A Genetic Algorithm for Joint Resource Allocation in Cooperative Cognitive Radio Networks

Wei Yang*, Dongsong Ban*, Weifa Liang† and Wenhua Dou*
*School of Computer Science, National University of Defense Technology, ChangSha, HuNan, 410073, P.R.China
†School of Computer Science, Australian National University, Canberra, ACT 0200, Australia
Email: {weiyang, wliang}@cs.anu.edu.au, dsban@nudt.edu.cn, douwh@vip.sina.com

***Abstract*—Existing literature in Cooperative Cognitive Radio Networks (CCRNs) always assumed a scenario where only one Primary User (PU) and several Secondary Users (SUs) coexist. However, in practice, multi-PUs and multi-SUs always coexist and the number of SUs is usually greater than that of PUs. Under such complex yet real scenarios, we assume that each PU not only allows a set of SUs to access its pre-allocated channel, but can leverage some of these SUs to improve its transmission rate via cooperative technologies. We consider a joint channel allocation and cooperation set partition problem in CCRNs, in which we aim to allocate a channel and assign a cooperation set that consists of several SUs for each PU, such that for a given period of time, the average transmission rates gained by all the users achieve maximum proportional fairness. We formulate the problem as a 0-1 non-linear programming model. Due to its NP-hardness, we propose a suboptimal Centralized Genetic Algorithm (CGA) for the problem. Extensive simulations demonstrate that CGA not only converges rapidly, but is shown to perform as well as 92% of the optimal solution delivered by brutal search, in terms of the fitness that reflects the fairness degree of the transmission performance gained by all the users.**

keywords: cooperative cognitive radio networks; channel allocation; cooperation set partition.

I. Introduction

Recently, a new paradigm, termed *cooperative cognitive radio networks* (CCRNs) is advocated [1]. In such networks, a set of SUs cooperatively relay data for a PU to access its pre-allocated channel, while the PU can leverage all of these SUs to improve the transmission performance, result a 'win-win' solution. Note that almost all existing studies in CCRNs considered a relatively naive scenario [1], [2], where there is only one PU and a few SUs in the entire network. As a matter of fact, multi-PUs and multi-SUs always coexist in a CCRN, the number of SUs is even greater than that of PUs. In this paper, we focus on this complex yet real scenario. We assume that on the one hand, each PU allows a set of SUs to access its pre-allocated channel in a time-division manner. On the other, the PU can leverage some of these SUs to improve its transmission rate via cooperative technologies. Specifically, assume a channel is pre-allocated to a certain PU, all the SUs that require using this channel form a set, namely the cooperation set of this PU. In this paper, we consider a problem on how to jointly allocate a channel and assign the cooperation set for each PU, such that for a given period of time, the average transmission rates gained by all the users achieve maximum 'proportional fairness'. The main contributions is a 0-1 non-linear programming model of this *Joint Channel Allocation and Cooperation Set Partition problem (JCA-CSP)*. Subsequently, due to its NP-hardness, we propose a suboptimal Centralized Genetic Algorithm (CGA) for the problem. The simulation results demonstrate that CGA converges rapidly and is shown to perform as well as 92% of the optimal solution by brutal search under given CCRNs.

The rest of the paper is organized as follows. We present the system model and the problem definition in Section II. Then we make some analysis to the problem in Section III. What followed in Section IV is to describe the details of algorithm CGA. In Section V, we conduct the performance evaluation of the proposed algorithm. We conclude in Section VI.

II. Preliminaries

A. System Model

We consider a CCRN that consists of M PU links $\{PT_i \to PR_i\}_{i=1}^{M}$ and N SU links $\{ST_j \to SR_j\}_{j=1}^{N}$, with $M < N$. We use $PT = \{PT_i | i \in [1, M]\}$ and $ST = \{ST_j | j \in [1, N]\}$ to denote the set of all the primary transmitters and secondary transmitters, respectively. During a given period of time $Tslot$, each PT_i in PT allows other secondary transmitters to access its channel, while each ST_j in ST expects to use a channel owned by a certain primary transmitter. The spectrum of interest is divided into M non-overlapping channels (each with identical bandwidth) that are licensed to M PU links. Unless otherwise specified, we use i, j, and k to index the primary transmitters, the secondary transmitters, and the available channels, respectively. In the following, all definitions and denotations hold for all $i, k \in [1, M]$ and $j \in [1, N]$.

