This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2010 proceedings.

Chemical Reaction Optimization for Cognitive Radio Spectrum Allocation

Albert Y.S. Lam and Victor O.K. Li
Department of Electrical and Electronic Engineering
The University of Hong Kong
Pokfulam Road, Hong Kong
Email: {ayslam, vli}@eee.hku.hk

***Abstract*—Cognitive radio can help increase the capacity of wireless networks by allowing unlicensed users to use the licensed bands, provided that the occupancy do not affect the prioritized licensed users. One of the fundamental problems in cognitive radio is how to allocate the available channels to the unlicensed users in order to maximize the utility. In this work, we develop an allocation algorithm based on the newly proposed chemical reaction-inspired metaheuristic called Chemical Reaction Optimization (CRO). We study three utility functions for utilization and fairness, with the consideration of the hardware constraint. No matter which utility function is used, simulation results show that the CRO-based algorithm always outperforms the others dramatically.**

I. Introduction

Wireless communication is one of the fastest growing segments in the communication industry. Wireless communication is realized through transmitting signals within a certain frequency range. To prevent interference of wireless signals, the frequency spectrum is divided into multiple bands for different purposes and it is regulated by government agencies. Some spectrum bands are licensed and limited to the use of authorized users (primary users) while some (i.e. unlicensed bands) can be used without restriction. Due to underutilization of the licensed bands and overcrowding of the unlicensed bands, the capacity can be dramatically increased if the unlicensed users (secondary users) are allowed to use the licensed bands. However, primary users have priority in using their respective licensed bands and secondary users can only operate in these bands provided their activities do not affect the primary users. This paradigm, firstly proposed in [1], is known as cognitive radio opportunistic spectrum access.

One important problem in this new paradigm is the allocation of radio spectrum to secondary users effectively in the presence of primary users and we call it the cognitive radio spectrum allocation problem (CRSAP) in this paper. This problem can be solved by a centralized approach or by a distributed strategy. The former refers to the situation in which a central authority (e.g., a spectrum policy server) possesses all necessary (primary and secondary) user information in a given geographical area and assigns available spectrum segments to the secondary users [2], [3]. This is particularly useful for infrastructure-based networks with static environmental and user conditions in a certain period of time. In the latter case, secondary users detect available channels themselves and negotiate channel acquisition with their neighbors according to local information [2], [4]. This favors decentralized ad hoc networks where centralized authorities are unavailable. In this work, we focus on centralized approaches.

CRSAP, a non-convex optimization problem, is proved to be *NP*-hard [2]. Since evolutionary computing techniques (e.g. Genetic Algorithm [5], Ant Colony Optimization [6], Particle Swarm Optimization (PSO) [7], Chemical Reaction Optimization (CRO) [8], etc.) have been successfully applied to these non-convex problems to give near-optimal results, we try to develop a CRO-based centralized algorithm to solve CRSAP. CRO is a (variable) population-based general-purpose optimization metaheuristic. It mimics the interactions of molecules driving towards the minimum state of free energy (i.e. the most stable state). The manipulated agents are molecules, each of which has a molecular structure, potential energy (*PE*), kinetic energy (*KE*), and some other optional attributes. The molecular structure and *PE* corresponds to a solution of a given problem and its objective function value, respectively. *KE* represents the tolerance of a molecule getting a worse solution than the existing one, thus allowing CRO to escape from local optimum solutions. Imagine that we have a set of molecules in a closed container. They move and collide either on the walls of the container (uni-molecular collisions) or with each other (inter-molecular collisions). Each collision results in one of the four types of elementary reactions, including on-wall ineffective collision, decomposition, inter-molecular ineffective collision, and synthesis. They have different characteristics and extent of change to the solutions. With the conservation of energy, the solutions change from high to low energy states and we output the molecular structure with the lowest found *PE* as the best solution.

The rest of this paper is organized as follows. We formulate CRSAP in Section II. In Section III, we describe the proposed CRO-based algorithm for the problem. Section IV gives the simulation results, comparing CRO with some other evolutionary algorithms. We conclude this paper and suggest possible future work in Section V.

