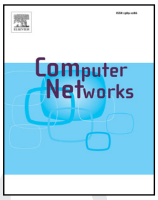
Computer Networks xxx (2016) xxx–xxx

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



Energy efficient power allocation in cognitive radio network using coevolution chaotic particle swarm optimization

Meiqin Tang*, Yalin Xin

Institute of Mathematics and Statistics, Ludong University, Yantai 264025, PR China

A R T I C L E I N F O

Article history:
Received 1 April 2015
Revised 6 February 2016
Accepted 8 February 2016
Available online xxx

Keywords:
Cognitive radio (CR) network
Power allocation
Large scale global optimization
Coevolutionary chaotic particle swarm optimization (CCPSO)

A B S T R A C T

In this paper, the trade-off between utility and energy consumption in orthogonal frequency division multiplexing (OFDM)-based cognitive radio (CR) network is investigated. Energy efficiency problem is very important in the field of CR network, where the utility is maximized and the energy consumption is minimized in such a CR network. Since the trade-off between them has been paying more attentions in literature, this study summarizes the power allocation as an optimization problem that maximizes the energy efficiency via a new energy efficiency metric defined by this paper. The formulated problem is a large-scale nonconvex problem, which is very difficult to solve. In this paper, we present an improved particle swarm optimization (PSO) algorithm to solve the difficult large-scale optimization problem directly. Given the weak convergence of the original PSO around local optima, an improved version that combines the chaos theory is proposed in this study, where chaos theory can help PSO search for solutions around the personal and global bests. In addition, for the purpose of accelerating the convergence process when facing with such a large-scale optimization, the original problem is decomposed into a number of small ones by employing the coevolutionary methodology, and then divide-and-conquer strategy is used to avoid producing infeasible solutions. Simulations demonstrate that the proposed coevolution chaotic PSO needs a smaller number of iterations and can achieve more energy efficiency than the other algorithms.

1. Introduction

Nowadays almost all wireless spectrum has been licensed to existing wireless communications applications. With the increasing demand for wireless data service, spectrum scarcity will become a big problem in future development of wireless communications networks. One promising solution to overcome the spectrum scarcity problem is to use opportunistic spectrum access techniques such as cognitive radio [1], which lets unlicensed users (called secondary users or cognitive users) temporarily utilize a licensed spectrum band, if the licensed users (called primary users) are idle or the interference received at primary users from secondary transmissions is tolerable (in other words, secondary transmissions do not affect the transmission quality of primary users). Due to its potential to largely improve the spectrum utilization efficiency, CR has received much attention from academia, industry, and spectrum regulation agencies [2].

In a cognitive radio (CR) network, primary users have the highest priority to use the spectrum. Secondary users are aware of the transmission environments, and can adapt their transmission/reception patterns to the varying spectrum environments. As an example, consider that a secondary user uses a licensed spectrum band. If the corresponding primary user is back (i.e., the primary user needs to use the spectrum band), the secondary user needs to stop using the spectrum band, and try to find

* Corresponding author. Tel.: +86 15063815887.
E-mail addresses: meiqintang@aliyun.com (M. Tang), xyl_aya@sohu.com (Y. Xin).

http://dx.doi.org/10.1016/j.comnet.2016.02.010
1389-1286/

other idle spectrum bands to continue its wireless access service. Dynamic spectrum allocation is a main challenge in the design of cognitive radio networks, which enables wireless devices to opportunistically access portions of the spectrum as they become available.

In this paper, we consider an orthogonal frequency division multiplexing (OFDM)-based CR network, and we focus on the network utility (to be defined in Section 3). We define "utility per Joule" as the energy efficiency metric, which can effectively characterize the trade-off between utility and energy. A power allocation problem is formulated, which maximizes the energy efficiency. Since there are base stations (BSs) in the system, the optimization problem is centralized and nonconvex, and is hard to be transformed to a convex problem. So in this paper, we adopt PSO (particle swarm optimization) algorithm , which can solve the nonconvex optimization efficiently. PSO algorithms are modern heuristic algorithms based on bird flocking, there is no theory proof for PSO to get the global optimum, but they have demonstrated their potential in solving complicated optimization problems [3–6] and network optimization problems [7–9]. The advantages of PSO algorithms include: they have simple theoretical structure with good convergence properties; they are easy to implement; they do not require the objective functions to be continuous. PSO methods have been popularly used in wireless networks. For example, Zhao et al. [7] uses PSO to optimize CR parameters based on the spectrum environments and user needs; a PSO-based distributed resource allocation algorithm in wireless mesh networks is proposed in Ref. [8]; and Lin [9] applies PSO to deal with router node placement problem in a dynamic wireless mesh network such that the network connectivity and client coverage are maximized.

Q3 However, it is very likely that traditional PSO algorithms may be trapped into local optimal solutions (which are not global optimal). Therefore, in the literature, chaos, which has the features of randomness, ergodicity and regularity, has been used in PSO algorithms recently [10–14]. Chaotic PSO algorithms can maintain the population diversity, which is a nice property. Liu et al. [10] applies chaotic dynamic in PSO algorithms, using the chaotic local searching behavior. Coelho and Herrera [11] considers fuzzy identification, which enhances PSO algorithms with chaotic Zaslavskii map sequences and efficient Gustafson–Kessel clustering. The chaotic PSO algorithm is shown to be effective in building a good TS fuzzy model. In [12], the authors consider prediction of silicon content in hot metal, in which PSO algorithms are enhanced with chaotic under the logistic equation. A binary PSO is used in [13] to predict operon in bacterial genomes, and chaotic sequence is introduced when updating inertia weight. In [14], the authors apply a PSO algorithm to estimate the unknown parameters for a hybrid-forecasting model, in which initial values of unknown constants in particle velocity and position equations are generated by chaotic mapping. Due to the nice features of chaos theory in PSO algorithms, we adopt a chaotic PSO algorithm in this paper.

Besides the chaos, we also apply the cooperative coevolution theory, since cooperative coevolution theory is very suitable for large scale optimization problems. An application of cooperative coevolution theory in PSO can be found in [15], in which PSO position update rule relies on Cauchy and Gaussian distributions. And in our recent paper [16], chaos theory is combined into cooperative coevolving PSO, as chaos theory can help PSO search for solutions around the personal and global bests, thus avoiding being trapped into local optimal points. And a belief space is used to store the experiences for individuals to learn from each other indirectly. In this paper for CR networks, coevolution and chaos theory are all combined with PSO, referred to as CCPSO. But there is no need to set a belief space. Two populations of PSO are included in the CCPSO, and the problem is solved by using max–min approach.

The rest of the paper is organized as follows. Related work is given in Section 2. The system model and problem description are presented in Section 3 The proposed CCPSO algorithm is given in Section 4. Numerical results are provided in Section 5, followed by conclusions in Section 6.

