International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 13 (2017) pp. 3879-3887

Social Spider Algorithm-based Spectrum Allocation Optimization for Cognitive Radio Networks

Binh Thanh Dang
Faculty of Information Technology, Industrial University of HCMC,
Ho Chi Minh City, Vietnam.

Minh Cong Vo
Faculty of Computer Science, University of Information Technology,
Faculty of Information Technology, Industrial University of HCMC,
Ho Chi Minh City, Vietnam.

Tung Khac Truong
Faculty of Information Technology,
Industrial University of HCMC, Ho Chi Minh City, Vietnam.

Abstract

In recent years, applying intelligent algorithms in solving hard problems has become a favorite topic in next generation networking research. Although classical methods have achieved many important outcomes, this new trend promises significant results. This paper addresses the spectrum allocation problem in Cognitive Radio Networks (CRN), in which we proposed a new solution based on Social Spider Algorithm (SSA) to search for the optimal allocation scheme. The numerical results prove the superior of our approach in comparison to other methods.

Keywords: Spectrum allocation; optimization; cognitive radio networks; swarm-based algorithm

INTRODUCTION

In recent years, wireless spectrum allocation optimization has become a favorite topic in next generation networking research [1]. Traditionally, the spectrum is managed by the state and regulations, in which the major approach is constructing a fixed allocation system. As an apparent result, the main drawbacks of this tactic are low utilization of the bandwidth, waste of sparse or unassigned channels while others are so crowded and lead to undesired interference level in the current communication channel. Those disadvantages may not be a serious problem in the past, but with the rapid growth in the quantity of the wireless devices connecting to the network, which is forming the Internet of Things (IoT), the situation has changed dramatically, where unoptimized utilization of the spectrum resources are not acceptable any more. Cognitive radio [2][3][4] is commonly considered to be the solution for the case: it promised more intelligent spectrum allocation schemes in a more dynamic and fruitful style.

As time goes by, many proposals to solve the spectrum allocation problem have been presented [5]. In [6], the authors investigated the influence of multi-cell, multi-operator interference on the wireless resources in case many operators co-exist and consume a shared spectrum repository. Moreover, they suggested a framework designed to improve the bandwidth utilization while providing a reasonable QoS degree by ensuring only a slight interference level among operators.

Another approach is using intelligent algorithms to search for the optimum allocation scheme, as implemented in [7], [8], and [9]. The identical idea appeared in these works are using swarm-based algorithms to carry out a global search for the best answer satisfying a pre-defined threshold. In this paper, we propose a centralized optimal spectrum allocation (COSA) scheme that is possible to satisfy QoS requirements, which were described in [10], at a better degree than previous solutions. Meanwhile, we also present a modified version of Social Spider Algorithm (SSA) [11] to benefit its ability in allocation matrix decision optimization. The rest of this paper consists of below parts. Firstly, related works are described in part 2. Then, Cognitive Radio Networks Spectrum Allocation Problem (CRNSAP) is mentioned in part 3. Afterwards, part 4 presents our proposed SSA-based spectrum allocation scheme, where major modifications in the algorithm to support cognitive radio networks are clarified. Our simulation preparation process and numerical results are discussed in part 5. Eventually, conclusions and future considerations are provided in the last section.

RELATED WORKS

In this section, we will focus on spectrum allocation improvements based on optimization algorithms. We choose to describe here one of the most famous ones, spectrum allocation based on Particle Swarm Optimization (PSO) [12], which was introduced in [9]. This part of the paper will also describe the base algorithm for our proposed scheme, Social Spider Algorithm [11].

A. PSO for Cognitive Radio Networks

The PSO algorithm, which was presented by Kennedy and Eberhart in [12], is an evolutionary algorithm based on the replication of the social behavior of a bird's population. This population is called a swarm while the individuals in the population are named particles. A particle represents a probable solution for the optimization problem.

Particle i at iteration t is determined by its position in D-dimension search $x_i^t = \left[x_{i1}^t, x_{i2}^t, ..., x_{iD}^t\right]$. The particles are evaluated by the fitness function $f\left(x_i^t\right)$.

