Research Article

A Multiobjective Evolutionary Algorithm for Energy-Efficient Cooperative Spectrum Sensing in Cognitive Radio Sensor Network

Weirong Liu, Gaorong Qin, Shuo Li, Jian He, and Xiaoyong Zhang

School of Information Science and Engineering, Central South University, Changsha, Hunan 410000, China

Correspondence should be addressed to Xiaoyong Zhang; zhangxy@csu.edu.cn

Received 8 August 2014; Accepted 29 September 2014

Academic Editor: Shigeng Zhang

Copyright © 2015 Weirong Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Cognitive radio has emerged as a promising solution to address the problems posed by coming spectrum scarcity for the inherently resource-constrained sensor networks. Reliability and energy consumption are key objectives for spectrum sensing in cognitive sensor networks. In this paper, a fast differential evolution algorithm is proposed to optimize the energy consumption and spectrum sensing performance jointly. By constructing a comprehensive performance metric, the joint optimization is transferred to a multiobjective optimal problem, in which the sleeping schedule and censoring mechanism are taken into consideration. The main objective of the proposed algorithm is to minimize the network energy consumption subjected to constraints on the detection performance by optimally deriving the censoring and sleeping probabilities. To accelerate the convergence speed and maintain the diversity, the algorithm utilizes the advantages of opposite-based learning for generating the initial population and a tournament scheme in mutation step. In the crossover step, a control parameters dynamic adjustment scheme is applied to make a trade-off between exploration and exploitation. Finally, a selection mechanism is introduced for generating a well-distributed Pareto optimal front. The simulation results show that the proposed algorithm can reduce the average energy consumption of cognitive sensor node, while improving the global probability of spectrum sensing.

1. Introduction

Wireless Sensor Network (WSN), as an important information acquisition technology, is a basic method to realize the fusion of physical world and information world [1–3]. However, the traditional fixed spectrum allocation cannot be able to meet the rapid development of wireless sensor network. A number of spectrum measurements [4–6] have shown that licensed spectra are underutilized and there exist spectrum portions unused over space and time. Therefore, to improve the utilization of such spectrum portions, dynamic spectrum access models based on cognitive radio have been proposed recently [7, 8]. Cognitive radio technology makes secondary users able to access the vacant spectrum with limiting harmful interference to licensed primary users. To avoid causing interference to primary users, spectrum sensing is employed to determine whether the channels are vacant or not. Therefore, reliable spectrum sensing for the determination of vacant channels is a critical problem.

Incorporating cognitive radio technology into wireless sensor network yields the cognitive radio sensor network. By dynamically changing operating parameters, the cognitive radio sensor node could sense the vacant spectrum and makes use of these available spectra in an opportunistic manner to improve the overall spectrum utilization [9, 10]. Spectrum sensing technology plays an important role to improve the reliability for the determination of vacant channels. Nevertheless, wireless communication, characterized by multipath fading and adjacent channel interference, may be greatly influenced by surrounding environment. What is more, as for the communication and processing resource-constraint of sensor nodes, the stand-alone spectrum sensing by using the traditional approaches, such as matched filtering, energy sensing, and feature detection, may not be reliable.
Based on the traditional spectrum sensing approaches, a variety of cooperative spectrum sensing technologies have been proposed for improving the spectrum sensing performance in cognitive radio sensor network. For example, a consensus scheme between the secondary users has been introduced to design an optimal sensing selection mechanism to improve the performance by predicting the primary users’ action [11]. In [12], an efficient one-bit decision based on three-phase spatiotemporal sensing algorithm, which is suitable for large-scale distributed wireless sensor network, is proposed by sensing vacant spectrum in both time and space domain. Similarly, a parallel cooperative spectrum sensing mechanism is proposed to maximize the throughput of data [13].

Nevertheless, many existing research efforts on improving the spectrum efficiency neglect the energy-constraint of cognitive wireless sensor nodes. The increasing energy consumption will reduce the lifetime of sensor node. Therefore, a distributed spectrum-aware clustering mechanism is proposed in [14], which can reduce the energy consumption by optimizing the number of clusters. Based on the censoring scheme, [15] proposed a cooperative spectrum sensing scheme by utilizing a Takagi-Sugeno fuzzy inference system to make a decision about the presence of primary users. These methods save the energy by reducing the information transmission.

Works presented in [16–18] make further efforts to optimize the energy efficiency by combining sleeping schedule. However, the decreasing energy consumption may degrade spectrum sensing performance and directly affects the accuracy of spectrum decision.

In this paper, we consider the energy-efficient optimization for cooperative spectrum sensing in cognitive radio sensor network with a combination of sleeping schedule and censoring scheme. The objective is to minimize the energy consumption for spectrum sensing as well as guaranteeing the performance. The optimization problem is formulated as a multiobjective optimization problem by analyzing the global probability of spectrum sensing and the global probability of false alarm. Typically, multiobjective optimization problem is often converted to a single-objective optimization problem [19, 20] by predefining weighting factors for different objectives, describing the relative importance of each objective. However, the optimal solution of a multiobjective optimization problem is generally a set of optimal solutions (largely known as Pareto optimal solutions), which is not the same as in single-objective optimization. If a classical optimization method is used for finding the solutions, it may take much time to converge and may not converge to optimal solutions [21].

During the past decade, evolutionary algorithm has gained popularity in solving difficult multiobjective optimization problems since it is capable of dealing with objective functions, which are not mathematically well behaving, being, for example, discontinuous, nonconvex, multimodal, and nondifferentiable [22–25]. Differential evolution (DE) algorithm, introduced by Storn and Price in 1997, is one of the most popular evolutionary algorithms [26] and may be used in a wide variety of highly nonlinear and complex optimization problems. This algorithm is simply structured and is easy to use, while demonstrating great robustness and fast convergence in solving single-objective global optimization problems [27, 28]. Recently, there are several attempts to extend the DE to solve multiobjective problems [29, 30], with superior performance to other multiobjective algorithms widely reported and verified [31].

Based on the aforementioned discussion, a fast multiobjective differential evolution algorithm is proposed to solve the multiobjective optimization problem formulated in this paper. Different from classical DE, this proposed algorithm applies an opposition-based learning technique for population initialization operation. Opposition numbers are used to improve the exploration and convergence capabilities of the optimization process. Besides, the performance of the DE algorithm is highly dependent on the control parameters. Therefore, to further accelerate the convergence rate without losing solution diversity on the Pareto front, this paper introduces a tournament scheme in mutation step and a control parameters dynamic adjustment scheme in crossover step. Finally, a selection mechanism is introduced for generating a well-distributed Pareto optimal front.

The remainder of the paper is organized as follows. In Section 2, the cooperative spectrum sensing system model and optimization problem are presented to incorporate the sleeping schedule and censoring scheme. In Section 3, the optimization problem is analyzed and a fast multiobjective differential evolution algorithm is proposed to solve it. A comprehensive performance evaluation is given in Section 4 to show the performance of the proposed algorithm. Section 5 gives conclusion and future work.

2. System Model and Problem Formulation

We consider a cognitive radio sensor network that shares a common frequency band with primary users. This paper is of interest to formulate the energy-efficient cooperative spectrum sensing as a multiobjective optimization problem. In this section, we present the models for cooperative spectrum sensing energy-efficient optimization in cognitive radio sensor network to analyze sensing performance and the energy consumption including spectrum sensing model and global sensing model.