2. Related work

The utility maximization in a multi-cell CR network under a total transmit power constraint is considered in [17]. A cooperative secure resource allocation in CR Networks was considered in [18], since the problem is NP hard, the problem is transformed into a generalized geometric programing model. Since secondary users are usually powered by battery, energy consumption in their wireless transmissions is an important issue. Further, large energy consumption is often due to large transmission power, which actually generates large interference to users in the vicinity and degrades service of those users. Accordingly, in this paper, we consider energy efficient CR networks that employ orthogonal frequency division multiplexing (OFDM) technology [19]. The reason we consider OFDM is that OFDM is very suitable for high speed broadband wireless access due to its immunity to inter-symbol-interference (ISI).

Energy efficiency has been well-investigated in traditional wireless networks [20,21]. Code division multiple access (CDMA) networks are considered in [22], which develops a cross-layer algorithm for energy efficiency. It is proved that the algorithm is Pareto-optimal under certain conditions. Meshkati et al. [23] also considers a CDMA network, which studies the trade-off between energy efficiency and delay. A game theoretical approach is presented to maximize the utility by selecting the transmit power under a delay requirement.

Due to the popularity of CR research, energy efficiency in CR has also received a lot of attention. Buzzi and Saturnino [24] consider a cognitive CDMA wireless network, and presents a game-theoretic algorithm to achieve energy efficiency in a one-shot fashion. Wu and Tsang [25] investigates the sensing and transmission durations of secondary users. A nonconvex optimization problem is formulated to achieve energy-efficient power allocation, which is solved by analyzing three special cases. In [26], the authors develop an energy efficient power control algorithm for OFDM-based CR networks. The objective function is based on the "throughput per Joule" metric. The formulated optimization problem, which is nonconvex, is transformed Q4

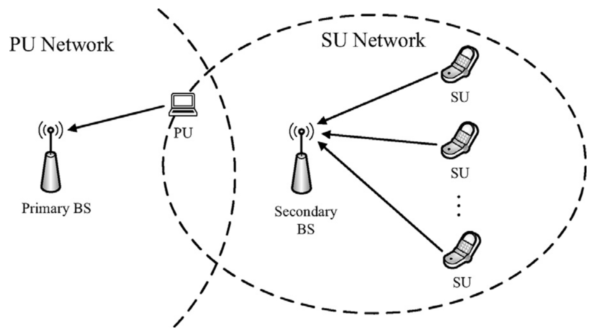


Fig. 1. The network system.

into a convex problem based on parametric programing. Mao et al. [27] considers multiple-input multiple-output cognitive systems and achieves energy efficient spectrum optimization under three constraints (on the total power, the interference power, and the minimum system throughput). The formulated optimization problem, which is also nonconvex, is transformed to a one-dimension problem that has a quasi-concave objective function. Zhong and Wang [28] uses "rate per Joule" as the energy efficiency metric, which stands for the number of information bits that are successfully transmitted per Joule of energy over a normalized bandwidth.

We consider network utility as sigmoid form, which is more suitable for a practical network. Accordingly, we define energy efficiency as "utility per Joule", to characterize the trade-off between the utility and the energy consumption. The formulated optimization problem is nonconvex and has a large scale, which is difficult to solve. Previous optimization works usually try to transform nonconvex problems to convex ones based on some assumptions or try to get suboptimal solutions. In this paper, we develop power allocation algorithms based on an improved CCPSO. We solve the difficult large-scale optimization problem directly, without any assumptions or transforms. In this paper, chaos technique is combined into PSO, whose randomizing and erogeneity characteristics can help avoid being trapped into local optimal points. Coevolution idea is used as well to PSO, to exploit its nice feature in dealing with large scale problems. The proposed algorithm has two populations, and we solve the problem by using max–min approach. Simulation results show that the proposed algorithm can solve the large scale nonconvex power allocation problem effectively and efficiently. The chaotic initialization and search can help the algorithm jump out of local optima and the divide-and-conquer strategy used in the coevolution theory can help the algorithm avoid producing infeasible solutions, the statistical test results reveal that the proposed method outperforms other existing methods and has stronger robustness than other methods.

3. System model and problem description

3.1. System model

As shown in Fig. 1, consider a CR network consisting of N secondary users (SUs) as transmitters and one secondary base station (BS) as the common secondary receiver. Here uplink communication is considered (downlink communication can be treated similarly). The secondary transmitters are allowed to use the licensed spectrum band of a primary transceiver pair: a primary user (PU) as the transmitter and a primary BS as the receiver. OFDM technology is used in both secondary and primary transmissions. To avoid degrading the transmission quality of primary users, it is required that the interference from secondary transmitters to the primary BS is below a threshold denoted I_{th}.

The licensed spectrum band is divided into K subchannels using OFDM [19]. For the wireless links (desired communication links and interference links), block fading model is assumed. In specific, for each wireless link over each subchannel, its link gain is fixed for a time slot, and changes independently in the next time slot. For each link, the link gains over different subchannels are independent from each other. In this paper, we consider power allocation of the secondary transmitters in each time slot.

For a specific time slot, denote P_k^i as the transmit power of SU i over subcarrier $k(k = 1, 2, \ldots, K)$, and $G_k^{ii}(i = 1, 2, \ldots, N)$ as the link gain from SU i to the secondary BS. Then the signal to interference ratio (SINR) for SU i's communication over subchannel k, denoted γ_k^i, is given as

$$\gamma_k^i = \frac{G_k^{ii} P_k^i}{\sum_{j \neq i} G_k^{jj} P_k^j + \sigma^2}, \tag{1}$$

where σ is the background noise.

3.2. Utility function

For SU i's communication over subchannel k, next we define a utility function denoted as U_k^i. The utility function is expected to reflect the satisfaction level of the service quality. In this paper, we adopt the sigmoidal form utility introduced in [30], which means that the user is more and more satisfied with the service as the quality improves. The sigmoidal function can capture the value of the service to the user quite naturally, and be defined as

$$U_k^i = \frac{1}{1 + e^{-a_i(\gamma_k^i - b_i)}} \tag{2}$$

where a_i is slope parameter (a large a_i means that the application has a soft quality-of-service requirement) and b_i is a shift parameter that is actually the required average data rate of the application.

Targeting at higher utility function with less energy consumption, we define an energy efficiency metric as the ratio of the utility to the transmission power, given as

$$E = \frac{\sum_{i=1}^{N} U_k^i}{\sum_{i=1}^{N} P_k^i + P_c} \tag{3}$$

where P_c is static power consumption other than wireless transmissions (for example, circuit power consumption). In other words, the energy efficiency metric is actually "utility per Joule".

3.3. Constraints

Recall that over each subchannel, the interference received by the primary BS from all secondary transmissions should be bounded by a threshold I_{th}. Denote $G_k^{0i}(k =$

$1,\ldots,K$) as the link gain from the SU i to the primary BS. Thus, the interference power constraint is given as

$$\sum_{i=1}^{N} G_k^{0i} P_k^i \le I_{th}. \tag{4}$$

It is also required that the total transmission power over each subchannel is bounded by P_{max}. So we also have a total transmit power constraint, given as

$$\sum_{i=1}^{N} P_k^i \le P_{max}. \tag{5}$$

3.4. Problem description

In this work, we consider the maximal energy efficiency, subject to the interference power constraint and the total transmit power constraint. An optimization problem can be formulated as follows:

$$\begin{aligned} &\text{maximize } E = \frac{\sum_{i=1}^{N} U_k^i}{\sum_{i=1}^{N} P_k^i + P_c} \\ &\text{subject to} \sum_{i=1}^{N} G_k^{0i} P_k^i \le I_{th} \\ &\qquad\qquad \sum_{i=1}^{N} P_k^i \le P_{max}, \\ &\qquad\qquad P_k^i \ge 0. \end{aligned} \tag{6}$$

For the objective function of problem (6), the following theorem is in order.