In PSO, each particle travels towards the best position. The mobility and the velocity are affected by two aspects: the personal best position of each particle, so-called *pbest* $p_i^t = \left[p_{i1}^t, p_{i2}^t, ..., p_{iD}^t\right]$, and the global best position of the entire population named *gbest* $p_g^t = \left[p_{g1}^t, p_{g2}^t, ..., p_{gD}^t\right]$. The velocity of the particle i is represented as $v_i^t = \left[v_{i1}^t, v_{i2}^t, ..., v_{iD}^t\right]$. The velocity and the position of particle i are brought up-to-date depending on the below formulas:

$$v_{id}^{t+1} = \omega v_{id}^t + c_1 r_1 \left(p_{id}^t - x_{id}^t\right) + c_2 r_2 \left(p_{gd}^t - x_{gd}^t\right), \quad (1)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (2)$$

where ω is the inertia weight; c_1 and c_2 are learning rates; r_1 and r_2 are two uniformly distributed arbitrary numbers produced independently in $[0,1]$.

For the purpose of [9] is to make the most of the throughput of the entire cognitive radio network, the fitness function was outlined as follows

$$f\left(x_{in,im}^t\right) = \begin{cases} \sum_{n=1}^{N} \sum_{m \in C_s} x_{in,im}^t r_{n,m}, & SI_i^t \le I_{limit}, PI_{in}^t \le P_{limit} \\ -1, & others \end{cases} \quad (3)$$

Stepwise process of the PSO-based spectrum allocation algorithm written in [9] is as follows:

Step 1: Generate channel availability matrix L, channel bandwidth matrix B, channel state information matrix G, interference constraint matrix C, and transmit power matrix P.

Step 2: Establishment of the parameters required by the algorithm: the quantity of the particles, learning factors c_1, c_2, max repetition times t_{max}, inertia weight ω, the original temperature T_0, the cooling rate α, and the velocity range $\left[-V_{\max}, V_{\max}\right]$. The next tasks are randomly assignment of the positions and velocities of the particles in search space, ensuring that the positions of the particles meet the expectations of the constraint matrix C, calculating p_i, and finding the global best position p_g.

Step 3: Calculate the acceptance possibility value p_i^t, and the roulette portion Q_i^t at current temperature. Execute roulette approach to pick the global best position p_g from p_i.

Step 4: Modify the positions and velocities, making sure that the position of the particles pleases the restrictions of matrices L and C. If the velocity $v_{in,im}^{t+1} > V_{max}$, make $v_{in,im}^{t+1} = V_{max}$, and if $v_{in,im}^{t+1} < -V_{max}$, make $v_{in,im}^{t+1} = -V_{max}$.

Step 5: Reckon the fitness values of all particles, make *pbest* p_i up to date and decrease the temperature.

Step 6: If the iteration times reaches max iteration times t_{max}, stop the loop and output the global best position p_g as well as the fitness of p_g. If not, return to step 3.

B. Social Spider Algorithm (SSA)

SSA [11] is a metaheuristic global optimization algorithm encouraged by the behavior of social spider. In SSA, the solution of an optimization problem is characterized by the position of an artificial spider on the hyper-dimension spider web. The spiders move on the web and they share the position information via vibrations. Each spider could move independently as long as it does not leave the web. Whenever a spider moves to a new location, it produces a vibration, which is propagated through the web. Depend on the receive vibrations from other ones, a spider will be guided to the optimal position. Details of SSA will be described in the following subsections.

a) Spider

The artificial spiders are the major functioning entities of SSA. Every spider stays at a specific position on the hyper-dimension spider web, and the fitness value of this position is linked to the spider. Additionally, the spider has a memory space used to store its condition along with optimization parameters, including its present position, present fitness value, subsequent vibration at earlier iteration, inactive level, preceding movements, and a dimension mask. These characteristics help the spider to look for the ideal solution.

b) Vibration

Vibration is a vital concept in SSA. It is one of the key features that separate SSA from different metaheuristic algorithms. SSA uses two properties to describe a vibration, that is, the source position and the source intensity of the vibration.

The following equation defines how to compute the source vibration intensity

$$I(s) = \log(\frac{1}{f(s) - C} + 1) \quad (4)$$

where $f(s)$ is the fitness rate of the spider s, and C is a constant in order that the minimum fitness values are larger than C.

The equation below calculates the attenuation when the vibration is transmitted from spider s to spider s':

$$I(s, s') = I(s) \times \exp\left(\frac{D(s, s')}{\overline{\sigma} \times r_a}\right) \quad (5)$$

where $r_a \in (0, \infty)$ is a user-controlled operand. This parameter manipulates the attenuation rate of the vibration intensity over distance. Meanwhile, $\overline{\sigma}$ is the standard deviation of all spider positions along each dimension.

c) Search Pattern

SSA processes a population of spiders through a sequence of optimization phases. Particularly, each iteration of SSA could be separated into the below steps.