For each PT_i in PT, denote by C_i the index of its pre-allocated channel. We can define a *Network Channel Allocation Matrix*, denoted by $\mathbf{\Theta} = (\theta_{ik})_{M \times M}$, where if $C_i = k$, then $\theta_{ik} = 1$, otherwise $\theta_{ik} = 0$. Since we assume $M < N$, multi secondary transmitters (nodes) within the maximum transmission radius of PT_i may apply to use PT_i's pre-allocated channel together, these nodes constitute a *Potential Cooperation Set* of PT_i, denoted by PCS_i. In other words, the nodes in PCS_i can potentially act as the cooperative relays for PT_i. Then PT_i chooses a subset of nodes from PCS_i, such that the transmission performance of PT_i can be enhanced at most via cooperating with these chosen nodes. The chosen nodes then forms the *Real Cooperation Set* of PT_i, denoted by

RCS_i. Clearly $RCS_i \subseteq PCS_i$. Note that RCS_i can be an $\emptyset$, which means all the nodes in the PCS_i, due to their inferior channel conditions or improper locations, in reality are not helpful to increase the transmission rate of PT_i. Therefore we can define a *Network Potential Cooperation Matrix*, denoted by $\mathbf{A} = (a_{ij})_{M\times N}$, where $a_{ij} = 1$, if $ST_j \in PCS_i$, otherwise $a_{ij} = 0$. Similarly, we introduce a *Network Real Cooperation Matrix*, denoted by $\overline{\mathbf{A}} = (\overline{a_{ij}})_{M\times N}$, where if $\overline{a_{ij}} = 1$, if $ST_j \in RCS_i$, otherwise $\overline{a_{ij}} = 0$.

To facilitate further discussions, we define some notations as follows. g^k_{ij}, g^k_{ii}, g^k_{jj}, and g^k_{ji} denote the channel gains of links $PT_i \to ST_j$, $PT_i \to PR_i$, $ST_j \to SR_j$, and $ST_j \to PR_i$, using channel k, respectively. N_0 denotes the noise power spectrum density, which is assumed to be a constant. P^p_i and P^s_j denote the transmission power of PT_i and ST_j, respectively, which remain unchanged during the given period of time $Tslot$. Without loss of generality, in the following we assume $Tslot = 1$.

Given PT_i's pre-allocated channel, e.g., $C_i = k$, PCS_i and RCS_i, the *Cooperation Process* among PT_i and the nodes in PCS_i is as follows. The unit time-slot is first divided into two sub-slots φ_i and $1 - \varphi_i$, where $\varphi_i \in (0, 1)$.

(1) Sub-slot φ_i. Clearly, either $RCS_i \neq \emptyset$ or $RCS_i = \emptyset$.

1) If $RCS_i \neq \emptyset$, φ_i is further divided into two sub-slots $\varphi_i\gamma_i$ and $\varphi_i(1-\gamma_i)$, where $\gamma_i \in (0, 1)$.

i) $\varphi_i\gamma_i$: PT_i broadcasts the data to all the nodes in RCS_i. To ensure that all the nodes in RCS_i decode the data successfully, the transmission rate R^p_{i1} in $\varphi_i\gamma_i$ is dominated by the link with the lowest channel gain among that of all links between PT_i and each of the nodes in RCS_i, which is expressed as follows.