Theorem 1. *The objective function* $E = \frac{\sum_{i=1}^{N} U_k^i}{\sum_{i=1}^{N} P_k^i + P_c}$, *where* U_k^i *is given in (2), is a nonconvex function of* P_k^i.

Proof. See Appendix. □

Theorem 1 indicates that problem (6) is nonconvex. Since it is hard to transform the problem to a convex problem, we resort to PSO methods to solve it.

4. The proposed optimization algorithm

4.1. Standard particle swarm optimization

Similar to genetic algorithm (GA) [31,32] and differential evolution (DE) [33], PSO is a population-based optimization method which was first proposed by Kennedy and Eberhart [29]. The system is initialized with a population of random solutions and it searches for optima by updating generations. There are two learning processes in the generation of PSO: cognitive learning process based on individuals history, and social learning process based on a swarm's history accumulated by sharing information among all the particles in the swarm. Particles fly around in multidimensional search space.

During flight, each particle adjusts its position according to its own experience and the experience of neighboring particles, making use of the best position encountered by itself and its neighbors. The direction of movement of a particle is defined by the set of particles in the target particle's vicinity and the target particle's history experience. Each particle keeps track of its coordinates in the problem space, which are associated with the best solution achieved so far, denoted as *pbest*. Another best value tracked by the global version of the optimizer is the overall best value, and its location, obtained so far by any particle in the population, denoted as *gbest*. At each time step, the particle swarm optimization consists of velocity changes of each particle toward *pbest* and *gbest* locations. Acceleration is weighted by a random term, which separates random numbers being generated for acceleration toward *pbest* and *gbest* locations.

The ath particle's coordinates (position) is denoted as $X_a = (x_{a1}, \ldots, x_{ad}), d = 1, 2, \ldots, D,$ where D is the dimension of the optimal solution, and $V_a = (v_{a1}, \ldots, v_{ad})$ denotes the corresponding flight speed (velocity). Let $pbest_a = (x_{a1}^{pbest}, \ldots, x_{ad}^{pbest})$ and $gbest = (x_1^{gbest}, \ldots, x_d^{gbest})$ be the best position of individual a and its neighbors' best position so far, respectively. Using the information, the updated velocity of individual a is modified under the following equation in PSO:

$$V_a^{t+1} = \omega V_a^t + c_1 (pbest_a^t - X_a^t) + c_2 (gbest^t - X_a^t) \tag{7}$$

$$X_a^{t+1} = X_a^t + V_a^{t+1} \tag{8}$$

where t is the iteration number, c_1 and c_2 are constants, which represent the weighting of the stochastic acceleration terms that pull each particle toward *pbest* and *gbest* positions, and ω is the inertia weight parameter.

PSO is very efficient in solving complex optimization problems. But it is easy to fall into local optimal solutions. Then the inertia weight parameter is adjusted and chaos theory is combined to PSO.

4.2. Adaptive inertia weight factor(AIWF)

It is clear that Eq. (7) represents the influence of previous velocity, which provides the necessary momentum for particles to roam across the search space. The inertia weight ω is the modulus that controls the impact of previous velocity on the current one. So, the balance between exploration and exploitation in PSO is dictated by the value of ω. Shi and Eberhart in [34,35] made a significant improvement in the performance of the PSO with a linearly varying inertia weight over the generations. ω varies adaptively in response to the objective values of the particles. In particular, AIWF is determined as follows.

$$\omega = \begin{cases} \omega_{min} + \frac{(\omega_{max} - \omega_{min})(f - f_{min})}{f_{avg} - f_{min}} & f \le f_{avg}, \\ \omega_{max} & otherwise, \end{cases} \tag{9}$$

where ω_{max} and ω_{min} denote the maximum and minimum of ω, respectively, f is the current objective value of the particle, and f_{avg} and f_{min} are the average and minimum objective values of all particles, respectively. Under the guidance of Eq. (9), ω varies depending on the objective value of the particle so that particles with low objective values can be protected. AIWF provides a good way to maintain population diversity and to sustain fast convergence.

4.3. Chaos

Chaos theory demonstrates sensitive dependence on initial conditions and also includes infinite unstable periodic motions. Due to non-repetitive nature of chaos, it can carry out overall searches at higher speeds than stochastic esodic searches. The combination of optimization methods and chaotic systems is an important issue in non-linear science. Here, logistic equation is employed to obtain chaos queues, denoted as $z_1, z_2, \ldots$, as follows:

$$z_{n+1} = \mu z_n(1 - z_n), n = 0, 1, 2, \ldots \tag{10}$$

in which $0 \le z_0 \le 1$, and μ is the control parameter. When $\mu = 4$, the system of (9) has been proved to be entirely chaotic.

The essential procedure of chaotic particle swarm optimization (CPSO) is as follows:

Step 1: Chaos initialization for particle.

Step 2: Evaluate the fitness function of each particle.

Step 3: Update each particle's velocity and position.

Step 4: Optimize the global best value by chaos search.

Step 5: If the stopping criteria satisfied, then output the optimum solution, otherwise, loop to Step 2.

4.4. The large scale optimal algorithm using coevolution chaotic particle swarm optimization (CCPSO)

Since the large scale and nonconvex nature makes our formulated problem very complex, cooperative coevolution is applied, which has been proposed as a promising framework for tackling large scale optimization problems [36,37]. It can be regarded as an automatic approach to implement the divide-and-conquer strategy. The detailed procedure of the proposed coevolution chaotic particle swarm optimization (CCPSO) is as follows.

Particles' positions within the population in the CCPSO represent the candidate solutions for solving the control problem. That's to say, the procedure of particle searching for the best position is to search for the power of problem (6). The CCPSO runs at the secondary base station in the SU network, which is a centralized one. The secondary base station collects the necessary powers' information from the secondary uses and updates the memory to perform the calculation, and then broadcasts the solution to the secondary users.

The objective is to maximize aggregate source net utility per power consumption subject to the constraints in CR networks. Constraints are handled based on the penalty functions in the search space [38].

$$L(P_k^i, \mu_i, \kappa_i, \nu_i) = E + \mu_i(P_{max} - P_k^i) + \kappa_i\left(P_k^{th} - \sum_{i=1}^{N} G_k^{0i} P_k^i\right) + \nu_i(\gamma_k^{itar} - \gamma_k^i) + G(P_k^i) \tag{11}$$

where μ_i, κ_i and γ_i are the Lagrange multipliers for the constraints. Since the optimization problem is nonconvex, $G(P_k^i)$ is the penalty term added to the Lagrangian, which can assure the max–min problem and the original problem to be equal [38].