FITNESS EVALUATION

On the start of an iteration, the fitness values of the positions of all spiders is reassessed. These fitness values will be used in the vibration generation and broadcast procedure.

VIBRATION GENERATION

Firstly, a new vibration is created for each spider. Subsequently, that vibration is spread to all other spiders in the web with distance consideration: the vibration will be attenuated though the space. Later, the largest attenuated vibration intensity is carefully chosen depends on the received vibrations, and compare with the previous one. As a result, the larger intensity vibration is stored as the target vibration. If the spider decides to alter its stored vibration, the inactive level is increased by one, otherwise it is set to zero value. This level purpose is to help the algorithm avoid local optima.

MASK CHANGING

After the following vibrations of every spiders are computed, this phase will update their positions. In this stage, a dimension mask is used to guide the movement [11]. Each spider holds a binary vector mask whose length is the dimension of the optimization problem. In each iteration, a spider has a probability $1 - p_c^{N_{in}}$ to modify its mask. for clarification, N_{in} is the spider's inactive number. Whenever a mask is decided to be changed, each bit in the mask has a probability p_m to get a 1 value, and a probability $1 - p_m$ to be set a 0 value.

RANDOM WALK

At this phase, each spider carries out a random walk to improve their positions. After the walk, the new solution should be fixed to ensure that no spider moves out of the web. A more detailed description of the random walk considerations could be found in [11].

COGNITIVE RADIO NETWORKS SPECTRUM ALLOCATION PROBLEM (CRNSAP)

In a wireless network, a user is an object that uses a channel (a piece of the radio spectrum) to send and receive information. They are split into two types: primary users and secondary users. The basic rule is primary users always have higher priority over all secondary users in their registered frequency bands. In another way, secondary users may utilize these channels only when they are not being employed by primary users. Moreover, secondary users must give up these channels on any occasion the primary users want them.

In this paper, we will outline the problem as described in [10]. Suppose that every user (both primary or secondary) uses an omni-directional antenna and it could manipulate the transmission power and consequently its interference range.

We call $d_t(n, m)$ the interference range of user n, whose is being assigned with channel m and have the type is t, where t

could be either "*p*" or "*s*", which stands for a primary user or a secondary user, respectively. Firstly, all primary users decide their favorite channels and the corresponding interference ranges (by ruling the transmission powers). Then, the secondary users could fix the upper-bound transmission powers and the interference ranges in order that there should not be any interferences with primary users. Because of the hardware limitation, the interference range should be constrained, given by $d_{min} \le d_t(n,m) \le d_{max}$ for user n and channel m.

Assume that the network consists of N secondary users and M orthogonal channels. We could build the channel availability matrix $L = \left[l_{n,m} \middle| l_{n,m} \in \{0,1\} \right]_{N \times M}$ based on the positions and the interference ranges of all primary and secondary users. In this matrix, $l_{n.m} = 1$ has the meaning that channel m is ready for secondary user n to use. Otherwise, $l_{n.m}$ will be set to zero.

Additionally, we also define the channel reward matrix $B = \left[b_{n,m} \right]_{N \times M}$. Each element $b_{n.m}$ symbolizes the reward when a secondary user n chooses to use channel m. Furthermore, we illustrate the interference among the secondary users by the interference constraint matrix $C = \left[c_{n,k,m} \middle| c_{n,k,m} \in \{0,1\} \right]_{N \times N \times M}$, where $c_{n.k,m} = 1$ means that user n will interfere with user k if both of them employ channel m. Otherwise, $c_{n.k,m}$ is set the zero value.

Eventually, the channel assignment matrix $A = \left[a_{n,m} \middle| a_{n,m} \in \{0,1\} \right]_{N \times M}$ is used to specify which channels are permitted to be used by the secondary users, where $a_{n.m} = 1$ implies that channel m is allocated to secondary user n, and $a_{n.m} = 0$ means that channel m is not assigned to secondary user n.