2.1. Spectrum Sensing Model. As shown in Figure 1, we consider a cognitive radio sensor network with a fusion center and N cognitive sensor nodes. Each node senses the licensed spectrum by energy sensing and then makes its own local decision for the spectrum utilization and sends the decision to the fusion center. The fusion center collects these local decisions using a certain rule and makes the final decision [32, 33]. The fusion can be soft fusion or hard fusion [34]. We pay attention to hard fusion because hard fusion scheme only needs to send one bit per decision. This mechanism can save the energy consumption for transmission. The fusion center receives the results from sensor nodes and makes the final decision that will be sent back to the sensor nodes.
The local decision made by cognitive sensor node can be regarded as a binary decision between 0 and 1, where 0 denotes that the spectrum is available to secondary nodes and 1 represents otherwise. Cognitive sensor node senses the licensed spectrum under binary hypotheses $H_0$ and $H_1$, where hypothesis $H_0$ represents the absence of the primary signal and hypothesis $H_1$ denotes the presence of the primary signal. Sensor node collects $T_0$ samples during one sampling process. Therefore, the $k$th observation sample $e_i[k]$ sensed by node $i$ follows the following data model:

$$H_0 : e_i[k] = n_i[k], \quad k = 1, \ldots, N,$$

$$H_1 : e_i[k] = s_i[k] + n_i[k], \quad k = 1, \ldots, N,$$  \hspace{1cm} (1)

where $N$ is the number of cognitive radio sensor nodes, $s_i[k]$ is the received signal form primary users, and $n_i[k]$ is the white Gaussian noise signal with zero mean and variance $\sigma_n^2$. Sensor node senses the licensed spectrum by energy sensing, and the received accumulated energy collected over $T_0$ observation samples at the node $i$ can be given by

$$E_i = \sum_{k=1}^{T_0} x_i^2[k].$$ \hspace{1cm} (2)

Afterwards a censoring policy is employed at each sensor node [35]. The censoring thresholds $\lambda_1$ and $\lambda_2$ are employed at each sensor node. The range $\lambda_1 < E_i < \lambda_2$ is called the censoring region. If the collected energy is in the censoring threshold range, cognitive sensor node will not send the local decision to fusion center. The local censoring decision rule is given as follows:

$$d_{li} = \begin{cases} 1, & E_i \geq \lambda_2, \\ \text{no decision}, & \lambda_1 < E_i < \lambda_2, \\ 0, & E_i \leq \lambda_1, \end{cases} \hspace{1cm} (3)$$

where $d_{li}$ is the local decision made by sensor node $i$. $d_{li} = 1$ declares the presence of the primary signal and $d_{li} = 0$ declares the absence of the primary signal. If $E_i$ is located in the censoring region, the sensor node makes no decision.

In the real environment, for each cognitive sensor node, the average received signal-to-noise ratio (SNR) is different. Therefore, it is difficult to design the system parameter. For analytical tractability, we assume that the received signal-to-noise ratio (SNR) at each radio is the same, denoted by $\gamma$. It is reasonable that the probability of spectrum sensing $P_{dl}$ and the probability of false alarm $P_{fl}$ are the same for all nodes. Under this model, work presented in [36] shows that $E_i$ follows the central chi-squared distribution with $2T_0$ degrees of freedom under the hypothesis $H_0$. Oppositely, when the hypothesis is $H_1$, $E_i$ follows the noncentral chi-squared distribution with $2T_0$ degree of freedom and the noncentrality parameters $2\gamma$. Therefore, based on the above decision rule, the local probability of false alarm can be written as

$$P_{fl} = P_i (E_i \geq \lambda_2 | H_0) = \frac{\Gamma (T_0, \lambda_2/2)}{\Gamma (T_0)},$$ \hspace{1cm} (4)

and the local probability of spectrum sensing can be written as

$$P_{dl} = P_i (E_i \geq \lambda_2 | H_1) = Q_{T_0} \left( \sqrt{2\gamma}, \sqrt{\lambda_2} \right),$$ \hspace{1cm} (5)

where $\Gamma(a, x)$ is the incomplete Gamma function with $\Gamma(a, 0) = \Gamma(a)$, given by

$$\Gamma(a, x) = \int_{x}^{\infty} t^{a-1} e^{-t} dt.$$ \hspace{1cm} (6)

And $Q(a, x)$ is a generalized Marcum Q-function defined as follows:

$$Q(a, x) = \left( \frac{1}{a^{a-1}} \right) \int_{x}^{\infty} t^{a-1} e^{-t} I_{a-1} (a \cdot t) dt,$$ \hspace{1cm} (7)

where $a, \lambda_2$ are nonnegative real numbers and $u$ is a positive integer. In this expression, $I_{a-1}(\cdot)$ is the modified Bessel function of the first kind of order $u - 1$ and it is a monotone increasing function. Marcum Q-function is generally used to analyze the signal detection problem and the bit error rate of channel interference problem. Here, it is used to solve the probability of spectrum sensing.

### 2.2 Global Spectrum Sensing Model

For the purpose of minimizing energy consumption, we integrate sleeping schedule and censoring scheme into cooperative spectrum sensing. The status of each cognitive sensor node is determined by two schemes: (1) a sleeping schedule scheme determines whether or not the node is awake and (2) a censoring scheme determines whether or not it transmits its local decision to fusion center. The sleeping rate, that is, the probability that a node is turned off, is denoted by $\mu$. It means that only part of nodes sense the primary spectrum and the others will turn into the sleeping state with low energy consumption. Correspondingly, denote $\rho$ to be the censoring rate, that is, the probability which is adopted to filter those less-than-ideal decisions for reducing unnecessary data transmission. Sensor
nodes, which are not in the sleeping state, sense the channel signal energy at the specialized time slot and make a local decision for the spectrum occupancy. The local decision can be regarded as a binary decision between 0 and 1, where 0 denotes that the spectrum is available to secondary nodes and 1 represents the opposite. In the end, the local decision will be sent to the fusion center.

The reported local decisions are combined at the FC and the final decision regarding the presence or absence of the primary user is made according to a certain fusion rule. However, several fusion schemes have been proposed in literature [37–39]. We consider a hard fusion scheme due to its improved energy and bandwidth efficiency. Among them, the OR and AND rules have been studied extensively in literature [40–42], since these rules are easily implementable by simple logics. And [42] shows that the OR rule outperforms the AND rule in terms of energy efficiency; thus, we employ the OR rule. In a similar way, the global probability of spectrum sensing based on the simplified equation can be explained by the OR rule.

In the process of spectrum sensing, the energy consumption of sensor node mainly includes spectrum sensing. The local decision can be regarded as a binary decision between 0 and 1, where 0 represents the opposite. In the end, the local decision will be sent to the fusion center. The term \( P_r(D_{fc} = 1 \mid H_0, L, K) \) is the probability that fusion center makes a false decision. That is, the probability that the channel is declared occupied, is conditioned on hypothesis \( H_0 \) for fixed \( K \) and \( L \).