II. Problem Formulation

In a wireless network, a channel user is an entity which utilizes a channel (a segment of the radio spectrum) to transmit and to receive data. Primary users have higher priority in

978-1-4244-5638-3/10/$26.00 ©2010 IEEE

their licensed bands over secondary users. In other words, the secondary users can only employ those channels when they are not being used by the primary users and they must give up these channels whenever the primary users need them.

We define the problem according to [2]. Assume that every (primary or secondary) user has an omni-directional antenna. It can control its transmission power and hence its interference range. Let $d_t(n,m)$ be the interference range of user n of type t with channel m, where t can be either "p" for a primary user or "s" for a secondary user. After all primary users have decided their desired channels and the corresponding interference ranges (through controlling the transmission powers), the secondary users can then determine the maximum transmission powers (and also the interference ranges) so that they do not interfere with any primary users. Due to the hardware constraint, the interference range should be bounded, given by $d_{min} \leq d_t(n,m) \leq d_{max}$ for user n and channel m. Assume that there are total N homogeneous secondary users and M orthogonal channels. According to the locations and the interference ranges of both primary and secondary users, we can have the channel availability matrix $L = [l_{n,m} | l_{n,m} \in \{0,1\}]_{N\times M}$, where $l_{n,m} = 1$ means that channel m is available for secondary user n to use. Otherwise, $l_{n,m}$ is equal to zero. We also have the channel reward matrix $B = [b_{n,m}]_{N\times M}$, where $b_{n,m}$ characterizes the reward when secondary user n adopts channel m. Moreover, we describe the interference between the secondary users with the interference constraint matrix $C = [c_{n,k,m} | c_{n,k,m} \in \{0,1\}]_{N\times N\times M}$, where $c_{n,k,m} = 1$ implies that user n will interfere user k if they both use channel m. Otherwise, $c_{n,k,m}$ equals zero. Finally, the channel assignment matrix $A = [a_{n,m} | a_{n,m} \in \{0,1\}]_{N\times M}$ is used to indicate which channels are allowed to be utilized by the secondary users, where $a_{n,m} = 1$ means that channel m is allocated to secondary user n, and $a_{n,m} = 0$ otherwise. An assignment A is conflict-free if a secondary user is only assigned with channels which do not conflict with any other user. This can be described by

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

Moreover, due to hardware limitations, each cognitive radio interface should have a limit C_{max} on the maximum number of channels assigned [2], [9]. This can be expressed as

$$\sum_{m=1}^{M} a_{n,m} \leq C_{max}, \quad \forall 1 \leq n \leq N. \tag{2}$$

We decide to maximize the reward gained from an assignment A represented by utility function $U(A)$. As in [2], [3], we express the utility as:

1) Max-Sum-Reward (MSR):

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

2) Max-Min-Reward (MMR):

$$U_{MMR}(A) = \min_{1\leq n\leq N} \sum_{m=1}^{M} a_{n,m} \cdot b_{n,m}; \text{ and}$$

3) Max-Proportional-Fair (MPF):

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

MSR and MMR maximize the utilization of the whole network and that of the most disadvantaged user, respectively. MPF is for fairness.

An assignment A is a solution to the problem. Those assignments which satisfy both Constraints (1) and (2) form the feasible solution set Λ. Mathematically, CRSAP is presented as

$$\max_{A\in\Lambda} \quad U(A) \tag{3}$$

subject to

$$\begin{aligned} &a_{n,m} + a_{k,m} \leq 1, && \forall c_{n,k,m} = 1, 1 \leq n,k \leq N, \\ & && 1 \leq m \leq M, \\ &\sum_{m=1}^{M} a_{n,m} \leq C_{max}, && \forall 1 \leq n \leq N, \end{aligned}$$

where $U(A)$ can be $U_{MSR}(A)$, $U_{MMR}(A)$, or $U_{MPF}(A)$.