Next, we will discuss about system constraints. If all secondary user channel assignments, which are carried by matrix A, only allocated with channels which do not conflict with any other users, we could conclude that matrix A is in a nonconflicting condition. This can be defined by equation (6) below.

$$a_{n,m} + a_{k,m} \le 1, \quad \forall c_{n,k,m} = 1, 1 \le n,k \le N, 1 \le m \le M \tag{6}$$

Besides, because of hardware limits, every radio interface in a CRN system should have a limitation C_{max} on the maximum number of channels allotted [10]. This can be stated as

$$\sum_{m=1}^{M} a_{n,m} \le C_{max}, \qquad \forall 1 \le n \le N \tag{7}$$

Ultimately, the purpose of solving the CRNSAP is maximizing the reward obtained from an assignment A. This could be characterized by the utility function $U(A)$. As mentioned in [10], we define the utility as:

1) Max-Sum-Reward (MSR):

$$U_{MSR}(A) = \sum_{n=1}^{N} \sum_{m=1}^{M} a_{n,m} b_{n,m} \tag{8}$$

2) Max-Min-Reward (MMR):

$$U_{MMR}(A) = \min_{1 \le n \le N} \sum_{m=1}^{M} a_{n,m} b_{n,m} \tag{9}$$

3) Max-Proportional-Fair (MPF):

$$U_{MPF}(A) = \left(\prod_{n=1}^{N} \left(\sum_{m=1}^{M} a_{n,m} b_{n,m} + 10^{-6} \right) \right)^{\frac{1}{N}} \tag{10}$$

In these objective functions, MSR and MMR try to boost the reward of the entire system and that of the most disadvantaged user, respectively. Meanwhile, MPF is designed to ensure the fairness in channel assignment.

The solution to the problem is an assignment matrix A. Those assignments which satisfy both constraints defined in equations (6) and (7) together form the reasonable solution set Λ. Generally, CRNSAP could be expressed as

$$\max_{A \in \Lambda} U(A) \tag{11}$$

subject to

$$a_{n,m} + a_{k,m} \le 1, \forall c_{n,k,m} = 1,$$
$$1 \le n,k \le N, 1 \le m \le M,$$
$$\sum_{m=1}^{M} a_{n,m} \le C_{max}, \forall 1 \le n \le N,$$

where $U(A)$ could be $U_{MSR}(A)$, $U_{MMR}(A)$, $U_{MPF}(A)$

PROPOSED SSA-BASED SPECTRUM ALLOCATION ALGORITHM FOR CRN

We have made many modifications to the original SSA algorithm to support the spectrum allocation problem in CRN.

A detailed flowchart of our version of SSA for the CRNSAP is provided in Fig.1. The below sub-sections will clarify what we altered to the original SSA proposal.

A. Solution Representation

A solution for the CRNSAP is a channel assignment matrix called *A*, in which the columns stand for channels and rows represent secondary users (SU). For example, the matrix below:

$$A = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

shows that SU1 is granted channels 1 and 3, SU2 is assigned channels 2 and 3, while SU3 gets channel 1.

In our algorithm, an intermediate channel assignment matrix used to be called *X*. Note that the final *X* after all alternations and evolutionary is exactly the *A* matrix.

The shape of *X* in the processing procedure could be a binary vector or a matrix.

B. Real-to-Binary Number Conversion Function

In our proposed algorithm, the sigmoid function is utilized to convert real numbers to binary ones. This function was also used in [13], [14], and [15].

Our real-to-binary number conversion function plays its part at two places in our simulation code: after initiation of the assignment matrix and after the random walk. The positions are in real values then and they should be converted to binary values by using the equation (13) below:

$$X_{s,i}(t+1) = \begin{cases} 0 & if\ rand() \geq S(P_{s,i}(t+1)) \\ 1 & if\ rand() < S(P_{s,i}(t+1)) \end{cases} \tag{12}$$

where $S(\cdot)$ is the sigmoid function for transforming the velocity to the probability as the following equation:

$$S(P_{s,i}(t+1)) = \frac{1}{1+e^{P_{s,i}(t+1)}} \tag{13}$$

Fig.2 will describe the graph drawn from the outputs of the sigmoid function.

C. Constraints Resolution

A specific channel assignment matrix will be checked with these stuffs to be valid:

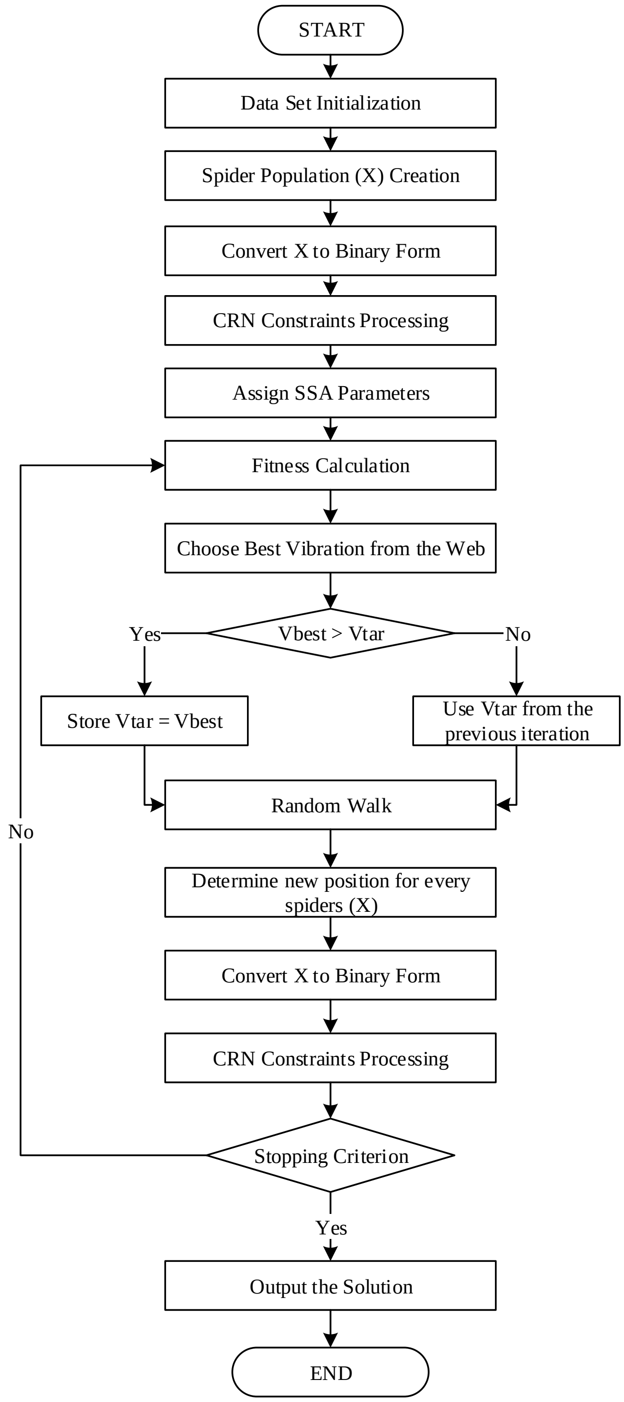

Figure 1: Our SSA-based CRN Spectrum Allocation Algorithm

- Channel availability matrix *L*
- Interference constraint matrix *C*
- Maximum number of channels allotted C_{max}

Rules fox handling a channel assignment matrix *X* are as follows. Firstly, we check *X*'s elements with matrix *L*. If *L(t)=0* then *X(i,t)* is set the zero value, for *i* in range 1 to the population size, and *t* ranges from 1 to the number of dimensions. Secondly, the resulting *X* will be matched with the interference constraint *C*.

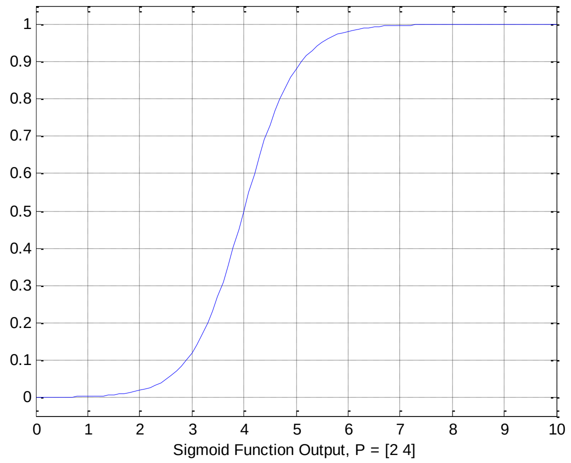

Figure 2: Sigmoid function used in our proposed algorithm

In particular, the modifications in a channel assignment matrix *X* will be controlled by the value of an element *C(i,j,m)* in the matrix *C*. If *C(i,j,m)*=1 then there will be a collision when both *X(i,m)* and *X(j,m)* are set the value 1s. In this context, *X(i,m)* or *X(j,m)* will be set the value 0 based on a random probability.

Afterwards, *X* will be checked with C_{max}. Each row in the *X* matrix will be examined and reassign the value if the requirement mentioned in equation (7) is violated.

PERFORMANCE EVALUATIONS

We conduct all the simulations on the same hardware system and use only on system parameters set for fair comparison between our proposed algorithm and the referenced solution. An Intel(R) Xeon(R) E5-2620 6 cores (12 threads) CPU with 8GB RAM computer system is used.