$$R^p_{i1} = \log_2(1 + \frac{\arg\min \overline{a_{ij}} g^k_{ij} P^p_i}{N_0}) \qquad (1)$$

ii) $\varphi_i(1-\gamma_i)$: all the nodes in RCS_i cooperatively relay data for PT_i to PR_i. Assume PR_i exploits maximum ratio combining before decoding the signals [3], the accumulated Signal to Noise Ratio (SNR) at PR_i therefore equals to the sum of SNRs from all the transmitters. Thus, the transmission rate R^p_{i2} in $\varphi_i(1-\gamma_i)$ is expressed as follows.

$$R^p_{i2} = \log_2(1 + \frac{P^p_i g^k_{ii}}{N_0} + \sum\nolimits_{j=1}^{N} \frac{\overline{a_{ij}} P^s_j g^k_{ji}}{N_0}). \qquad (2)$$

Following Eqs. (1) and (2), the average transmission rate of PT_i when $RCS_i \neq \emptyset$ is given by

$$R^p_i(RCS_i)_{RCS_i\neq\emptyset} = \min\{\varphi_i\gamma_i R^p_{i1}, \varphi_i(1-\gamma_i)R^p_{i2}\}. \qquad (3)$$

2) If $RCS_i = \emptyset$, PT_i transmits directly to PR_i. Thus, the average transmission rate of PT_i when $RCS_i = \emptyset$ is given by

$$R^p_i(RCS_i)_{RCS_i=\emptyset} = \varphi_i \log_2(1 + \frac{P^p_i g^k_{ii}}{N_0}). \qquad (4)$$

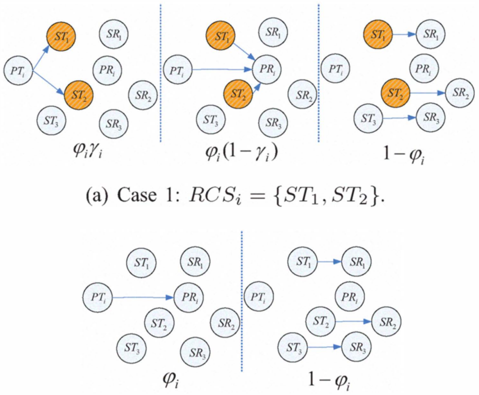
Apparently, if PT_i can figure out a RCS_i such that $\max R^p_i(RCS_i)_{RCS_i\neq\emptyset} > R^p_i(RCS_i)_{RCS_i=\emptyset}$, it would rather transmit cooperatively with the nodes in RCS_i than transmit directly by itself. Therefore, the average transmission rate of PT_i in this unit time-slot is expressed as follows.

$$R^p_i = \max\{\max R^p_i(RCS_i)_{RCS_i\neq\emptyset}, R^p_i(RCS_i)_{RCS_i=\emptyset}\} \qquad (5)$$

(2) Subslot $1 - \varphi_i$. Each node in PCS_i occupies a fraction of slot $1 - \varphi_i$ to use channel k. Specifically, for each ST_j in PCS_i, its share of time is $t_j = (1-\varphi_i)\omega_j / \sum_{l=1}^{N} a_{il}\omega_l$, where ω_j is its weight. In practice, this weight can be its bidding price to compete for this channel. Thus, the average transmission rate of ST_j in this unit time-slot is given by

$$R^s_j = (1-\varphi_i)\frac{\omega_j}{\sum_{l=1}^{N} a_{il}\omega_l} \log_2(1 + \frac{P^s_j g^k_{jj}}{N_0}). \qquad (6)$$

To this end, we have defined the cooperation process among PT_i and the nodes in PCS_i. Fig. 1 gives an example, in which we assume that $PCS_i = \{ST_1, ST_2, ST_3\}$. In case 1, as shown in Fig. 1(a), $RCS_i = \{ST_1, ST_2\} \neq \emptyset$, ST_1, ST_2 cooperatively relay data with PT_i to PR_i, while in case 2, as shown in Fig. 1(b), $RCS_i = \emptyset$, then PT_i transmits directly to PR_i by itself.



(a) Case 1: $RCS_i = \{ST_1, ST_2\}$.

(b) Case 2: $RCS_i = \emptyset$.