III. Algorithm Design

A. Solution Representation

As pointed out in Section II, a solution of the optimization problem is a channel assignment matrix A. Recall that each entity in the matrix is a 0/1 indicator specifying whether a particular channel is assigned to a certain secondary user. Such a matrix A gives the complete picture for all channels and users (Users in the following refer to secondary users only.). We decide to find an A which can maximize the objective function (3), by iteratively giving different 0/1 combinations for A. However, this matrix representation contains lots of redundancies. As indicated by the channel availability matrix L, not all channels are available to every user. Thus, changing the values of those entities in A (e.g., $a_{n,m}$) whose corresponding entities in L (i.e., $l_{n,m}$) equal zero is useless. Thus, we only consider those entities in A whose channels are available to the users. This creates a much smaller solution space to explore. To do this, similar to [3], we give a vector representation of each solution (see Fig. 1). From L, we can produce a vector *loc* indicating the locations [1] of entities in A which we should pay attention to. Those entities in A which are not located in *loc* are the channels unavailable to the users (we can simply assume they all have values of zero in A). With *loc*, we can easily make the one-to-one mappings between the matrix and vector forms of the solutions (i.e. *M2V* and *V2M* operators).

[1] The locations of entities in L or A are indexed from column to column as we follow the linear indexing of matrix in MATLAB [10] and we perform the simulation in MATLAB in Section IV.

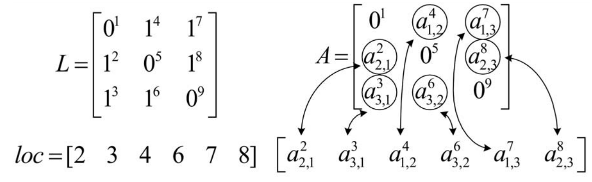


Fig. 1. Matrix and vector representations of solutions

B. Constraint Satisfaction

The problem has two constraints. Randomly generated solutions may result in constraint violation and they are infeasible. Whenever a new solution A is produced by the algorithm, we check and repair any constraint violations by invoking the "$repair$" procedure to generate a feasible A' from A, i.e. $A' = repair(A)$. $repair$ consists of two parts, each of which is used to tackle one of the two constraints. For Constraint (1), we adopt the mechanism given in [3]: if a common channel is allocated to any two conflicting users, we assign the channel to one of them only, instead of both. In other words, for all n, m, and k ($1 \leq n, k \leq N$ and $1 \leq m \leq M$) such that the corresponding $c_{n,k,m}$ equals one, if both $a_{n,m}$ and $a_{k,m}$ are equal to one, we will randomly assign one of them with zero. For Constraint (2), we propose a randomized mechanism to remove any violation: if a user is found to be assigned with z ($z > C_{max}$) channels, we remove the assignment of $z - C_{max}$ channels from the z channels at random.

C. Elementary Reaction Operators

We are going to describe the operators corresponding to the four elementary reactions of CRO. They all operate on the vector representation of solutions only. In the following, we denote a solution in vector form with ω.

1) On-wall Ineffective Collision: In this elementary reaction, a molecule hits a wall of the container. There is little perturbation to the molecule, and thus, a mechanism with a small change to the solution (corresponding to the molecule) can be adopted. In this work, we simply make a new solution ω' from an existing one ω by changing a random bit of ω. For example,

$$\underbrace{[0, 1, \mathbf{1}, 0, 1]}_{\omega} \rightarrow \underbrace{[0, 1, \mathbf{0}, 0, 1]}_{\omega'}$$