The optimization problem in CR is now in the form of augmented lagrangian. Therefore, the problem is solved by using max–min approach. Two populations of PSO are included in the CCPSO. In the first PSO, the variable is P_k^i, and μ_i, κ_i and γ_i are set to be constant. The fitness defines how well the position vector of each particle satisfies the requirements of the optimization problem. The fitness function for P_k^i is represented as

$$F_1(P_k^i) = max(L(P_k^i, \mu_i, \kappa_i, \gamma_i)). \tag{12}$$

And in the second PSO, P_k^i is set to be constant, while μ_i, κ_i and γ_i are all set to be variables. The fitness function for μ_i, κ_i and γ_i is given as

$$F_2(\mu_i, \kappa_i, \gamma_i) = min(L(P_k^i, \mu_i, \kappa_i, \gamma_i)). \tag{13}$$

The cooperation among particles is established through the "history" *pbest$_a$* and *gbest*, which are updated if better fitness is obtained. The general procedure of CCPSO algorithm can now be described by the following pseudocode:

Procedure of CCPSO

1: Initialization of two PSOs
2: Run the first PSO for generation 1
3: Re-evaluate the *pbest$_a$* values for the second PSO if it is not in the first cycle
4: Run the second PSO for generation 2
5: Re-evaluate the *pbest$_a$* values for the first PSO
6: If the termination condition is not met, go to Step 2;
7: Otherwise, output *gbest*.
8: End procedure

And the detailed improved PSO procedure is as follows.

Step 1: Chaos initialization is adopted to locate the positions of particles and to increase the diversity of the population and the ergodicity in the course of searching without changing the randomicity of algorithm when initializing the particles. Some initial population with good performances is chosen from the initial group with a large number of population. Initialize a vector $z_1, z_2, \ldots$ each component of which is set as a random value in the range [0, 1], and generate chaos queues $z_1, z_2, \ldots, z_a$ by the iteration of logistic Eq. (10). Transfer the chaos queues into the range of the parameters according to following equation:

$$x_a^t = s_a + (t_a - s_a) z_a^t \tag{14}$$

where $[s_a, t_a]$ is the value range of each particle.

Step 2: Compute the fitness values of the vectors and choose the best M solutions as the initial solutions of M particles. Randomly initialize the velocity of M particles.

Step 3: Using the global best and the individual best of each particle, each particle's velocity and position are updated according to Eq. (7).

Step 4: Evaluate the fitness of each particle and compare the evaluated fitness value of each particle to its individual best *pbest$_a$*. If *pbest$_a$* is better than current value, update *pbest$_a$* as current position.

Step 5: If current value of the fitness function is better than the global best *gbest*, update *gbest* as the current position.

Step 6: Optimize *gbest* by chaos search. Firstly, scale *gbest* into [0,1] according to

$$x_a = (Gbest - s_a)/(t_a - s_a) \tag{15}$$

Please cite this article as: M. Tang, Y. Xin, Energy efficient power allocation in cognitive radio network using coevolution chaotic particle swarm optimization, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.02.010

Table 1
Parameters of PSO.

Algorithm	Parameter settings	References
SA	$T_0 = 500°C,\ C = 0.95,\ \Gamma = 9$	[41]
GA	$CRA = 0.55,\ MR = 0.005,\ CO = 0.35,\ MO = 0.25$	[32]
DE	$F = 0.5,\ CR = 0.9$	[33]
PSO	$\omega = 0.57,\ c_1 = c_2 = 2, v_{max} = 4$	[29]
CPSO	$\omega = 0.9,\ c_1 = c_2 = 2, v_{max} = 4$	[39]
CCPSO	ω: [0.4, 0.9], $c_1 = c_2 = 2, v_{max} = 4$	–

and generate chaos queues x_a^t by iteration of logistic equation, then transfer the chaos queues into the optimization variable $gbest^t$ according to the following equation

$$gbest^t = s_a + (t_a - s_a)x_a^t \tag{16}$$

based on which the solution set is obtained: $gbest^t$. Compute the fitness value of each feasible solution $gbest^t$ in the problem space during chaotic search, and get the best solution P_k^{i*}.

Step 7: When the constraints are violated, we pay an extra charge proportional to the amount of violation with the penalty value. And when the maximum iteration is reached, then stop, we can get the global optimum P_k^{i*} that are the solutions of the power allocation problem. Otherwise, loop to Step 3.

5. Numerical results

In this section, we present numerical results for the proposed power allocation algorithm in a network as shown in Fig. 1. The purpose of the simulations is to show that CCPSO can solve the large scale nonconvex optimization problem. The total number of secondary users N is set to 200. Without loss of generality, 100 tested users are randomly selected for the simulations by 20 times. Accordingly, the numerical results are averaging results, and the number of the decision variables for the optimization problem is 100. Packet size is 1000 bytes and nodes are equipped with a single transmitter/receiver, which has a radio range of 500 m. An area of size 1000 m × 1000 m is considered. The maximum total transmit power is set as $P_{max} = 800$ mW. The background noise is set as $\sigma^2 = 5 \times 10^{-13}$ dB. No forward error correction is considered. The link gains follow the path loss model: $G_{ii} = K/d_{ii}^{\varrho}$, and $G_{0i} = K/d_{0i}^{\varrho}$, where d_{ii} and d_{0i} are the distance from SU i to the secondary BS and primary BS, respectively, ϱ is a path loss exponent and set as 4, and $K = 0.097$. The performance of the proposed algorithm is compared with simulated annealing (SA) [40,41], genetic algorithms (GA) [32], differential evolution (DE) [33], the standard PSO [29] and CPSO [39]. All empirical experiments are conducted with a population of 50, except for SA. The parameters of all the methods are all selected optimally in the simulations in Table 1 as selected in corresponding reference For SA method, T_0, C and Γ are the initial temperature, cooling rate and stopping parameter, respectively. For GA method, CRA, MR, CO and MO are the crossover rate, mutation rate, crossover operator, and mutation operator, respectively. For DE method, F is the weighting factor and CR is the crossover constant. Simulations were implemented on a PC with Intel $Core^{TM}$ i7-5820 CPU and memory capacity of 16G(8G*2) running Matlab 7.12.0.

5.1. Performance comparisons

5.1.1. Convergence comparisons

In this example, the performance of the proposed algorithm is compared with those of CPSO and PSO. The utility function is considered as $U_k^i = \frac{1}{1+e^{-a_i(\gamma_k^i - b_i)}}$, which is more reasonable for real networks. In this case, the maximum number of iterations of the algorithms is set to be 1000. N is set to be 100, so the energy consumption is fixed with the fixed number of N. Fig. 2 shows the average convergence of the best individuals of each iteration for the system with different methods, where the system parameters a_i is varied and b_i is fixed. In this case from the figure we can find that the proposed algorithm based on CCPSO converges less than 100 iterations. Compared to CPSO proposed in [39], which just uses the logistic mapping, the proposed algorithm in this paper can get better performance. It is because of the coevolutionary theory, which is more suitable for the large-scale-global characteristic of the problem. And the system is more stable when the parameter a_i is large with fixed b_i. Fig. 3 shows the convergence of the system with fixed a_i and varying b_i, and in this case a large b_i corresponds to a high total utility. As seen from Figs. 2 and 3, we can see that the parameters a_i and b_i can be used to tune the steepness and the center of the utility, respectively. And the proposed approach based on CCPSO shows better performance in the process of the algorithm and has faster convergence speed than other approaches.