Each algorithm is simulated 30 times and the values demonstrated in the diagrams are the average ones derived from these tests. In each run, our SSA-based algorithm uses the loop of 1000 iterations to search for the best solution. For equity, the PSO-based simulation uses the same number of loop.

As mentioned before, we have primary and secondary users in a CRN system. We decide to create *K* primary users and assign channels to them. Note that there are *M* channels, *M≠K*, and some primary users could share the same channel.

Afterwards, the matrices *L, B, C* are generated based on the pseudo codes given by [10]. An initial solution *X* will also be produced. We then convert *X* to a binary matrix to fulfil the requirements of the CRNSAP.

In addition, our simulation uses the parameters described in the table below. Note that the population size, number of channels, number of primary users and secondary users, number of dimensions of the optimization problem, maximum iteration, maximum transmission power of secondary users, minimum transmission power of secondary users are the same for the two algorithms. Other variable specific to each algorithm is used as default values, which are described in [7] and [9].

Table 1: Simulation parameters

Parameter	**Variable name**	**Value**
Number of individuals	*pop_size*	20
Number of dimenssions	*dim*	M×N
Maximum iteration	*max_iter*	1000
Rate of vibration attenuation when propagating over the web	r_a	1
Interference range (protection area)	*DPR*	2
Probability of changing the mask	p_c	0.7
The probability for each bit of the vector to be assigned the value 1 if the mask changed	p_m	0.1

To secure the fairness, we use the same data set which is generated prior to the simulations as the standard problem for both SSA and PSO-based spectrum allocation schemes. Additionally, there are other stuffs to be produced before the tests, including the matrices *L, B, C* and specifications for primary and secondary users.

We used the pseudo codes given in [10] and [16] to create these data and all codes are programmed in MATLAB. For CRNSAP is a maximization issue while SSA is proposed for minimization, we conduct some modifications to the fitness function as in [7]. Rather than maximizing $U(A)$, we decide to minimize the utility function $U'(A) = 1000 - U(A)$ in the iterations of SSA. Finally, we output $U(A_{best}) = 1000 - U'(A_{best})$ as the concluded solution for each simulation.

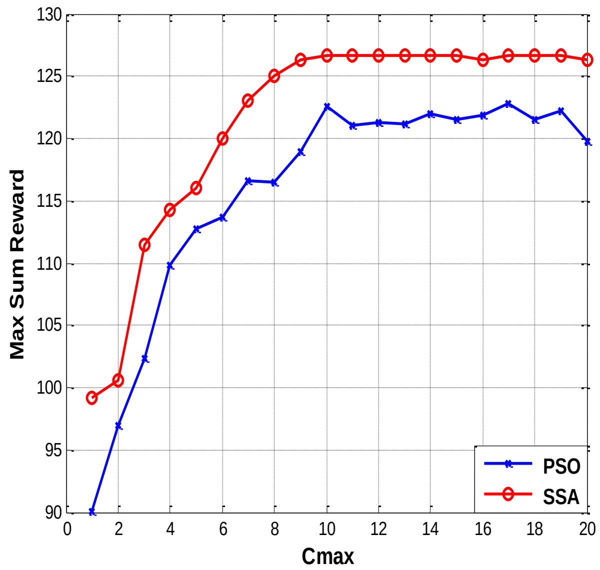

Figure 3: MSR comparison between SSA and PSO

In this paper, we study the impact on the amount of permitted allocated channels to the users by increasing the value of C_{max} from 1 to 20 (step=1). A clear conclusion could be drawn from the benchmarks is the smaller C_{max} is, the smaller the number of channels could be given to the users, and as a result, the smaller the fitness values could be achieved.

Our simulation results are demonstrated in below diagrams. Each graph visualizes the performance of SSA-based and PSO-based algorithms for spectrum allocation problem in CRN in a different requirement.

Fig. 3 shows their capabilities in a system focus on maximization of the total of reward values of channel assignments. Our proposed algorithm always passes the results attained by PSO-based scheme. The differences vary from 2.92% to 10.04%. for each C_{max} value.

Fig.4 and Fig.5 are where our proposed method outperforms the PSO-based scheme. Fig.4 clarifies the performance of the two approaches when trying to maximize the minimum reward values when we increase the C_{max} value from 1 to 20. While PSO-based technique has unstable outputs, SSA-based algorithm always has significantly higher MMR. Moreover, our solution also proves that it could achieve the best solution faster than the other one.