Fig. 1. An example of the cooperation process among PT_i and the secondary transmitters in PCS_i.

B. Problem Definition

Definition 1: Joint Channel Allocation and Cooperation Set Partitioning problem (JCA-CSP). Given a CCRN defined above. The JCA-CSP is to allocate a channel, assign a potential cooperation set, and identify a real cooperation set for each PT_i in PT, such that for a given unit time, the transmission rates gained by all the users achieve maximum proportional fairness. The problem is formalized as follows.

$$\textbf{Determine}: \mathbf{A} = (a_{ij})_{M\times N}, \overline{\mathbf{A}} = (\overline{a_{ij}})_{M\times N},$$
$$\mathbf{\Theta} = (\theta_{ik})_{M\times M}, \mathbf{\Phi} = (\varphi_i)_{1\times M}, \mathrm{H} = (\gamma_i)_{1\times M}$$
$$\textbf{Maximize}: \sum\nolimits_{i=1}^{M} \ln(R^p_i) + \sum\nolimits_{j=1}^{N} \ln(R^s_j) \qquad (7)$$
$$\textbf{Subject to}: N - M + 1 \geq \sum\nolimits_{j=1}^{N} a_{ij} \geq 1, \forall i \in [1, M] \qquad (8)$$
$$\sum\nolimits_{i=1}^{M} a_{ij} = 1, \forall j \in [1, N] \qquad (9)$$
$$\sum\nolimits_{i=1}^{M} \theta_{ik} = 1, \forall k \in [1, M] \qquad (10)$$
$$\sum\nolimits_{k=1}^{M} \theta_{ik} = 1, \forall i \in [1, M] \qquad (11)$$
$$a_{ij}, \overline{a_{ij}}, \theta_{ik} \in \{0, 1\}, \forall i, k \in [1, M], \forall j \in [1, N] \qquad (12)$$
$$0 < \varphi_i < 1, 0 \leq \gamma_i < 1, \forall i \in [1, M] \qquad (13)$$
$$\overline{a_{ij}} \leq a_{ij}, \forall i \in [1, M], \forall j \in [1, N] \qquad (14)$$

The objective function (7) aims to maximize the proportional fairness among the transmission rates gained by all the users. The essence of this criteria is that for any other feasible solution $\mathbf{A}', \overline{\mathbf{A}}', \mathbf{\Theta}', \mathbf{\Phi}', \mathrm{H}'$, and the corresponding $R_i^{p\prime}$ and $R_j^{s\prime}$, the aggregate proportional changes in transmission rates is zero or negative [4], i.e., $\sum_{i=1}^{M}(R_i^{p\prime} - R_i^p)/R_i^p + \sum_{j=1}^{N}(R_j^{s\prime} - R_j^s)/R_j^s \leq 0$. Constraint (8) guarantees that at least one secondary transmitter is affiliated with each primary transmitter, while constraint (9) ensures that each secondary transmitter can finally access to a certain channel that belongs to a certain primary transmitter. Constraints (8) and (9) further indicate that ST should be partitioned into M disjoint potential cooperation sets, i.e., $\bigcap_{i=1}^{M} PCS_i = \emptyset$ and $\bigcup_{i=1}^{M} PCS_i = ST$. Constraints (10) and (11) guarantee the one-to-one matching between the available channels and primary transmitters. The values of the elements in matrixes $\mathbf{A}, \overline{\mathbf{A}}, \mathbf{\Theta}, \mathbf{\Phi}, \mathrm{H}$ are constrained by (12) and (13). Constraint (14) requires that for each PT_i in PT, $RCS_i \subseteq PCS_i$ must be satisfied. Obviously, this is a 0-1 non-linear programming problem. In the following, we shown some properties of the problem and prove its NP-hardness.