We can get a further simplified form of (8) using the binomial expansion theorem. After some algebraic manipulation, we can obtain

\[
Q_F = 1 - \{1 - (1 - \mu)(1 - \rho) P_{dl}\}^N. \tag{9}
\]

This simplified equation can be explained by the OR rule based global probability of false alarm when considering \((1 - \mu)(1 - \rho) P_{dl}\) to be the local probability of false alarm with sleeping schedule and censoring scheme integrated. By using the “OR rule”, fusion center could make an accurate decision for whether the spectrum is occupied by primary users or not.

In a similar way, the global probability of spectrum sensing can be derived as

\[
Q_D = P_r(D_{fc} = 1, L \geq 1, K \geq 1 \mid H_1)
\]

\[
= \sum_{K=1}^{N} P_r(D_{fc} = 1, L \geq 1, K \mid H_1)
\]

\[
= \sum_{K=1}^{N} P_r(K \mid H_1) P_r(D_{fc} = 1, L \geq 1 \mid H_1, K)
\]

\[
= \sum_{K=1}^{N} \left( \frac{N}{K} \right) \mu^{N-K} (1 - \mu)^K \cdot \sum_{L=1}^{N} P_r(D_{fc} = 1, L \mid H_1, K) \]

\[
= \sum_{K=1}^{N} \left( \frac{N}{K} \right) \mu^{N-K} (1 - \mu)^K \cdot \sum_{L=1}^{K} P_r(L \mid H_1, K) P_r(D_{fc} = 1 \mid H_1, K, L)
\]

\[
= \sum_{K=1}^{N} \left( \frac{N}{K} \right) \mu^{N-K} (1 - \mu)^K \cdot \sum_{L=1}^{K} \left( \frac{K}{L} \right) \rho^{K-L} (1 - \rho)^L \cdot \left[ 1 - (1 - P_{dl})^L \right]
\]

\[
= 1 - \{1 - (1 - \mu)(1 - \rho) P_{dl}\}^N,
\]

where \( P_{dl} \) is local probability of false alarm given by (4). \( P_r(K \mid H_0) \) denotes the probability that \( K \) cognitive sensor nodes do not turn into sleeping state under hypothesis \( H_0 \). \( P_r(L \mid K, H_0) \) is the probability that \( L \) out of \( K \) awake cognitive sensor nodes, for a fixed \( K \) and \( H_0 \), send the decision to the fusion center. The term \( P_r(D_{fc} = 1 \mid H_0, L, K) \) is the probability that fusion center makes a false decision. That is, the probability that the channel is declared occupied, is conditioned on hypothesis \( H_0 \) for fixed \( K \) and \( L \).

2.3. The Energy Consumption and Multiobjective Optimization. In the process of spectrum sensing, the energy consumption of sensor node mainly includes spectrum sensing.
and data transmission. For the entire cognitive radio sensor network, the average energy consumption is given by

\[ C_T = (1 - \mu) \sum_{i=1}^{N} (C_{S_i} + C_{T_i} (1 - \rho)) , \]  

(11)

where \( \rho = P_r (\lambda_1 < E_i < \lambda_2) \) denotes the censoring rate. \( C_{S_i} \) and \( C_{T_i} \) represent the energy consumption of the \( i \)th node for spectrum sensing and data transmission, respectively. The spectrum sensing energy consumption includes monitoring channel, sampling, and local decision. The data transmission energy consumption is the energy consumption for the sensor node to send 1 bit decision to fusion center.

We describe the probability that the spectrum is available as \( \pi_0 = P_r (H_0) \) and the probability that the primary user is transmitting data is \( \pi_1 = P_r (H_1) \). In this paper, it is assumed that the \( \pi_0 \) and \( \pi_1 \) are unknown and that \( \pi_1 \) is much smaller than \( \pi_0 \), reflecting channel underutilization. In this case, we can follow the definition of [35] for the censoring rate under blind Neyman-Pearson setup

\[ \rho = P_r (\lambda_1 < E_i < \lambda_2 \mid H_0) . \]  

(12)

To connect the probability and the threshold values, we can change (12) by combining (3) as follows:

\[ \rho = \frac{\Gamma (T_0, \lambda_1/2) - \Gamma (T_0, \lambda_2/2)}{\Gamma (T_0)} . \]  

(13)

In order to guarantee the spectrum sensing performance, our main objective is to optimize the global probability of spectrum sensing and the global probability of false alarm. In practice, it is desirable to have the global probability of spectrum sensing close to zero and the global probability of false alarm close to unity. In such cases, the spectrum can be fully utilized and will not cause interference to the primary user. Nevertheless, it is difficult to balance the energy consumption and global sensing performance. Therefore a joint optimization of energy consumption, global probability of false alarm, and spectrum sensing is built to balance the energy consumption and sensing performance. This multiobjective joint optimization problem can be described as follows:

\[ \min F(X) = (C_T, Q_{f_1}, (-Q_{D}))^T \]

s.t.  
\[ 0 < \mu < 1 \]  
\[ \lambda_1 < \lambda_2 \]  
\[ 0 < \rho < 1 , \]  

(14)

where \( X = (\mu, \lambda_1, \lambda_2) \), the constraint \( \rho \) can be obtained by (12). We take the opposite of the global probability of spectrum sensing to uniform the optimization objective.

3. Proposed Algorithm

In this section, we first illustrate the relationship between the Pareto solutions and the multiobjective optimization and then provide a fast multiobjective differential evolution algorithm to solve the multiobjective optimization problem. To accelerate the convergence rate and maintain the diversity, a dynamical adjustment scheme with a crossover parameter and a new population selection scheme based on the concept of Pareto optimum solutions are proposed, which could minimize the energy consumption under the performance constraint.

3.1. Multiobjective Optimization and Pareto Solutions. The traditional single objective optimization problem is to seek the optimal solution for a certain objective. The mathematical description can be described as follows:

\[ \min f(x) \]

\[ \text{s.t. } g_i(x) \leq 0, \quad i = 1, 2, \ldots, q \]

\[ h_j(x) = 0, \quad j = 1, 2, \ldots, p , \]  

(15)

where \( f(x) \) is a scalar function, \( g_i(x) \) \( (i = 1, 2, \ldots, q) \) defines \( q \) inequality constraint functions, and \( h_j(x) \) \( (j = 1, 2, \ldots, p) \) defines \( p \) equality constraints functions.

However, for the multiobjective optimization problem, the objective function will no longer be a scalar function, but a multidimensional vector. It is assumed that the multiobjective optimization problem has \( n \) decision variables and \( m \) objective functions, it can be expressed as follows:

\[ \min F(X) = (f_1(X), f_2(X), \ldots, f_m(X))^T \]

\[ \text{s.t. } g_i(X) \leq 0, \quad i = 1, 2, \ldots, q \]

\[ h_j(X) = 0, \quad j = 1, 2, \ldots, p , \]  

(16)

where \( X = (x_1, x_2, \ldots, x_n) \in X^n \subset R^n \) is the \( n \) dimensional solution vector. \( X^n \) denotes the \( n \) dimensional solution space and \( F(x) \) denotes \( m \) objective functions. \( g_i(X) \) and \( h_j(X) \) define \( q \) inequality constraints functions and \( p \) equality constraints functions, respectively. Currently, the solution of multiobjective problem is mainly based on the Pareto optimal solution. Thus, the multiobjective optimization problem is to determine the particular set of values which yield the optimum values of the objective function. Here, we give the following Pareto definition introduced in [43].