2) Decomposition: One molecule ω tries to split into two, ω'_1 and ω'_2. The resultant molecules have great perturbations from the original one and thus, ω'_1 and ω'_2 are quite different from ω. To do this, we examine every bit of ω. Consider the i-bit $\omega(i)$. If it is equal to one, its value is copied to the same position of ω'_1 (i.e. $\omega'_1(i)$) and $\omega'_2(i)$ is set at random. Otherwise, the value of $\omega(i)$ is copied to $\omega'_2(i)$ and $\omega'_1(i)$ is set at random. In this way, there are also great differences between ω'_1 and ω'_2. This seems to place search "seeds" in two new and different regions of the solution space and thus increases the exploration ability of CRO. For example,

$$\underbrace{[\underline{0}, \mathbf{1}, \mathbf{1}, \underline{0}, \mathbf{1}]}_{\omega} \rightarrow \underbrace{[1, \mathbf{1}, \mathbf{1}, 0, \mathbf{1}]}_{\omega'_1} \text{ AND } \underbrace{[\underline{0}, 1, 0, \underline{0}, 1]}_{\omega'_2}$$

3) Inter-molecular Ineffective Collision: Two molecules, ω_1 and ω_2, collide with each other. Two new solutions, ω'_1 and ω'_2, are produced by adding small perturbations to ω_1 and ω_2, respectively. To do this, we simply apply the mechanism used for the on-wall ineffective collision to both ω_1 and ω_2 separately. For example,

$$\underbrace{[0, 1, 1, \mathbf{0}, 1]}_{\omega_1} \text{ AND } \underbrace{[\underline{1}, 0, 1, 0, 1]}_{\omega_2} \rightarrow$$
$$\underbrace{[0, 1, 0, \mathbf{1}, 1]}_{\omega'_1} \text{ AND } \underbrace{[\underline{0}, 0, 1, 0, 1]}_{\omega'_2}$$

4) Synthesis: This tries to combine two molecule ω_1 and ω_2 into a new one ω'. ω' should be quite different from ω_1 and ω_2 when compared with the ineffective collisions. We adopt "exclusive or" to ω_1 and ω_2 to produce ω'. For example,

$$\underbrace{[0, 1, 1, 0, 1]}_{\omega_1} \oplus \underbrace{[1, 0, 1, 0, 1]}_{\omega_2} \rightarrow \underbrace{[1, 1, 0, 0, 0]}_{\omega'}$$

D. Algorithm Outline

As mentioned in Section III-A, each solution has two representations, i.e. the matrix and vector forms. The former contains problem-specific information (e.g. how many channels are allocated to a user) and thus a solution in this form can be used to evaluate the objective function and test against the constraints. The latter is deduced from the former by removing some problem-specific information and it is specially designed for the internal use of an algorithm. We define the algorithmic core as the portion of an algorithm for exploring the solution space. The algorithmic core selects solutions from the solution space and the solutions are in turn examined by the problem to give the objective function values and to test for constraint violations. We can basically divide the operators used in the algorithm into two groups. One is related to the problem and it includes $repair$ and the function evaluation operator U. The other group is for the algorithmic core and consists of the four operators for the elementary reactions.

We basically follow the design framework described in [8] to develop a CRO-based algorithm to solve CRSAP. We provide a flow chart of this algorithm in Fig. 2, where the algorithmic core is shown in the dotted box. In the initialization, we create the initial set of molecules with size equal to *PopSize* and their molecular structures are solutions in the matrix representation by randomly setting every bit. We pass each solution to $repair$ to ensure it is feasible. Then the objective function is evaluated and the corresponding values are the *PE* of the molecules. The initial *KE* of every molecule is set to the value of *InitialKE*. In each iteration, we first convert the solutions from the matrix to vector form with the *M2V* operator. Then we decide whether a uni-molecular or an inter-molecular reaction is carried out in the iteration by