In order to verify the trade-off of utility vs energy. In this case, the energy efficiency with the increase of the value of N is given in Fig. 4. The utility function is considered as $U_k^i = \frac{1}{1+e^{-a_i(\gamma_k^i - b_i)}}$ with $a = 5$, $b = 20$. It is observed that energy efficiency initially increases due to the node number increases. This is because that the utility of the system increases relatively higher than the consumption of power increases. It indicates a trade-off between system utility and total energy consumption. From the figure, we can find that N is found to be 80 that identifies the minimum energy consumption and the maximum utility function.

5.1.2. Performance comparisons with different methods

Tables 2–5 list the best, worst, mean value of the optimal energy efficiency solved by GA, SA, DE, PSO, CPSO and CCPSO in the 1000 runs with different a and b with

Please cite this article as: M. Tang, Y. Xin, Energy efficient power allocation in cognitive radio network using coevolution chaotic particle swarm optimization, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.02.010

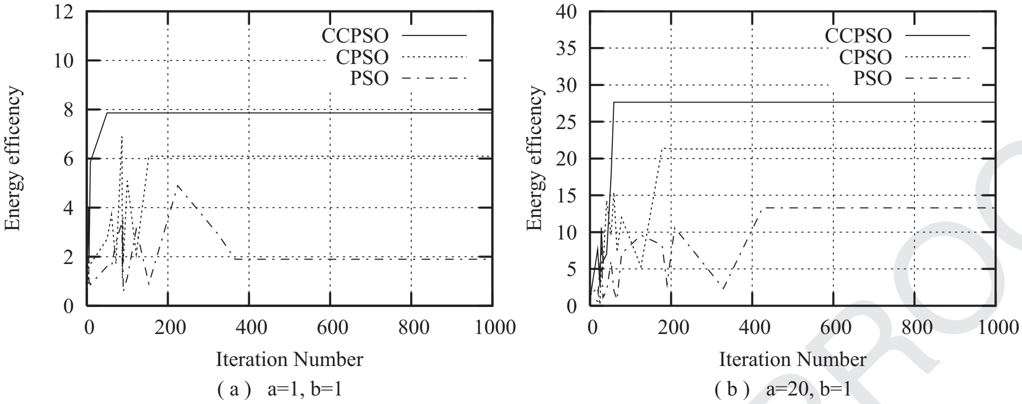


Fig. 2. Energy efficiency with varying a and fixed b when (a) $a = 1$, $b = 1$ (b) $a = 20$, $b = 1$.

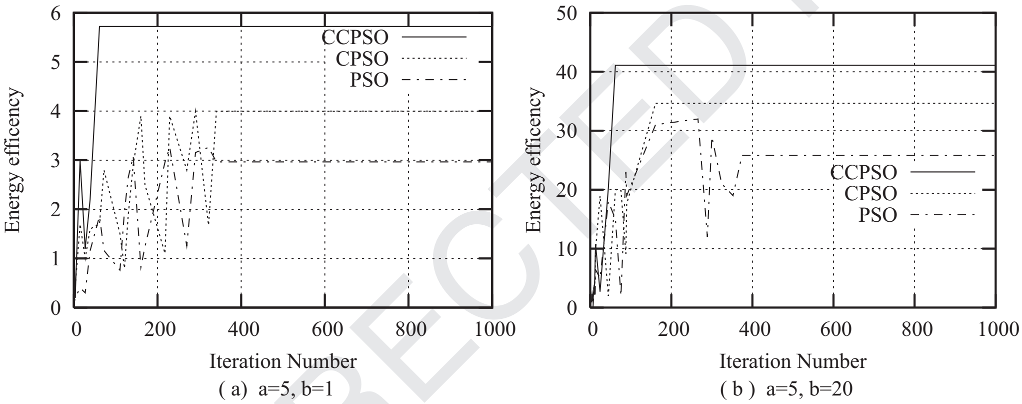


Fig. 3. Energy efficiency with varying b and fixed a when (a) $a = 5$, $b = 1$ (b) $a = 5$, $b = 20$.

Table 2
Energy efficiency when $a = 1, b = 1$ with fixed b.

Algorithms	Best	Worst	Average	SD	Average CPU time (s)
SA	1.9699	1.1084	1.502	0.1694	10.04
GA	2.1014	1.209	1.6568	0.1911	10.69
DE	5.6148	4.8811	5.0426	0.0694	11.46
PSO	2.3611	1.3408	1.8961	0.1999	7.24
CPSO	7.045	7.4876	7.1092	0.0102	10.36
CCPSO	7.8652	7.8652	7.8652	0	12.47

Table 3
Energy efficiency when $a = 20, b = 1$ with fixed b.

Algorithms	Best	Worst	Average	SD	Average CPU time (s)
SA	12.3072	9.7767	10.8538	0.4630	22.32
GA	16.4327	10.6643	11.9427	0.5495	23.65
DE	22.3606	19.3033	21.0428	0.5665	26.74
PSO	17.5826	11.9132	13.3112	0.6009	16.53
CPSO	24.5542	22.323	23.394	0.4604	22.46
CCPSO	27.6501	27.3951	27.4776	0.0355	29.38

$N = 100$. In order to verify the statistical performance of the proposed algorithm, we also give the standard deviation (SD) in the tables. It can be seen that the CCPSO algorithm can provide better “Best”, “Worst”,“Mean”, “SD” and ”Average CPU time” results for the test functions. SD means the volatility of the data, from the tables we can see that some SD of the proposed algorithm are 0, which shows that the proposed algorithm is more stable. Because the coevolution theory solve the proposed large and complex problem using a divide-and-conquer strategy, which avoids producing infeasible solutions, so these statistical test results reveal that the proposed method outperforms other

Table 4
Energy efficiency when $a = 5, b = 1$ with fixed a.

Algorithms	Best	Worst	Average	SD	Average CPU time (s)
SA	2.5152	1.9473	2.3087	0.0888	6.34
GA	3.1082	2.3675	2.6652	0.1280	7.02
DE	4.1462	4.0615	4.1028	0.0178	8.64
PSO	3.6724	2.5032	2.9659	0.1989	5.19
CPSO	4.654	4.112	4.2978	0.2821	6.89
CCPSO	5.7209	5.7143	5.7189	0.0009	9.65

Table 5
Energy efficiency when $a = 5, b = 20$ with fixed a.