Definition 1 (feasible solution). Assuming that a certain solution \( X \in X^n \) satisfies all the constraints conditions which are represented as \( g_i(x) \leq 0 \) \( (i = 1, 2, \ldots, q) \) and \( h_j(x) = 0 \) \( (j = 1, 2, \ldots, p) \) for the multiobjective optimization, the solution \( X \) is called feasible solution.

Definition 2 (feasible solution set). The collection which consists of all the feasible solutions is defined as a feasible solution set \( X_f \), and \( X_f \subseteq X^n \).

Definition 3 (Pareto dominance). If two feasible solutions \( X_1, X_2 \in X_f \) satisfy (17), then \( X_1 \) is Pareto dominance in
comparison with $X_2$. That can be denoted by $X_1 \preceq X_2$ and can also be called $X_1$ dominating $X_2$. One has
\begin{align}
\forall i \in \{1, 2, \ldots, m\}, & \quad f_i(X_1) \leq f_i(X_2), \\
\exists j \in \{1, 2, \ldots, m\}, & \quad f_j(X_1) < f_j(X_2).
\end{align}

**Definition 4** (Pareto optimal solution). If the feasible solution $X$ satisfies (18), then $X$ is the Pareto optimal solution. Consider
\[ \neg \exists X' \in X_f : F\left(X'\right) \preceq F\left(X\right). \]

**Definition 5** (Pareto optimal solution set). All the Pareto optimal solutions compose the Pareto optimal solution set. It can be defined as
\[ P^* := \{ X \in X_f | \neg \exists X' \in X_f : F\left(X'\right) \preceq F\left(X\right)\}. \]

**Definition 6** (Pareto front). The Pareto front $P^*$ which consists of target vectors corresponding to all the Pareto optimal solutions in the Pareto optimal solution set $P^*$ can be defined as
\[ \text{PF}^* \triangleq \{ F\left(X\right) = (f_1\left(X\right), f_2\left(X\right), \ldots, f_m\left(X\right))^T | X \in P^* \}. \]

### 3.2. Differential Evolution Algorithm

Differential evolution (DE) algorithm is a kind of effective stochastic parameter optimization algorithm [26]. Like other evolution algorithms, the differential evolution algorithm is population based evolution search strategy. For the purpose of reducing the algorithm complexity and improving its practicability, this algorithm is different in some respects, such as real-number encoding, simple differential mutation, and the tournament scheme. Differential Evolution is a parallel direct search method which utilizes $P$ $D$-dimensional parameter vectors as a population for each generation $G$. The parameter $P$ does not change during the minimization process.

The initial vector population is chosen randomly and must cover the entire parameter space. As a rule, the initial population should be a uniform probability distribution. By adding the weighted difference between two population vectors to the third vector, the differential evolution algorithm generates a new parameter vector. This operation is called mutation. The mutated vector’s parameter then is mixed with the parameters of another predetermined target vector to yield the trial vector. This operation is called crossover.

Finally, comparing the value of the trial vector and the target vector, the vector that yields a lower cost function is chosen to replace the target vector in the following generation. This last operation is called selection.

The process can be described as follows.

#### 3.2.1. Initial Population Vector

The initial parameter vector, which follows the uniform probability distribution, is chosen randomly from the entire parameter space. We denote the parameter as $X_{i,G}, i = 1, 2, \ldots, N_p$, where $G$ is the population generation.

#### 3.2.2. Mutation

For each target vector $X_{i,G}$, a mutant vector is generated from three population vectors $\{X_{r_1}, X_{r_2}, X_{r_3}\}$, which is chosen form population vectors according to
\[ V_{i,G+1} = X_{r_1,G} + F\left(X_{r_3,G} - X_{r_3,G}\right), \]
where $i = \{1, 2, \ldots, P\}$ and indexes $r_1$, $r_2$, and $r_3$ are mutually different random integer indices selected from $\{1, 2, \ldots, P\}$. $F \in [0, 2]$ is the control parameter which controls the amplification of the differential variation. Larger values for $F$ result in higher diversity in the generated population and lower values cause faster convergence.

#### 3.2.3. Crossover

In order to increase the diversity of the perturbed parameter vectors, crossover is introduced. The crossover operation is used to generate new solutions by shuffling competing vectors, and it can also increase the diversity of the population. The main objective is to generate the trial vector as follows:
\[ U_{i,G+1} = (u_{i1,G+1}, u_{i2,G+1}, \ldots, u_{iD,G+1}), \]
where $D$ denotes the number of parameters in the optimization problem. The traditional crossover is binomial crossover. To this end, the trial vector can be obtained by
\[ u_{ij,G+1} = \begin{cases} v_{ij,G+1} & \text{if } \text{randb}\left(j\right) \leq \text{CR} \text{ or } j = \text{rnbr}\left(i\right) \\
X_{ij,G} & \text{if } \text{randb}\left(j\right) > \text{CR} \text{ and } j \neq \text{rnbr}\left(i\right), \end{cases} \]
where randb($j$) is the $j$th element’s evaluation which is a uniform random number generator. CR $\in [0, 1]$ is the crossover constant which has to be determined by the user. rnbr($j$) $\in \{1, 2, \ldots, D\}$ is a randomly chosen index which makes sure that $U_{i,G+1}$ gets at least one parameter from $V_{i,G+1}$. Figure 2 shows the crossover process.

#### 3.2.4. Selection

To make a decision for whether or not it should become a member of generation $G + 1$, we should compare the trial vector $U_{i,G+1}$ with the target vector $X_{i,G}$. If the trial vector $U_{i,G+1}$ yields a lower cost function value than $X_{i,G}$, then $X_{i,G+1}$ is set to $U_{i,G+1}$; otherwise the old value is retained.

### 3.3. Fast Multiobjective Differential Evolution Algorithm

There are some limitations for using the traditional differential evolution algorithm directly to solve the multiobjective optimization problem. Therefore, we need to improve the convergence speed and maintain the diversity.

As mentioned above, the differential evolutionary optimization method starts with the initial population and tries to improve them toward some optimal solutions. The computation time is related to the distance of these initial random solutions from the optimal solutions. We can improve our chance of starting with a fitter solution by simultaneously computing its opposite solution. In fact, according to probability theory, an individual will be further from the solution.
than its opposite individual 50% probability. Therefore, starting with the closer of the two guesses has the potential to accelerate convergence. The same approach can be applied not only to initial solutions, but also continuously to each solution in the current population [44]. As introduced in [45], the opposite point has a straightforward definition as follows.

**Definition 7** (opposite number). Let \( x \in [a, b] \) be a real number. The opposite number \( \bar{x} \) is defined by

\[
\bar{x} = a + b - x
\]  

Similarly, the definition is generalized to higher dimensions as follows.

**Definition 8** (opposite point). Let \( X = (x_1, x_2, \ldots, x_D) \) be a point in a \( D \)-dimensional space, where \( x_i, x_2, \ldots, x_D \in \mathbb{R} \) and \( x_i \in [a_i, b_i] \) \( \forall i \in \{1, 2, \ldots, D\} \). The opposite point \( \bar{X} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_D) \) is defined by

\[
\bar{x}_i = a_i + b_i - x_i.
\]  

By applying the definition of opposite point, the opposition-based optimization can be defined as follows.