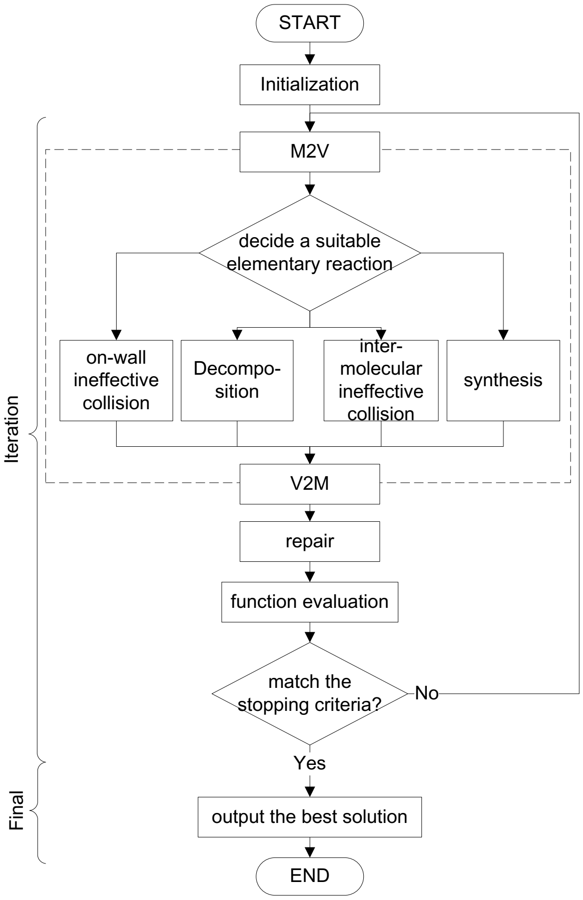

START
Initialization
M2V
decide a suitable elementary reaction
on-wall ineffective collision
Decomposition
inter-molecular ineffective collision
synthesis
V2M
repair
function evaluation
match the stopping criteria?
No
Yes
output the best solution
END
Iteration
Final

Fig. 2. Flow chart of CRO

comparing a random number $h \in [0, 1]$ with *MoleColl*. We select an appropriate subset of molecules to undergo an elementary reaction determined by the decomposition criterion or the synthesis criterion (depending on whether the elementary reaction is uni-molecular or inter-molecular). Next we convert the solutions back to the matrix form with the *V2M* operator. After repairing, the objective functions of the solutions are evaluated. The iteration process continues until a stopping criterion is satisfied. We output the best-so-far solution in the final stage. Due to space limitations, in this paper, we do not describe in details the new solution acceptance rules, the central energy buffer, and the conservation of energy. We only highlight the differences between this CRO implementation and and the general framework in [8] in this paper. For more information about how to build the complete algorithm, interested readers may refer to [8].

IV. Simulation

In this section, we will compare the performance of CRO with three other evolutionary algorithms, i.e. Canonical Genetic Algorithm (CGA) [5], Quantum Genetic Algorithm (QGA) [11], and Particle Swarm Optimization (PSO) [7]. They are chosen because they are adopted to solve a similar problem (which is the same problem addressed in this paper but without Constraint (2)) in [3]. They are all implemented with the same solution-space-reduction (i.e. conversion between matrix and vector representations of solutions) and constraint-violation-removal (i.e. the $repair$ operator) techniques as CRO as described in Section III. We do not consider the heuristic graph coloring approach proposed in [2] for comparison since [3] has already shown that the evolutionary algorithms have much better performance than the graph coloring method. Thus, we expect CRO to also outperform the heuristic approach.

To have fair comparisons of performance over various optimization strategies, we create 10 topologies of primary and secondary users as the set of benchmark problems. Assume that there are 20 secondary users and 20 orthogonal channels, similar to an instance given in [3]. As in [2], we deploy 20 primary users in a region of area 10×10 and every primary user has a constant interference range, equal to 2, for all channels, i.e. $d_p(n, m) = 2, \forall 1 \leq n \leq 20, 1 \leq m \leq 20$. The minimum and maximum interference ranges of the secondary users are 1 and 4, respectively, i.e. $d_{min} = 1$ and $d_{max} = 4$. We follow [2] to determine the interference range of the secondary users $d_s(n, m)$ and set $b_{n,m} = d_s(n, m)^2$, which means that reward is proportional to the service coverage area. By randomly assigning the locations of the primary and secondary users in the given area, we generate 10 sets of channel availability matrix L, channel reward matrix B, and interference constraint matrix C, i.e. 10 problem instances. We compare the performance of the algorithms on these 10 benchmarks.