Algorithms	Best	Worst	Average	SD	Average CPU time (s)
SA	18.6461	10.9909	13.2166	0.9567	23.98
GA	23.6888	14.0908	18.6109	1.9430	24.66
DE	32.0232	28.7112	31.0655	0.4117	33.04
PSO	29.1253	17.2218	22.2320	1.0890	22.07
CPSO	39.099	37.20878	38.627	0.4694	24.47
CCPSO	41.0927	41.0927	41.0927	0	35.68

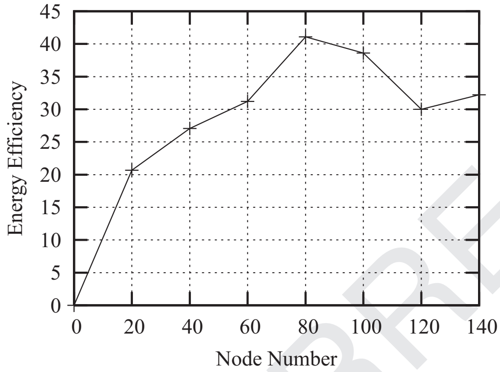


Fig. 4. Energy efficiency with the increase of node number.

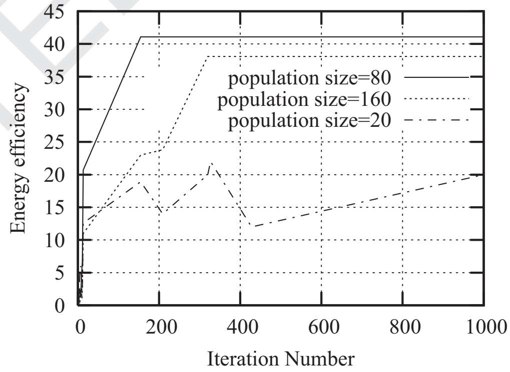


Fig. 5. Effect of population size.

existing methods and has stronger robustness than other methods. Since the proposed algorithm is a dual-swarm one, the convergence time of the proposed algorithm is more than the other algorithms, but the results are comparable. The chaotic initialization and search have strong ability to jump out of local optima, which can help the algorithm reduce the search time.

5.2. Influence of population size

The population size is an important factor which influences the performance of the stochastic search algorithm. Too small population may not be able to reach the maximum value and achieve an optimum, while too large population makes the proposed algorithm slow and computationally inefficient. Tests are carried out for population of 20, 80, and 160. We consider the utility function in the case of $a = 5, b = 20$. As shown from Fig. 5, we can see that the convergence to the optimum is hardly achieved for the proposed optimization algorithm when the population is set as 20, and the convergence speed is slower when the population is set as 160. The size 80 is found to be optimal.

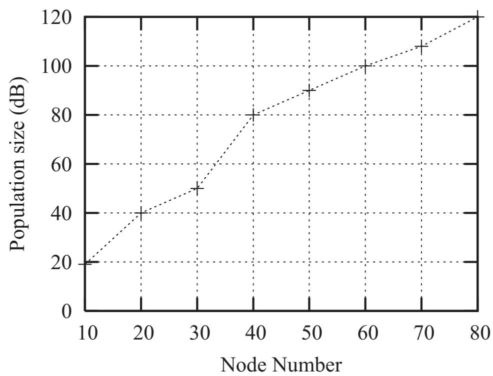


Fig. 6. The relationship between the population size and the number of nodes.

And the relationship between the number of nodes and particles is studied in this case. As shown from Fig. 6, when the number of nodes varies from 10 to 100, the number of particles needs to increase from 40 to 120. This is because when the network is in a large scale,

Please cite this article as: M. Tang, Y. Xin, Energy efficient power allocation in cognitive radio network using coevolution chaotic particle swarm optimization, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.02.010

the particles need to be vectors with higher dimensions, which leads to a larger search space.

6. Conclusion

In this paper, a new metric that reflects the trade-off between the utility and energy consumption is defined in CR networks. Since the optimization problem is a large scale and nonconvex one, our proposed algorithm exploits the coevolutionary and chaotic ideas for the dynamic power allocation problem in CR networks. Strict assumptions such as continuity, differentiability, and convexity of the objective function are not necessary. The formu-

$$\frac{\partial E}{\partial P_k^i} = \frac{\frac{\partial U_k^i}{\partial P_k^i}\left(P_k^i + \frac{P_c}{N}\right) - U_k^i}{\left(P_k^i + \frac{P_c}{N}\right)^2} \tag{18}$$

$$= \frac{\frac{\partial U_k^i}{\partial Q}\frac{\partial Q}{\partial \gamma_k^i}\frac{\partial \gamma_k^i}{\partial P_k^i}\left(P_k^i + \frac{P_c}{N}\right) - U_k^i}{\left(P_k^i + \frac{P_c}{N}\right)^2} \tag{19}$$

$$= \frac{1}{(P_k^i + \frac{P_c}{N})^2}\left[\frac{a_iQ}{(1+Q)^2}\frac{G_k^{ii}}{\Sigma_{j\neq i}G_k^{ij}P_k^i + \sigma^2}\left(P_k^i + \frac{P_c}{N}\right) - \frac{1}{1+Q}\right] \tag{20}$$

$$\frac{\partial^2 E}{(\partial P_k^i)^2} = \frac{\partial}{\partial P_k^i}\left(\frac{\partial E}{\partial P_k^i}\right) \tag{21}$$

$$= \frac{\left(\frac{\partial U_k^i}{\partial P_k^i}\left(P_k^i + \frac{P_c}{N}\right) - U_k^i\right)\left(P_k^i + \frac{P_c}{N}\right)^2 - 2\left(\frac{\partial U_k^i}{\partial P_k^i}\left(P_k^i + \frac{P_c}{N}\right) - U_k^i\right)\left(P_k^i + \frac{P_c}{N}\right)}{\left(P_k^i + \frac{P_c}{N}\right)^4} \tag{22}$$

$$= \frac{\left(\frac{\partial U_k^i}{\partial P_k^i}\right)\left(P_k^i + \frac{P_c}{N}\right)^3 - 2\left(P_k^i + \frac{P_c}{N}\right)\left[\frac{a_iQ}{(1+Q)^2}\frac{G_k^{ii}}{\Sigma_{j\neq i}G_k^{ij}P_k^i + \sigma^2}\left(P_k^i + \frac{P_c}{N}\right) - \frac{1}{1+Q}\right]}{\left(P_k^i + \frac{P_c}{N}\right)^4} \tag{23}$$

$$= \frac{\left(\frac{a_iQ}{(1+Q)^2}\frac{G_k^{ii}}{\Sigma_{j\neq i}G_k^{ij}P_k^i + \sigma^2}\right)\left(P_k^i + \frac{P_c}{N}\right)^3 - 2\left(P_k^i + \frac{P_c}{N}\right)\left[\frac{a_iQ}{(1+Q)^2}\frac{G_k^{ii}}{\Sigma_{j\neq i}G_k^{ij}P_k^i + \sigma^2}\left(P_k^i + \frac{P_c}{N}\right) - \frac{1}{1+Q}\right]}{\left(P_k^i + \frac{P_c}{N}\right)^4} \tag{24}$$

$$= \frac{\frac{a_i^2Q(Q-1)}{(1+Q)^3}\left(\frac{G_k^{ii}}{\Sigma_{j\neq i}G_k^{ij}P_k^j + \sigma^2}\right)^2\left(P_k^i + \frac{P_c}{N}\right)^3 - 2\left(P_k^i + \frac{P_c}{N}\right)\left[\frac{a_iQ}{(1+Q)^2}\frac{G_k^{ii}}{\Sigma_{j\neq i}G_k^{ij}P_k^j + \sigma^2}\left(P_k^i + \frac{P_c}{N}\right) - \frac{1}{1+Q}\right]}{(P_k^i + \frac{P_c}{N}))^4} \tag{25}$$