**Opposition-Based Optimization.** Let variable \( X = (x_1, x_2, \ldots, x_D) \) be a point in a \( D \)-dimensional space (i.e., a candidate solution). Assume \( f(X) \) is a fitness function which is used to evaluate the candidate’s fitness. According to the definition of the opposite point, \( \bar{X} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_D) \) is the opposite of \( X = (x_1, x_2, \ldots, x_D) \). If \( f(\bar{X}) \) is better than \( f(X) \), then update \( X \) with \( \bar{X} \); otherwise keep the current point \( X \). Hence, the current point and its opposite point are evaluated simultaneously in order to continue with the fitter one.

From the above analysis in Section 3.1, we need to solve the multiobjective optimization problem with three objective functions \( C_T, Q_f, \) and \(-Q_D \) and a solution space \( X = (\mu, \lambda_1, \lambda_2) \). For convenience, we make some assumptions: \( x_1 = \mu, x_2 = \lambda_1, \) and \( x_3 = \lambda_2 \). Thereby, the solution space can be expressed as \( X = (x_1, x_2, x_3) \). To solve this problem, a fast multiobjective differential evolution algorithm is proposed using opposite-based learning scheme. Similar to the population-based optimization algorithms, we use opposite-based learning to generate the initial population. For the sake of rapid convergence and diversity maintenance, we have modified each process using different schemes. The improved differential evolution can be described as follows.

### 3.3.1. Initial Population Vector

In the case of no prior knowledge, initial population vector is generated according to the uniform random distribution. In order to accelerate the convergence and optimize the solutions, we take advantage of the opposition-based learning for initializing the population considering stochastic and opposite solutions simultaneously. By utilizing opposite-based learning, we can obtain fitter starting candidate solutions even when there is no a priori knowledge about the solution. The procedure of opposition-based initialization is presented as the following steps.

1. **Initialize the population \( P \) randomly.**
2. **Calculate opposite population \( OP \) by**
   \[
   OX_{ij} = a_j + b_j - X_{i,j},
   \]
   \( i = 1, 2, \ldots, N_p; \quad j = 1, 2, \ldots, D, \)
   where \( X_{i,j} \) and \( OX_{ij} \) denote \( j \)th variable of the \( i \)th vector of the population and the opposite population, respectively.
3. **Select the \( N_p \) fittest individuals from \( \{P, OP\} \) as initial population, where, the selection operation is based on the concept of Pareto dominance. According to the dominance relation between \( X_{ij} \) and \( OX_{ij} \), there may be at most three situations [43]: (1) \( X_{ij} \) dominates \( OX_{ij} \); (2) \( OX_{ij} \) dominates \( X_{ij} \); (3) \( X_{ij} \) and \( OX_{ij} \) are nondominated with each other; select the vector with the less crowding distance.
3.3.2. Mutation. In the traditional differential evolution algorithm, the base vector \( X_{r_i} \) is chosen from three population vectors randomly to achieve the exploratory searching. However, it reduces the convergence of the algorithm. To accelerate the convergence rate and maintain the diversity, a tournament scheme is proposed to choose the base vector. If \( X_{r_i} \) has better fitness function value than the other two solutions. The region around \( X_{r_i} \) is searched with the hope of getting a better solution, so as to make the algorithm converge faster. If there is no dominant relationship between the three solutions, then base vector is selected randomly. Assuming that \( X_{th} \) is the most appropriate solution, the mutation process can be defined as follows:

\[
V_{i, G+1} = X_{th, G} + F ( X_{r_i, G} - X_{r_j, G} ).
\]

(27)

This variation gradually transforms itself into search intensification feature for fast convergence when the points in the population form a cluster around the global minima.

3.3.3. Crossover. The target of the crossover is to generate the trial vector. Crossover operation is to find a better solution and maintain the diversity by exchanging components of the target vector and the mutant vector to diversify the current population by constant updating and exploring. For traditional binomial crossover process, the crossover parameter CR is preset and fixed. The convergence is influenced by this parameter. A dynamical adjustment scheme is proposed to update CR as follows:

\[
CR_G = \frac{1}{D} \left( 1 - \frac{G-1}{G_{\text{max}}} \right),
\]

(28)

where \( D \) denotes the dimensions of the solution, \( G \) denotes the current generation, and \( G_{\text{max}} \) is the max generation. As we can know from (28), at the beginning of evolution process, the parameter \( CR_G \) is in the maximum. We can know from (23) that the components of the child vector come from vector \( V_{i, G+1} \) with a bigger probability. In this phase, the main objective of the crossover operation is mainly to explore so as to improve the diversity of solutions. Then, \( CR_G \) decreases with the generation, and the algorithm will tend to exploitation.

3.3.4. Selection. The careful selection of candidate solutions facilitates the generation of a good Pareto front. In the selection process, we used a similar selection mechanism and the entropy crowding technique proposed in [43] to generate the next generation. The selection mechanism uses two populations for the selection process in each generation. They are the current population with \( N_p \) population vectors and the advanced population with \( N_{p}' \) population vectors. The advanced population is empty at the beginning, and it is used to store the Pareto optimality found so far in the whole evolution process. As the evolution progresses, three rules are used to update the advanced population. After the mutation operation, the trial vectors that are not dominated by the corresponding target vectors obtained at each generation are compared one by one with the current advanced population, and good solutions enter the advanced population. The three update rules are described as follows.

1. If the trial vector is dominated by a member(s) of advanced population, the trial vector is rejected.
2. If the trial vector dominates some member(s) of the advanced population, then the dominated members are deleted from advanced population and the trial vector enters advanced population.
3. If the trial vector does not dominate any members and none of the advanced population member dominates the trial vector, which implies that the trial vector belongs to the Pareto front and it enters the archive.

In order to have a good diversity among generated nondominated solutions in the advanced population of fixed size, the crowding entropy [43] is utilized to evaluate the crowding degree around each nondominated solution. The procedure of the proposed algorithm is described as follows.

Step 1 (initial population operation). Set the generation number \( G = 0 \), randomly initialize the population \( P_G = \{X_{1,G}, X_{2,G}, ..., X_{N_p,G}\} \) with vector \( X_{i,G} = \{x_{i1,G}, x_{i2,G}, ..., x_{iD,G}\} \) \( i = 1, 2, ..., N_p \), uniformly distributed in the parameter space, initialize the advanced population \( A_G = \emptyset \) with the maximum size \( N_{p}' \), and specify the maximum number of generations \( G_{\text{max}} \).

Step 2. Generate the opposite individual \( OX_{i,G} \) of the \( i \)th individual \( X_{i,G} \) according to (26).

Step 3. Select the \( N_p \) fittest individuals from \( \{P, OP\} \) as initial population based on the concept of Pareto dominance as described in Section 3.3.1.

Step 4. While the maximum number of generations \( G_{\text{max}} \) has not been reached, do the next steps for each individual vector.

Step 5 (mutation operation). Calculate CR according to (28) and generate a mutated vector \( V_{i,G+1} \) corresponding to the target vector \( X_{i,G} \) according to (27).

Step 6 (crossover operation). Generate a trial vector \( U_{i,G+1} \) for each target vector \( X_{i,G} \) according to (23).

Step 7 (selection operation). Evaluate the trial vector \( U_{i,G+1} \).