III. Problem Analysis

Theorem 1: For each PT_i in PT, given its pre-allocated channel $C_i = k$ and its potential cooperation set PCS_i, there exists an algorithm with time complexity $O(n^2)$ that can find RCS_i and the corresponding φ_i and γ_i, such that $f_i(C_i, PT_i, PCS_i) = \ln(R_i^p) + \sum_{ST_j \in PCS_i} \ln(R_j^s)$ is maximized, where $n = |PCS_i|$.

Proof: (1) Firstly, we prove the existence of φ_i and γ_i. Clearly, either $RCS_i \neq \emptyset$ or $RCS_i = \emptyset$ holds.

1) If $RCS_i \neq \emptyset$, PT_i cooperatively transmit with the secondary transmitters in RCS_i using its pre-allocated channel, e.g., $C_i = k$. Thus, by Eqs. (1)(2)(3)(6), we have $f_i(C_i, PT_i, PCS_i) = \ln(\min\{\varphi_i\gamma_i R_{i1}^p, \varphi_i(1-\gamma_i)R_{i2}^p\}) + \sum_{ST_j \in PCS_i} \ln(\frac{(1-\varphi_i)\omega_j}{\sum_{l=1}^{N} a_{il}\omega_l} \log_2(1 + \frac{P_j^s g_{jj}^k}{N_0}))$. Since we assume that the transmission power of all the users are constants, R_{i1}^p and R_{i2}^p are constants too. Then we observe that $\varphi_i\gamma_i R_{i1}^p$ and $\varphi_i(1-\gamma_i)R_{i2}^p$ are the increasing and decreasing functions of γ_i respectively, thus $\ln(\min\{\varphi_i\gamma_i R_{i1}^p, \varphi_i(1-\gamma_i)R_{i2}^p\})$ achieves its maximum as $\varphi_i\gamma_i R_{i1}^p = \varphi_i(1-\gamma_i)R_{i2}^p$, that is $\gamma = \gamma_i^* = R_{i2}^p/(R_{i1}^p + R_{i2}^p)$. For each ST_j in PCS_i, for simplicity, we denote $\frac{\omega_j}{\sum_{l=1}^{N} a_{il}\omega_l} \log_2(1 + \frac{P_j^s g_{jj}^k}{N_0})$ by Γ_j, then we have $f_i = \ln(\gamma_i^* \varphi_i R_{i1}^p) + \sum_{ST_j \in PCS_i} \ln((1-\varphi_i)\Gamma_{3j})$. Note that it is a concave function with respect to φ_i on $\varphi_i \in (0, 1)$, f_i reaches its maximum $\ln(\gamma_i^* \varphi_i^* R_{i1}^p) + \sum_{ST_j \in PCS_i} \ln((1-\varphi_i^*)\Gamma_j)$ at $\varphi_i = \varphi_i^* = (\sum_{l=1}^{N} a_{il})/(\sum_{l=1}^{N} a_{il} + 1)$.

2) If $RCS_i = \emptyset$, the primary transmitter does not cooperate with other secondary transmitters. Clearly, $\gamma_i = 0$. By Eqs. (4) and (6), if we denote $\log_2(1 + \frac{P_i^p g_{ii}^k}{N_0})$ by a constant Γ_i, we have $f_i = \ln(\varphi_i \Gamma_i) + \sum_{ST_j \in PCS_i} \ln((1-\varphi_i)\Gamma_j)$. Obviously, it is a concave function with respect to φ_i on $\varphi_i \in (0, 1)$. As a result, f_i reaches its maximum $\ln(\Gamma_i \varphi_i^*) + \sum_{ST_j \in PCS_i} \ln((1-\varphi_i^*)\Gamma_j)$ at $\varphi_i = \varphi_i^* = (\sum_{l=1}^{N} a_{il})/(\sum_{l=1}^{N} a_{il} + 1)$.