The output of SSA-based algorithm quickly reaches the peak value and saturates at C_{max}=4, while PSO-based one could not increase the desired value since $C_{max}=6$. Subsequently, the performance of our algorithm is very balanced whilst PSO-based one shows that it does not deals with this situation very well.

A reason for this circumstance is the stopping criterion. While the output could be better, the stopping criterion prevents the algorithms to dig deeper into the search space. Nonetheless, our algorithm continually show that it is a better choice in any contexts.

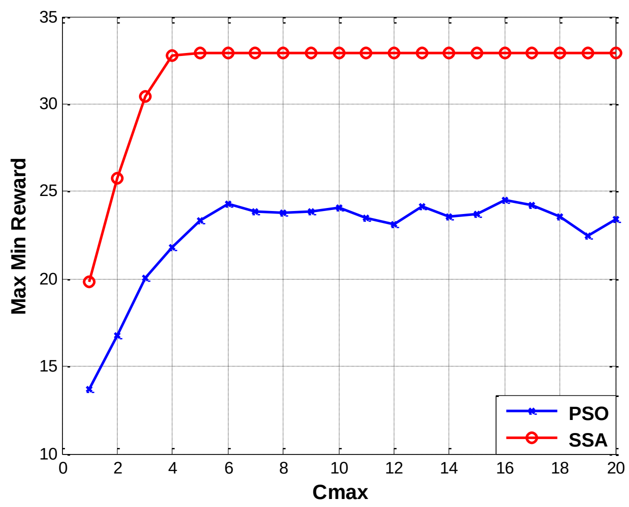

Figure 4: MMR comparison between SSA and PSO

The same phenomenon reappears in Fig.5. The trendlines in Fig.5 also show a huge gap between our algorithm and the PSO-based one. The separations between the two algorithms ranges from 50.73% to 71.22% at C_{max}=12. The difference is so large that we have to research to find out why this could happen.

The dramatic capacity of our algorithm could be explained by the nature of the approaches. SSA has a novel solution where every spider contributes to the decision of where to move to in the next step of a specific one. A spider supports other ones and its weight on a specific movement relies on its attenuated vibration at the one that should decide what to do.

Anther fact should be take into consideration is that our proposal reaches the saturation point at a higher C_{max} value than that of the PSO solution for CRNSAP. This situation could be explained by the different priorities of the objective functions.

More detailed numerical results are provided in the Table II.

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 13 (2017) pp. 3879-3887
© Research India Publications. http://www.ripublication.com

Table II. Performance Analysis Results

Num. Run = 30	***SSA***			***PSO***		
	Max	***Mean***	***Std***	***Max***	***Mean***	***Std***
C_{max}=5	120.0311	115.5627	2.0891	114.3800	112.6613	2.4149
C_{max} =10	126.6924	126.3442	1.5056	126.6900	120.3907	4.6760
C_{max} =15	126.6924	126.6091	0.4564	126.6900	123.4263	4.8822
C_{max} =20	126.6924	126.6924	8.67E-14	126.6900	121.8513	5.4909

(a) Max Sum Reward

Num. Run = 30	***SSA***			***PSO***		
	Max	***Mean***	***Std***	***Max***	***Mean***	***Std***
C_{max}=5	32.9270	32.9270	1.4454E-14	26.6630	23.3863	1.9024
C_{max} =10	32.9267	32.9267	1.4454E-14	28.1380	23.2286	1.9260
C_{max} =15	32.9270	32.8768	0.2750	27.8450	22.9881	2.1387
C_{max} =20	32.9270	32.9270	1.4454E-14	31.2250	23.8922	2.7711

(b) Max Min Reward

Num. Run = 30	***SSA***			***PSO***		
	Max	***Mean***	***Std***	***Max***	***Mean***	***Std***
C_{max}=5	70.8380	68.8248	0.8959	48.4730	44.2879	1.8531
C_{max} =10	78.6840	77.7004	0.6372	53.7180	46.0645	2.8729
C_{max} =15	78.8160	77.9925	0.5220	50.2800	45.8494	2.3497
C_{max} =20	78.6260	77.9900	0.4987	53.6750	46.4900	2.1688

(c) Max Proportional Fair

The sub-tables of Table II. once again represent the domination of our algorithm over the referenced one. It is firmly better on max and mean values while the standard deviation (Std) is significantly lower than PSO-based approach.