If \( X_{i,G} \) dominates \( U_{i,G+1} \), \( U_{i,G+1} \) is rejected.

If \( U_{i,G+1} \) dominates \( X_{i,G} \), \( X_{i,G+1} = U_{i,G+1} \), and update \( A_G \) with the three update rules.

If \( X_{i,G} \) and \( U_{i,G+1} \) are nondominated with respect to each other, update \( A_G \) with the update rules, and the less crowded one based on the crowding entropy diversity measure tactic will be the next generation target vector \( X_{i,G+1} \).

Step 8. When \( A_G \) exceeds the maximum size, the less crowded vectors keep the size of the advanced population at \( N_{p}' \).

Step 9. Consider \( G = G + 1 \); return to Step 4 until maximum number of generations.
3.4. Computational Complexity Analysis. We consider the complexity of one iteration of the entire algorithm. Basic operations and their worst case complexities are as follows.

1. To select the $N_p$ fittest individuals as the initial population based on the concept of Pareto dominance. The complexity is $O \left( D \times N_p \right)$ for $D \times N_p$ comparisons.

2. To check tournament best of three solutions for mutation operation for one iteration, the complexity is $O \left( 2 \times D \times N_p \right)$.

3. To check the domination status of new solution with target solution for one iteration: $O \left( D \times N_p \right)$.

4. To select of $N_p$ solutions for next generation using nondominated and crowding entropy. The complexity of calculating and sorting the crowding entropy are $O \left( D \times N_p \times \log N_p \right)$ and $O \left( N_p \times \log N_p \right)$.

Therefore, the overall complexity is $O \left( D \times N_p \times \log N_p \right)$. Here, $D$ and $N_p$ represent the number of objective functions and population size, respectively. The traditional differential evolution (DE) algorithm is with the overall complexity $O \left( D \times N^3 \right)$. The nondominated sorting genetic algorithm (NSGA-II) and the opposite-based differential evolution (ODE) algorithm are with the overall complexity $O \left( D \times N^2 \right)$. Compared with them, the proposed algorithm has a low computational complexity. Besides, a tournament is used in the mutation operation to explore the region around the best solution, which may get a better solution. The variation gradually transforms itself into search intensification feature for fast convergence when the points in the population form a cluster around the global minima. Thus, our algorithm can achieve fast convergence.

4. Performance Evaluation

In this section, we present numerical simulations to demonstrate the effectiveness of the proposed fast multiobjective differential evolution algorithm. A cluster is formed by several cognitive sensor nodes to sense the vacant spectrum cooperatively and the cluster head is designed as the fusion center. In order to compute the sensing energy $C_{Si}$ and the transmission energy $C_{Ti}$, the wireless transceiver chip is assumed to be CC2420. The communication protocol is 802.15.4/ZigBee. The largest communication range is 100 m and the highest transmission rate is 250 Kb/s. The sensor nodes lie in a circle with a radius of 70 m and the fusion center is in the center of the circle.

The energy consumption for spectrum sensing mainly includes two parts: channel monitoring to make decision and processing the signal to be sent to fusion center. The first part depends on the spectrum sensing time. We assume the sampling times $T_0 = 5$; the corresponding sensing time is $1 \mu s$. Considering the typical ZigBee circuit power is 40 mW, the channel monitoring energy consumption is 40 nJ. In the signal processing phase, assuming that the voltage is 2.1 V and electric current is 17.4 A, then the energy consumption of the signal processing is 150 nJ/bit. In this paper, the signal processing is about 1 bit; therefore, the spectrum sensing energy consumption is $C_{Si} = 190$ nJ. Because the energy consumption of sending decision-making data is influenced by transmission distance and path loss, we assume $C_{Ti}$ is equal to $C_{Si}$.

In order to demonstrate the performance of the proposed fast multiobjective differential evolution (FMODE) algorithm, we evaluate the performance of the optimization algorithm using convergence and diversity metrics. Comparisons with widely used multiobjective evolutionary algorithms (NSGA-II, DE, and ODE) are shown to verify the efficiency and effectiveness of the proposed algorithm. For all conducted experiments, parameter settings have been chosen according to reported setting in the previous literature [46]. That is, the population size $N_p$ is set at 100, differential amplification factor $F$ is set at 0.5, and the maximum generation $G_{max}$ is set as 250. The crossover probability for NSGA-II, DE, and ODE is set at 0.9. For fair performance comparison, all the four algorithms used the same parameters setting.

4.1. Convergence and Diversity. There are three goals in a multiobjective optimization: (1) convergence to the Pareto optimal set, (2) maintenance of diversity in solutions of the Pareto optimal set, and (3) maximal distribution bound of the Pareto optimal set. In order to evaluate the performance of the proposed fast multiobjective differential evolution algorithm, we define the convergence criteria and diversity criteria.

In convergence metrics, we use the generational distance, which is introduced in [47], to evaluate the convergence criteria. The generational distance is used to measure how far the elements are in the set of nondominated vectors found so far from those in the Pareto optimal set. It is defined as

$$\zeta = \frac{\sum_{i=1}^{M} d_i}{M},$$

where $M$ is the number of vectors in the set of nondominated solutions searched so far and $d_i$ is the Euclidean distance between each of nondominated solutions and the nearest solution in the Pareto optimal set. It is clear that a value $\zeta = 0$ indicates that all the generated elements are in the Pareto front. A smaller value of $\zeta$ demonstrates a better convergence to the Pareto front.

In diversity metric, the diffusion range between the nondominated solutions, this value implies the extent of spread between the Pareto optimal solutions. A smaller value indicates a better distribution and diversity of nondominated solutions. Pareto optimal solution set is uniformly distributed when this value is 0. We define the diversity criteria introduced in [47] as follows:

$$\Delta = \frac{\sum_{i=1}^{k} d \left( E_i, \Omega \right) + \sum_{X \in \Omega} \left| d \left( X, \Omega \right) - \overline{d} \right|}{\sum_{i=1}^{k} d \left( E_i, \Omega \right) + \left| \Omega - k \right| \overline{d}},$$

$$d \left( X, \Omega \right) = \min_{Y \in \Omega \setminus X} \left\| F \left( X \right) - F \left( Y \right) \right\|,$$

$$\overline{d} = \frac{1}{|\Omega|} \sum_{X \in \Omega} d \left( X, \Omega \right),$$
where $\Omega$ denotes the set of solutions and $(E_1, E_2, \ldots, E_k)$ are $k$ extreme solutions in the set of Pareto optimal solution set. 

$k$ is the number of objectives.

In order to know how competitive the proposed approach is, we compare the differential evolution algorithms by convergence and diversity metrics. Figure 3 presents the comparison of the convergence metric. It shows that FMODE obtains better values for the convergence metric than the other three algorithms. This means that the resulting Pareto fronts from FMODE are closer to the optimal Pareto fronts than those computed by NSGA-II, DE, and ODE. This supports that the convergence performance of the FMODE is superior to that of the three other algorithms. ODE obtains a competitive result compared to NSGA-II and DE. The reason is that opposition numbers are used to improve the exploration and convergence performance of the optimization process for both FMODE and ODE. Besides, a tournament scheme is utilized in mutation operation to accelerate the convergence.