$$= \frac{\frac{a_i^2Q(Q-1)}{(1+Q)^3}\left(\frac{G_k^{ii}}{\Sigma_{j\neq i}G_k^{ij}P_k^j + \sigma^2}\right)^2(P_k^i + \frac{P_c}{N})^2 - \frac{2a_iQ}{(1+Q)^2}\frac{G_k^{ii}}{\Sigma_{j\neq i}G_k^{ij}P_k^j + \sigma^2}(P_k^i + \frac{P_c}{N}) + \frac{2}{1+Q}}{\left(P_k^i + \frac{P_c}{N}\right)^3} \tag{26}$$

lated optimization problem is solved by using max–min approach, where two populations of PSO are included. The performance of the proposed algorithm is compared with those of related methods in the literature. It is observed that the proposed algorithm is indeed capable of quickly achieving energy-efficient solutions. Future research topics may include dynamic PSO algorithms for cooperative CR networks in which one SU may help relay other SUs' signal to the secondary BS such that cooperative diversity can be achieved. Then each SU may need to distribute its power budget in transmitting its own signal and in relaying other SUs' signals. The problem is much more complex, and deserves further investigation.

Appendix. Proof of Theorem 1

Proof. We use proof by contradiction. Suppose that $E = \frac{\sum_{i=1}^N U_k^i}{\sum_{i=1}^N P_k^i + P_c}$ is a convex function of P_k^i, then

$$\frac{\partial^2 E}{(\partial P_k^i)^2} \leq 0 \tag{17}$$

and Eq. (17) is a necessary condition. For convenience, we set $e^{-a_i(\gamma_k^i - b_i)}$ as Q, thus:

$$\frac{a_i^2Q(Q-1)}{(1+Q)^3}\left(\frac{G_k^{ii}}{\Sigma_{j\neq i}G_k^{ij}P_k^j + \sigma^2}\right)^2\left(P_k^i + \frac{P_c}{N}\right)^2 - \frac{2a_iQ}{(1+Q)^2}\frac{G_k^{ii}}{\Sigma_{j\neq i}G_k^{ij}P_k^j + \sigma^2}\left(P_k^i + \frac{P_c}{N}\right) + \frac{2}{1+Q} = 0 \tag{27}$$

We set

$$\frac{a_i^2Q(Q-1)}{(1+Q)^3}\left(\frac{G_k^{ii}}{\Sigma_{j\neq i}G_k^{ij}P_k^j + \sigma^2}\right)^2\left(P_k^i + \frac{P_c}{N}\right)^2 = a \tag{28}$$

$$\frac{2a_iQ}{(1+Q)^2}\frac{G_k^{ii}}{\Sigma_{j\neq i}G_k^{ij}P_k^j + \sigma^2}\left(P_k^i + \frac{P_c}{N}\right) = b \tag{29}$$

$$\frac{2}{1+Q} = c \tag{30}$$

Then

$$b^2 - 4ac = \left(\frac{2a_iQ}{(1+Q)^2}\frac{G_k^{ii}}{\Sigma_{j\neq i}G_k^{ij}P_k^j + \sigma^2}\right)^2 - \frac{8}{1+Q}\frac{a_i^2Q(Q-1)}{(1+Q)^3}]\left(\frac{G_k^{ii}}{\Sigma_{j\neq i}G_k^{ij}P_k^j + \sigma^2}\right)^2 \tag{31}$$

Please cite this article as: M. Tang, Y. Xin, Energy efficient power allocation in cognitive radio network using coevolution chaotic particle swarm optimization, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.02.010

$$= \left[\left(\frac{2a_i Q}{(1+Q)^2} \right)^2 - \frac{8}{1+Q} \frac{a_i^2 Q(Q-1)}{(1+Q)^3} \right] \left(\frac{G_k^{ii}}{\Sigma_{j \neq i} G_k^{ij} P_k^j + \sigma^2} \right)^2 \tag{32}$$

$$= \frac{4a_i^2 Q(2-Q)}{(1+Q)^4} \left(\frac{G_k^{ii}}{\Sigma_{j \neq i} G_k^{ij} P_k^j + \sigma^2} \right)^2 \tag{33}$$

When $Q > 2$, $b^2 - 4ac < 0$; when $Q < 2$, $b^2 - 4ac > 0$, which cannot assurance that $\frac{\partial^2 E}{(\partial P_k^i)^2} \leq 0$. So E is a nonconvex function. □