CONCLUSIONS

In this paper, we have introduced a promising approach in solving the CRNSAP. Its amazing performance proves that there is always some way to enhance the system performance, especially when we utilize a revolutionary swarm-based algorithm to search for best solution for hard problems. Our proposed algorithm shows some unstable results in MSR objective function. There also more ways to optimize the performance of the algorithm, such as parallelize the computing system or combine with solutions from other algorithms. That will be where we focus in improving our proposal in the next works.

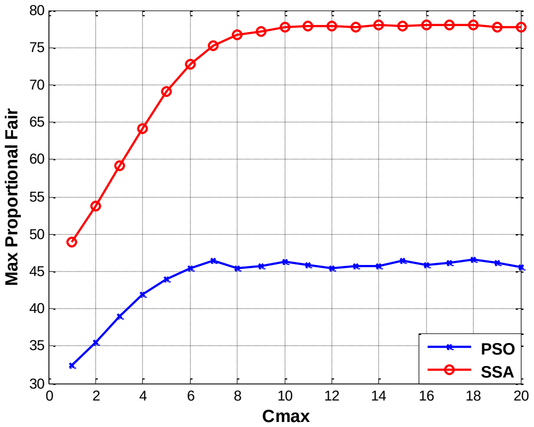

Figure 5: MPF comparison between SSA and PSO

REFERENCES

[1] J. Marinho and E. Monteiro, “Cognitive radio: Survey on communication protocols, spectrum decision issues, and future research directions,” *Wirel. Networks*, vol. 18, no. 2, pp. 147–164, 2012.

[2] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, “NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey,” *Comput. Networks*, vol. 50, no. 13, pp. 2127–2159, 2006.

[3] J. Mitola and G. Q. Maguire, “Cognitive radio: making software radios more personal,” *IEEE Pers. Commun.*, vol. 6, no. 4, pp. 13–18, 1999.

[4] Beibei Wang and K. J. R. Liu, “Advances in Cognitive Radio Networks: A survey,” *IEEE J. Sel. Top. Signal Process.*, vol. 5, no. 1, pp. 5–23, 2011.

[5] E. Z. Tragos, S. Zeadally, A. G. Fragkiadakis, and V. A. Siris, "Spectrum Assignment in Cognitive Radio Networks, A Comprehensive Survey," *IEEE Commun. Surv. Tutorials*, vol. 15, no. 3, pp. 1108–1135, 2013.

[6] G. Alnwaimi, K. Arshad, and K. Moessner, "Dynamic Spectrum Allocation Algorithm with Interference Management in Co-Existing Networks," *IEEE Commun. Lett.*, vol. 15, no. 9, pp. 932–934, 2011.

[7] A. Y. S. Lam and V. O. K. Li, "Chemical Reaction Optimization for Cognitive Radio Spectrum Allocation," *2010 IEEE Glob. Telecommun. Conf. GLOBECOM 2010*, pp. 1–5, 2010.

[8] T. Siddique and A. Azam, "Spectrum Optimization in Cognitive Radio Networks using Genetic Algorithms," Blekinge Institute of Technology, 2010.

[9] Z. Jie and L. Tiejun, "Spectrum Allocation in Cognitive Radio with Particle Swarm Optimization Algorithm," pp. 1–10, 2012.

[10] C. Peng, H. Zheng, and B. Y. Zhao, "Utilization and fairness in spectrum assignment for opportunistic spectrum access," *Mob. Networks Appl.*, vol. 11, no. 4, pp. 555–576, 2006.

[11] J. J. Q. Yu and V. O. K. Li, "A social spider algorithm for global optimization," *Appl. Soft Comput. J.*, vol. 30, pp. 614–627, 2015.

[12] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," *Encyclopedia of Machine Learningachine learning*. Springer US, pp. 760–766, 2011.

[13] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," *1997 IEEE Int. Conf. Syst. Man, Cybern. Comput. Cybern. Simul.*, vol. 5, pp. 4104–4108, 1997.

[14] M. A. Khanesar, M. Teshnehlab, M. A. Shoorehdeli, and E. E. Faculty, "A novel binary particle swarm optimization," *2007 Mediterr. Conf. Control Autom.*, vol. 1, no. 1, pp. 1–6, 2007.

[15] S. Mirjalili, S. M. Mirjalili, and X. S. Yang, "Binary bat algorithm," *Neural Comput. Appl.*, pp. 1–19, 2013.

[16] A. Y. S. Lam, V. O. K. Li, J. J. Q. Yu, and S. Member, "Power-Controlled Cognitive Radio Spectrum Allocation with Chemical Reaction Optimization," vol. 12, no. 7, pp. 3180–3190, 2013.