Figure 4 illustrates the comparison of diversity metric for different algorithms. It shows that the diversity metric of fast multiobjective differential evolution algorithm is lower than the traditional differential evolution algorithm which indicates a better distribution and diversity of nondominated solutions. NSGA-II obtains competitive results to DE and ODE. That is because a diversity preservation method based on crowding distance is adopted in the selection operation of ODE, and a crowding entropy diversity measure tactic is proposed to preserve the diversity of Pareto optimality in FMODE.

4.2. Energy Consumption. As it is shown from Figure 5, for the spectrum sensing with no optimization, the energy consumption increases rapidly with the number of cognitive sensor nodes. All of the four differential evolution algorithms could obtain more stable energy consumption with sleeping schedule and censoring scheme integrated. And the proposed algorithm can implement lower energy consumption than the other compared algorithms.

4.3. The Global Probabilities of Spectrum Sensing and False Alarm. Figures 6 and 7 present the comparison of the global probabilities of spectrum sensing and false alarm. It shows that in both Figures 6 and 7 the global probabilities of spectrum sensing and false alarm are at a relatively ideal region. With the increase of cognitive sensor nodes, the global probability of spectrum sensing will increase, and the global probability of false alarm will reduce. It is different from the traditional ad Hoc scenario. The reason is that the network may be modeled in terms of clusters in an ad Hoc scenario. Each cluster may have its particular decision process. Using this, the average decision of each cluster may be combined to produce the net total decision in the network. Though the OR rule produces a very high $Q_D$, it has the drawback of having a high $Q_F$. In the presence of a faulty sensor (which always shows $H_1$), the performance is degraded significantly.

For a reasonably large intensity, that is to say, a particular sensor is surrounded by at least 8 closest neighbors.
Such a network can now be tested for effects with the logical OR rule [41]. The parameters are constant including the probability of spectrum sensing and SNR in the sampling time $T_0$ for each sensor node. Based on these parameters, the effects of OR decision fusion with the number of cooperative sensor nodes may be tested. We adopt the centralized decision fusion in the paper. Therefore, the global probability becomes the individual probability of the sensor node in distributed spectrum sensing. Figure 7 shows the effect of the number of cooperative sensor nodes ($N$) on the global probability of false alarm. We assume that the global probability of spectrum sensing which can be obtained from Figure 6 is a constant in the sampling time $T_0$. We can observe a gradual decrease in the $Q_F$ with increase in the number of cooperative sensor nodes. It gets a similar result to [41].

Likewise, we can also come to a conclusion that both the global probability of spectrum sensing and the global probability of false alarm increase slowly with the increase of cognitive sensor nodes. The reason is that the number of sleeping nodes and censoring information increase at the same time. Thus, the sensing performance changes slower. From the results obtained from Figures 6 and 7, it is observed that the global spectrum sensing performance and the global probability of false alarm are better than NSGA-II, DE, and ODE to balance the sensing performance and energy consumption.

5. Conclusion

In this paper, we investigated the problem of how to balance sensing performance and energy consumption. With sleeping schedule and censoring scheme integrated, we formulated the energy-efficient cooperative spectrum sensing as a multiobjective optimization problem to derive the optimum sleeping and censoring probabilities. Based on the traditional differential evolution algorithm, we applied the opposition-based learning to initial the population and introduced a tournament scheme in the mutation operation. In order to converge faster and maintain the diversity, we proposed a control parameters dynamic adjustment scheme and a novel population selection scheme to get Pareto optimum solutions. In addition, the performance evaluation shows that the proposed fast multiobjective differential evolution algorithm can not only reduce the average node energy consumption, but also improve the global probability of spectrum sensing.

In the future, to apply the algorithm to the large-scale wireless cognitive radio sensor network and further prolong the lifetime of the network, we take the actuators into consideration. Using the actuators to collect the sensor nodes’ local decision and sensing data can make this algorithm adapt to large-scale sensor network and further decreases the node’s energy consumption.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors would like to acknowledge that this work was partially supported by the National Natural Science Foundation of China (Grant nos. 61379111, 61202342, 61402538, and 61403424) and Research Fund for the Doctoral Program of Higher Education of China (Grant no. 20110162110042).

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[2] L. M. Oliveira and J. J. Rodrigues, "Wireless sensor networks: a survey on environmental monitoring," *Journal of Communications*, vol. 6, no. 2, pp. 143–151, 2011.
[3] P. Rawat, K. D. Singh, H. Chaouchi, and J. M. Bonnin, “Wireless sensor networks: a survey on recent developments and potential synergies,” The Journal of Supercomputing, vol. 68, no. 1, pp. 1–48, 2013.

[4] D. A. Roberson, C. S. Hood, J. L. LoCicero, and J. T. MacDonald, “Spectral occupancy and interference studies in support of cognitive radio technology deployment,” in Proceedings of the 1st IEEE Workshop on Networking Technologies for Software Defined Radio Networks (SDR ’06), pp. 26–35, September 2006.

[5] B. Wang and K. J. R. Liu, “Advances in cognitive radio networks: a survey,” IEEE Journal on Selected Topics in Signal Processing, vol. 5, no. 1, pp. 5–23, 2011.

[6] Y. Zeng, Y.-C. Liang, A. T. Hoang, and R. Zhang, “A review on spectrum sensing for cognitive radio: challenges and solutions,” EURASIP Journal on Advances in Signal Processing, vol. 2010, Article ID 381465, 2010.

[7] Q. Zhao and B. M. Sadler, “A survey of dynamic spectrum access,” IEEE Signal Processing Magazine, vol. 24, no. 3, pp. 79–89, 2007.

[8] Y.-C. Liang, K.-C. Chen, G. Y. Li, and P. Mähönen, “Cognitive radio networking and communications: an overview,” IEEE Transactions on Vehicular Technology, vol. 60, no. 7, pp. 3386–3407, 2011.

[9] O. B. Akan, O. B. Karli, and O. Ergul, “Cognitive radio sensor networks,” IEEE Network, vol. 23, no. 4, pp. 34–40, 2009.

[10] K. B. Letaief and W. Zhang, “Cooperative communications for cognitive radio networks,” Proceedings of the IEEE, vol. 97, no. 5, pp. 878–893, 2009.

[11] M. Ozger and O. B. Akan, “Event-driven spectrum-aware clustering in cognitive radio sensor networks,” in Proceedings of the 32nd IEEE Conference on Computer Communications (INFOCOM ’13), pp. 1483–1491, Turin, Italy, April 2013.

[12] A. K. Karmokar and A. Anpalagan, “Energy-efficient cross-layer design of dynamic rate and power allocation techniques for cognitive green radio networks,” European Transactions on Telecommunications, vol. 24, no. 7-8, pp. 762–776, 2013.

[13] S. Xie, Y. Liu, Y. Zhang, and R. Yu, “A parallel cooperative spectrum sensing in cognitive radio networks,” IEEE Transactions on Vehicular Technology, vol. 59, no. 8, pp. 4079–4092, 2010.

[14] Z. Li, F. R. Yu, and M. Huang, “A distributed consensus-based cooperative spectrum-sensing scheme in cognitive radios,” IEEE Transactions on Vehicular Technology, vol. 59, no. 1, pp. 383–393, 2010.

[15] G. Ding, J. Wang, Q. Wu, F. Song, and Y. Chen, “Spectrum sensing in opportunity-heterogeneous cognitive sensor networks: how to cooperate,” IEEE Sensors Journal, vol. 13, no. 11, pp. 4247–4255, 2013.

