which has the same performance from energy-efficiency point of view. However, the computational complexity of the sub-optimal algorithm is significantly lower than the proposed heuristic algorithm.

### REFERENCES


This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2015.2503324, IEEE Sensors Journal


[50] Part 15.4:Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), IEEE 802.15.4 Standard, 2006.

Maryam Monemian received her B.Sc. degree in electrical engineering in 2007, and the M.Sc. degree in Network Communications engineering in 2010, both from the Isfahan University of Technology, Isfahan, Iran. She is currently pursuing the Ph.D. degree in communications at the Isfahan University of Technology, Isfahan, Iran. Her current research interests include spectrum sensing in wireless cognitive sensor networks, energy consumption modeling, bandwidth allocation and ad-hoc relaying in wireless systems.

Mehdi Mahdavi received his B.Sc. degree in electrical (Electronics) engineering from Isfahan University of Technology, the M.Sc. degree in electrical engineering from Amirkabir (Polytechnique) University of Technology, Tehran, Iran, and the Ph.D. degree in electrical engineering (Mobile Communication Networks) from the University of Sheffield, Sheffield, U.K. From 2004 to 2007, he was a member of the ICT faculty, Ministry of ICT, Tehran, Iran. In 2007, he joined the Electrical and Computer Engineering Department of Isfahan University of Technology where he is currently an associate professor. From 2008 to 2011, he was a member of the NGN committee of the Telecommunication Company of Iran (TCI), Tehran, Iran. He is the founder and the head of Advanced Network Traffic Engineering Lab (ANTELL). His research interests include Capacity planning of Wireless Mesh Networks; Cognitive Radio including: MAC protocols design and analysis, QoS analysis, Cooperative Spectrum Sensing; Teletraffic theory, network performance evaluation, network calculus, and telecommunication in general.

Mohammad Javad Omidi received his Ph.D. from University of Toronto in 1998. He has gained years of industry experience in Canada as the cofounder of an active research and development group designing broadband communication systems. He is an Associate Professor in the Department of Electrical and Computer Engineering, Isfahan University of Technology, Iran, and has been the chair of Information Technology Center (2005–2008), and chair of Electrical and Computer Engineering Department (2008–2011) and manager of Communication Group (2011–2012) at this university. His research interests are in the areas of Mobile Computing, Wireless Communications, Digital Communication Systems, Software Defined Radio and Cognitive Radio Systems, and VLSI Architectures for Communication Algorithms. He has several publications, US and international patents and inventions on all the areas of his research interest.