The parameter values of CRO are given as follows: *PopSize* = 20, *KELossRate* = 0.2, *InitialKE* = 800, *MoleColl* = 0.5, $\alpha = 3000$, and $\beta = 10$. For CGA, QGA, and PSO, the parameters are set according to [3].

We follow the pseudo-codes given in [3] to develop CGA, QGA, and PSO and all simulation codes (including CRO) are programmed in MATLAB. All simulations are run on the same computer with Intel Core 2 Quad Processor Q9650 and 3GB of RAM. As in [3], the stopping criterion is when the maximum number of function evaluations, equal to 6000, is reached (i.e. 300 generations for CGA, QGA, and PSO). Since CRSAP is a maximization problem but CRO is designed for minimization, we make a small modification to the objective function as in [12]. Instead of maximizing $U(A)$, we minimize $U'(A) = 1000 - U(A)$ in the iterations of CRO. We output $U(A_{best}) = 1000 - U'(A_{best})$ in the final stage. For reference, each run of CRO takes approximately 7 seconds, which is sufficiently short when compared with the period of system parameter changes (i.e., changes in L, A, and B) in a static network environment.

We investigate the impact on the number of allowed assigned channels to the users by changing the values of C_{max} from 1 to 20. The smaller C_{max}, the smaller the number of channels allocated to the users, and thus, the smaller the objective function values. This can be seen from Fig. 3. Each data point is the average of 500 simulation runs (50 times for each of the 10 topologies). As before, CRO outperforms the other algorithms dramatically. In Figs. 3(a) and 3(c), we can see the utility increases with C_{max} and the increase saturates after a certain value of C_{max}. This means that the algorithms cannot improve the performance further when more channels can be allocated to the users. In other words, the algorithms can only result in sub-optima with similar utility even when higher flexibility is allowed. In the meantime, CRO saturates at a larger C_{max} than the others, especially in Fig. 3(a). In Fig. 3(a), CRO saturates when C_{max} reaches 17 but the rest saturates after C_{max} equals 6. We can conclude that CRO can utilize the flexibility to improve the solution quality more effectively than the others. However, there is no such obvious saturation trend in Fig. 3(b) [2]. The main reason is that the stopping criterion prevents the algorithms from obtaining better solutions even when they may have the ability to do so after a small number of function evaluations. Nevertheless, CRO still has superior result even in this "per-mature" situation which happens often in practice.

V. Conclusion

Due the severe unbalanced demands of spectrum utilization in the licensed and unlicensed frequency bands, cognitive radio draws great attention as a way for increasing the capacity of wireless networks. One important problem in cognitive radio is CRSAP which attempts to allocate unused frequency channels to the unlicensed users effectively, without affecting any licensed users. We propose a CRO-based algorithm to tackle this problem. CRO is a chemical reaction-inspired metaheuristic for general optimization. With the framework of CRO, we develop several operators so as to make CRO capable of generating good solutions which satisfy the problem requirements and constraints of CRSAP. We also consider the hardware constraint of limiting the maximum number of channels to a user. Simulation results show that CRO outperforms other proposed algorithms for CRSAP dramatically. This suggests that CRO may be more suitable to solve similar problems than other evolutionary computing techniques. In the future, we will develop tailor-made operators for CRO to get even better performance and also compare CRO with a broader class of algorithms. Since CRO is a general-purpose metaheuristic, we will also try to apply it to other resource-constrained wireless networks.