References

[1] J. Mitola, G.Q. Maguire, Cognitive radio: Making software radios more personal, IEEE Pers. Commun. 6 (6) (1999) 13–18.
[2] Facilitating Opportunities for Flexible, Efficient, and Reliable Spectrum Use Employing Cognitive Radio Technologies, Technical Report (ET docket no. 03–108), Federal Communications Commission, Washington, DC, 2003. http://hraunfoss.fcc.gov/edocspublic/attachmatch/FCC-03-322A1.pdf.
[3] J.C. Hung, Adaptive fuzzy-GARCH model applied to forecasting the volatility of stock markets using particle swarm optimization, Inf. Sci. 181 (2011) 4673–4683.
[4] J. Cai, W.D. Pan, On fast and accurate block-based motion estimation algorithms using particle swarm optimization, Inf. Sci. 197 (2012) 53–64.
[5] L. Huang, Y.Q. Zhou, J.L. Hu, X. Han, J.L. Shi, Coverage optimization for femtocell clusters using modified particle swarm optimization, in: Proceeding of the 2012 IEEE International Conference on Communications, 2012, pp. 611–615.
[6] Q. Wu, R. Law, E. Wu, J.X. Lin, hybrid-forecasting model reducing Gaussian noise based on the Gaussian support vector regression machine and chaotic particle swarm optimization, Inf. Sci. 238 (2013) 96–110.
[7] Z. Zhao, S. Xu, S. Zheng, J. Shang, Cognitive radio adaptation using particle swarm optimization, Wirel. Commun. Mobile Comput. 9 (2009) 875–881.
[8] B. Wang, Z. He, Y. Sun, Distributed rate allocation and performance optimization for video communication over mesh networks, in: Proceeding of the IEEE International Conference on Image Processing, vol. 6, 2007, pp. 501–504.
[9] C.C. Lin, Dynamic router node placement in wireless mesh networks: A PSO approach with constriction coefficient and its convergence analysis, Inf. Sci. 232 (2013) 294–308.
[10] B. Liu, L. Wang, Y.H. Jin, et al., Improved particle swarm optimization combined with chaos, Chaos Solitons Fractals 25 (5) (2005) 1261–1271.
[11] L.S. Coelho, B.M. Herrera, Fuzzy identification based on a chaotic particle swarm optimization approach applied to a nonlinear yo-yo motion system, IEEE Trans. Ind. Electron. 54 (6) (2007) 3234–3244.
[12] X.L. Tang, L. Zhuang, C.J. Jiang, Prediction of silicon content in hot metal using support vector regression based on chaos particle swarm optimization, Expert Syst. Appl. 36 (2009) 11853–11857.
[13] L.Y. Chuang, C.H. Yang, J.H. Tsai, C.H. Yang, Operon prediction using chaos embedded particle swarm optimzation, IEEE/ACM Trans. Comput. Biol. Bioinf. 10 (5) (2013) 1299–1309.
[14] Q. Wu, R. Law, E. Wu, J.X. Lin, A hybrid-forecasting model reducing Gaussian noise based on the Gaussian support vector regression machine and chaotic particle swarm optimization 238 (2013) 96–110.
[15] X.D. Li, X. Yao, Cooperatively coevolving particle swarms for large scale optimization, IEEE Trans. Evol. Comput. 16 (2) (2012) 210–224.
[16] M.Q. Tang, Y.L. Xin, J. Li, J.G. Zhai, Nonconvex resource control and lifetime optimization in wireless video sensor networks based on chaotic particle swarm optimization, Appl. Soft Comput. 13 (2013) 3273–3284.
[17] F. Ahmed, O. Tirkkonen, A.A. Dowhuszko, M. Juntti, Distributed power allocation in cognitive radio networks under network power constraint, in: Proceeding of the 2014 Ninth International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM), 2014, pp. 492–497.
[18] N. Mokari, S. Parsaeefard, H. Saeedi, P. Azmi, Cooperative secure resource allocation in cognitive radio networks with guaranteed secrecy rate for primary users, IEEE Trans. Wirel. Commun. 13 (2) (2014) 1058–1073.
[19] A. Goldsmith, Wireless Communications, Cambridge University Press, 2005.
[20] M. Gursoy, D. Qiao, S. Velipasalar, Analysis of energy efficiency in fading channels under qos constraints, IEEE Trans. Wireless Commun. 8 (8) (2009) 425–4263.
[21] F.R. Yu, X. Zhang, V.C.M. Leung, Green Communications and Networking, CRC Press, 2012.
[22] S. Buzzi, H.V. Poor, Joint receiver and transmitter optimization for energy-efficient CDMA communications, IEEE J. Sel. Areas Commun. 26 (3) (2008) 459–472.
[23] F. Meshkati, H.V. Poor, S.C. Schwartz, Energy efficiency-delay tradeoffs in CDMA networks: A game-theoretic approach, IEEE Trans. Inf. Theory 55 (7) (2009) 3220–3228.
[24] S. Buzzi, D. Saturnino, A game-theoretic approach to energy efficient power control and receiver design in cognitive CDMA wireless networks, IEEE J. Sel. Topics Signal Process. 5 (1) (2011) 137–150.
[25] Y. Wu, D.H.K. Tsang, Energy-efficient spectrum sensing and transmission for cognitive radio system, IEEE Commun. Lett. 15 (5) (2011) 545–547.
[26] Y. Wang, W.J. Xu, K.W. Yang, J.R. Lin, Optimal energy-efficient power allocation for OFDM-based cognitive radio networks, IEEE Commun. Lett. 16 (9) (2012) 1420–1422.
[27] J. Mao, G. Xie, J.C. Gao, Y.N. Liu, Energy efficiency optimization for cognitive radio MIMO broadcast channels, IEEE Commun. Lett. 17 (2) (2013) 337–339.
[28] W. Zhong, J.H. Wang, Energy efficient spectrum sharing strategy selection for cognitive MIMO interference channels, IEEE Trans. Signal Process. 61 (14) (2013) 3705–3717.
[29] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceeding of the 1995 IEEE International Conference on Neural Networks ICNN, 1995, pp. 1942–1948.
[30] M. Xiao, N. Shroff, E. Chong, A utility-based power control scheme in wireless cellular systems, IEEE/ACM Trans. Netw. 11 (2) (2003) 210–221.
[31] S.A. Kazarlis, A.G. Bakirtzis, V. Petridis, A genetic algorithm solution to the unit commitment problem, IEEE Trans. Power Syst. 11 (1) (1996) 83–92.
[32] K. Deep, M. Thakur, A new mutation operator for real coded genetic algorithms, Appl. Math. Comput. 193 (2007) 211–230.
[33] K. Price, R. Storn, J. Lampinen, Differential Evolution: A Practical Approach to Global Optimization, Springer-Verlag, Berlin, Germany, 2005.
[34] Y. Shi, R.C. Eberhart, A modified particle swarm optimizer, in: Proceedings of the 2005 International Conference on Machine Learning and Cybernetics, 2005, pp. 69–73.
[35] Y. Shi, R.C. Eberhart, Empirical study of particle swarm optimization, in: Proceedings of the 1999 Congress on Evolutionary Computation, 1999, pp. 1945–1950.
[36] D. Sofge, K.D. Jong, A. Schultz, A blend population approach to cooperative coevolution for decomposition of complex problem, in: Proceeding of the 2002 Congress on Evolutionary Computation, vol. 1, 2002, pp. 413–418.
[37] Y. Shi, H. Teng, Z. Li, Coopertaive co-evolutionary differential evolution for function optimization, in: Proceedings of the First International Conference on Natural Computation, Springer-Verlag, 2005, pp. 1080–1088.
[38] R.A. Krohling, S. Coelho, Coevolutionary particle swarm optimization using Gaussian distribution for solving constrained optimization problems, IEEE Trans. Syst. Man Cybern. B Cybern. 36 (6) (2006) 1407–1416.
[39] Q. Wu, R. Law, E. Wu, J.X. Lin, A hybrid-forecasting model reducing Gaussian noise based on the Gaussian support vector regression machine and chaotic particle swarm optimization, Inf. Sci. 238 (2013) 96–110.
[40] F. Zhuang, F.D. Galian, Unit commitment by simulated annealing, IEEE Trans. Power Syst. 5 (1) (1990) 311–317.
[41] S.L. Rosen, C.M. Harmonosky, An improved simulated annealing simulation optimization method for discrete parameter stochastic systems, Comput. Oper. Res. 32 (2005) 343–358.

Please cite this article as: M. Tang, Y. Xin, Energy efficient power allocation in cognitive radio network using coevolution chaotic particle swarm optimization, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.02.010

Meiqin Tang received the BS degree in computer science and technology, and MS degree in operational research and control theory from Ludong University, China, in 2002 and 2005, respectively. She received Ph.D degrees in control theory and engineering from Yanshan University, China, in 2009. She is currently an associate professor at Ludong University. She visited Department of Electrical and Computer Engineering, University of Alberta in 2014. Her research interests include wireless networks and optimization theory.

Yalin Xin received the BS degree in computer science and technology, and MS degree in computer science from Ludong University and Ocean University of China, in 2002 and 2012, respectively. He is pursuing his Ph.D degree at Ocean University of China. His research interests include wireless networks and control theory.

Please cite this article as: M. Tang, Y. Xin, Energy efficient power allocation in cognitive radio network using coevolution chaotic particle swarm optimization, Computer Networks (2016), http://dx.doi.org/10.1016/j.comnet.2016.02.010