[16] H. Zhang, Z. Zhang, H. Dai, R. Yin, and X. Chen, “Distributed spectrum-aware clustering in cognitive radio sensor networks,” in Proceedings of the 54th Annual IEEE Global Telecommunications Conference: Energizing Global Communications (GLOBECOM ’11), pp. 1–6, Houston, Tex, USA, December 2011.

[17] T. Kieu-Xuan and I. Koo, “A censor-based cooperative spectrum sensing scheme using fuzzy logic for Cognitive Radio Sensor Networks,” IEICE Transactions on Communications, vol. 93, no. 12, pp. 3497–3500, 2010.

[18] K. Yamasaki and T. Ohtsuki, “Design of energy-efficient wireless sensor networks with censoring, on-off, and censoring and on-off sensors based on mutual information,” in Proceedings of the IEEE 61st Vehicular Technology Conference (VTC ’05), pp. 1312–1316, Dallas, Tex, USA, June 2005.

[19] B. V. Babu and M. M. L. Jehan, “Differential evolution for multi-objective optimization,” in Proceedings of the Congress on Evolutionary Computation (CEC ’03), pp. 2696–2703, Canberra, Australia, December 2003.

[20] C. S. Chang and D. Y. Xu, “Differential evolution based tuning of fuzzy automatic train operation for mass rapid transit system,” IEEE Proceedings: Electric Power Applications, vol. 147, no. 3, pp. 206–212, 2000.

[21] C. A. Coello, “Evolutionary multi-objective optimization: a historical view of the field,” IEEE Computational Intelligence Magazine, vol. 1, no. 1, pp. 28–36, 2006.

[22] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: improving the strength pareto evolutionary algorithm for multiobjective optimization,” in Proceedings of the 3rd Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems (EUROGEN ’01), pp. 95–100, Athens, Greece, September 2001.

[23] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182–197, 2002.

[24] A. J. Nebro, E. Alba, and F. Luna, “Multi-objective optimization using grid computing,” Soft Computing, vol. 11, no. 6, pp. 531–540, 2007.

[25] S. Kukkonen and J. Lampinen, “GDE3: the third evolution step of generalized differential evolution,” in Proceedings of the IEEE Congress on Evolutionary Computation, vol. 1, pp. 443–450, Edinburgh, Scotland, September 2005.

[26] R. Storn and K. Price, “Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces,” Journal of Global Optimization, vol. 11, no. 4, pp. 341–359, 1997.

[27] M.-Y. Cheng, D.-H. Tran, and Y.-W. Wu, “Using a fuzzy clustering chaotic-based differential evolution with serial method to solve resource-constrained project scheduling problems,” Automation in Construction, vol. 37, pp. 88–97, 2014.

[28] F. Zheng, A. C. Zecchin, and A. R. Simpson, “Self-adaptive differential evolution algorithm applied to water distribution system optimization,” Journal of Computing in Civil Engineering, vol. 27, no. 2, pp. 148–158, 2013.

[29] S. Das and P. N. Suganthan, “Differential evolution: a survey of the state-of-the-art,” IEEE Transactions on Evolutionary Computation, vol. 15, no. 1, pp. 4–31, 2011.

[30] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, “Multiobjective evolutionary algorithms: a survey of the state of the art,” Swarm and Evolutionary Computation, vol. 1, no. 1, pp. 32–49, 2011.

[31] M. Ali, P. Siarry, and M. Pant, “An efficient differential evolution based algorithm for solving multi-objective optimization problems,” European Journal of Operational Research, vol. 217, no. 2, pp. 404–416, 2012.

[32] S. Maleki, A. Pandharipande, and G. Leus, “Energy-efficient distributed spectrum sensing for cognitive sensor networks,” IEEE Sensors Journal, vol. 11, no. 3, pp. 565–573, 2011.

[33] M. Najimi, A. Ebrahimzadeh, S. M. H. Andargoli, and A. Fallahi, “A novel sensing nodes and decision node selection method for energy efficiency of cooperative spectrum sensing in cognitive sensor networks,” IEEE Sensors Journal, vol. 13, no. 5, pp. 1610–1621, 2013.

[34] S. M. Mishra, A. Sahai, and R. W. Brodersen, “Cooperative sensing among cognitive radios,” in Proceedings of the IEEE.
International Conference on Communications (ICC '06), pp. 1658–1663, July 2006.

[35] C. Rago, P. Willett, and Y. Bar-Shalom, "Censoring sensors: a low-communication-rate scheme for distributed detection," IEEE Transactions on Aerospace and Electronic Systems, vol. 32, no. 2, pp. 554–568, 1996.

[36] S. M. Kay, Fundamentals of Statistical Signal Processing, Volume 2: Detection Theory, Prentice-Hall, Englewood Cliffs, NJ, USA, 1998.

[37] P. K. Varshney, Distributed Detection and Data Fusion, Springer, New York, NY, USA, 1996.

[38] W. Zhang, R. K. Mallik, and K. Ben Letaief, "Optimization of cooperative spectrum sensing with energy detection in cognitive radio networks," IEEE Transactions on Wireless Communications, vol. 8, no. 12, pp. 5761–5766, 2009.

[39] E. C. Y. Peh, Y.-C. Liang, Y. L. Guan, and Y. Zeng, "Optimization of cooperative sensing in cognitive radio networks: a sensing-throughput tradeoff view," IEEE Transactions on Vehicular Technology, vol. 58, no. 9, pp. 5294–5299, 2009.

[40] S. Maleki, S. P. Chepuri, and G. Leus, "Optimization of hard fusion based spectrum sensing for energy-constrained cognitive radio networks," Physical Communication, vol. 9, pp. 193–198, 2013.

[41] A. K. Kattepur, A. T. Hoang, Y. C. Liang, and M. J. Er, "Data and decision fusion for distributed spectrum sensing in cognitive radio networks," in Proceedings of the IEEE 6th International Conference on Information, Communications & Signal Processing, pp. 1–5, Singapore, December 2007.

[42] S. Maleki, S. P. Chepuri, and G. Leus, "Optimal hard fusion strategies for cognitive radio networks," in Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '11), pp. 1926–1931, Cancún, Mexico, March 2011.

[43] Y.-N. Wang, L.-H. Wu, and X.-F. Yuan, "Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure," Soft Computing, vol. 14, no. 3, pp. 193–209, 2010.

[44] M. A. Ahandani and H. Alavi-Rad, "Opposition-based learning in the shuffled differential evolution algorithm," Soft Computing, vol. 16, no. 8, pp. 1303–1337, 2012.

[45] H. R. Tizhoosh, "Opposition-based learning: a new scheme for machine intelligence," in Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation, CIMCA and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (IAWTIC '05), pp. 695–701, Vienna, Austria, November 2005.

[46] R. S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution," IEEE Transactions on Evolutionary Computation, vol. 12, no. 1, pp. 64–79, 2008.

[47] M. Ali, P. Siarry, and M. Pant, "An efficient Differential Evolution based algorithm for solving multi-objective optimization problems," European Journal of Operational Research, vol. 217, no. 2, pp. 404–416, 2012.
Submit your manuscripts at
http://www.hindawi.com