References

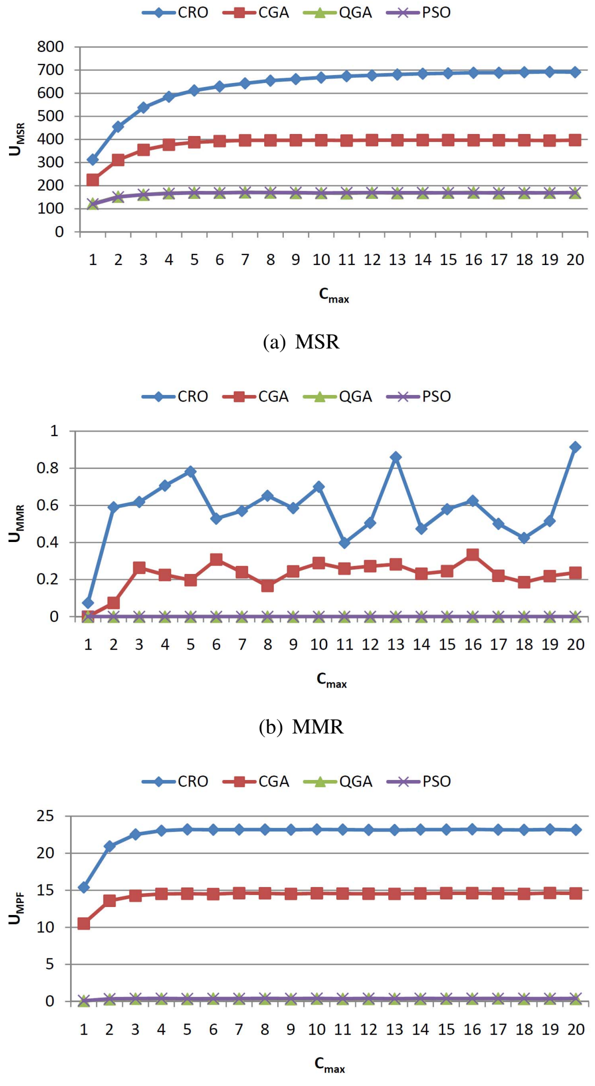
[1] J. Mitola, III, and G. Q. Maguire, Jr., "Cognitive Radio: Making Software Radios More Personal," *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, Aug. 1999.

[2] C. Peng, H. Zheng, and B. Y. Zhao, "Utilization and Fairness in Spectrum Assignment for Opportunistic Spectrum Access," *ACM/Kluwer Mobile Networks and Applications*, vol. 11, no. 4, pp. 555–576, Aug. 2006.

[3] Z. Zhao, Z. Peng, S. Zheng, and J. Shang, "Cognitive Radio Spectrum Allocation using Evolutionary Algorithms," *IEEE Transactions on Wireless Communications*, vol. 8, no. 9, pp. 4421–4425, Sept. 2009.

[4] Q. Zhao, L. Tong, A. Swami, and Y. Chen, "Decentralized Cognitive MAC for Opportunistic Spectrum Access in Ad Hoc Networks: A POMDP Framework," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 3, pp. 589–599, Apr. 2007.

[5] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press, 1975.



(a) MSR

(b) MMR

(c) MPF

Fig. 3. Simulation results of the performance for different number of channels allowed

[6] M. Dorigo and T. Stützle, *Ant Colony Optimization*. Cambridge, MA: MIT Press, 2004.

[7] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, 1995, pp. 1942–1948.

[8] A. Y. S. Lam and V. O. K. Li, "Chemical-Reaction-Inspired Metaheuristic for Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 381–399, Jun. 2010.

[9] H. Gang, L. Liu, Q. Zhang, and M. Xu, "Approximated Matching-Based Spectrum Access Algorithm for Heterogeneous Cognitive Networks," in *Proceedings of the IEEE International Conference on Communications*, 2009, pp. 1–6.

[10] Matrix Indexing:: Matrices and Arrays (MATLAB). [Online]. Available: http://www.mathworks.com/access/helpdesk/help/techdoc/math/f1-85462.html

[11] K.-H. Han and J.-H. Kim, "Quantum-Inspired Evolutionary Algorithm for a Class of Combinatorial Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 6, pp. 580–593, Dec. 2002.

[12] A. Y. S. Lam and V. O. K. Li, "Chemical Reaction Optimization for Population Transition in Peer-to-Peer Live Streaming," in *Proceedings of IEEE Congress on Evolutionary Computation*, 2010.

[2] QGA and PSO are not considered here as they have much worse performance when compared with CRO and CGA.