Therefore, by 1) and 2), to maximize f_i, we have

$$\begin{cases} \gamma_i = \gamma_i^* = \frac{R_{i2}^p}{R_{i1}^p + R_{i2}^p}, & \varphi_i = \varphi_i^* = \frac{\sum_{l=1}^{N} a_{il}}{\sum_{l=1}^{N} a_{il} + 1}, \quad (RCS_i \neq \emptyset) \\ \gamma_i = 0, & \varphi_i = \varphi_i^* = \frac{\sum_{l=1}^{N} a_{il}}{\sum_{l=1}^{N} a_{il} + 1}, \quad (RCS_i = \emptyset) \end{cases}$$

Clearly, φ_i^* are identical for both cases.

(2) Then we describe the idea of the algorithm to find such RCS_i that maximizes f_i. Note that if $RCS_i = \emptyset$, $R_i^p(RCS_i)_{RCS_i = \emptyset} = \varphi_i^* \Gamma_i$. The idea behind the algorithm is try to find a non-empty set such that $\max R_i^p(RCS_i)_{RCS_i \neq \emptyset} > \varphi_i^* \Gamma_i$, otherwise (if we fail), let $RCS_i = \emptyset$. We first find a set $PCS_i' = \{ST_j | ST_j \in PCS_i, d(ST_j, PR_i) < D_j^p\}$, i.e., we filter out the secondary transmitters that is geographically unable to reach PR_j. Then we observe that for each ST_j in PCS_i', if it is the first node that is added to RCS_i, then we can calculate out R_{i1}. For any other ST_l in PCS_i', if $g_{il}^k > g_{ij}^k$, ST_l can be added to RCS_i without changing the value of R_{i1}. We thus add all the nodes with $g_{il}^k > g_{ij}^k$ to RCS_i in order to maximize R_{i2}. Thus $R_i^p(RCS_i)_{RCS_i \neq \emptyset} = \varphi_i^* R_{i1}^p R_{i2}^p/(R_{i1}^p + R_{i2}^p) = \varphi_i^*/(1/R_{i1} + 1/R_{i2})$ is maximized. As a result we obtain a real cooperation set when g_{ij}^k is fixed, denoted by $RCS_i(g_{ij}^k)$, that maximizes $R_i^p(RCS_i(g_{ij}^k))_{RCS_i \neq \emptyset}$. With the above method, we can find a total of $|PCS_i'|$ different real cooperation sets and their corresponding maxima. If $\max\{R_i^p(RCS_i(g_{ij}^k)) | ST_j \in PCS_i'\} > \varphi_i^* \Gamma_i$, then let $RCS_i = \arg\max\{R_i^p(RCS_i(g_{ij}^k)) | ST_j \in PCS_i'\}$, otherwise let $RCS_i = \emptyset$. Clearly, the algorithm takes at most $O(n^2)$ time, where $n = |PCS_i|$.

Thus, Theorem 1 is proved by (1) and (2). ■

Theorem 2: Given the network potential cooperation matrix $\mathbf{A}$ and network channel allocation matrix $\mathbf{\Theta}$ that satisfy constraint (7)- (11), there exist $\overline{\mathbf{A}}$, $\mathbf{\Phi}$, and H, such that the objective function of JCA-CSP, i.e., $\sum_{i=1}^{M} \ln(R_i^p) + \sum_{j=1}^{N} \ln(R_j^s)$, is maximized.

Proof: By the definition of $\mathbf{A}$ and $\mathbf{\Theta}$, we can obtain PCS_i and C_i for each PT_i in PT. Note that for each PT_i, by Theorem 1, RCS_i, φ_i, and γ_i can be found such that $f_i(C_i, PT_i, PCS_i)$ is maximized. Note that there are totally M primary transmitters, correspondingly, we can obtain $\overline{\mathbf{A}}$, $\mathbf{\Phi}$, and H, such that $\sum_{i=1}^{M} \ln(R_i^p) + \sum_{j=1}^{N} \ln(R_j^s) = \sum_{i=1}^{M}(\ln(R_i^p) + \sum_{ST_j \in PCS_i} \ln(R_j^s)) = \sum_{i=1}^{M} f_i(C_i, PT_i, PCS_i)$ is maximized too. ■

Theorem 3: The JCA-CSP is NP-hard.

Proof: To prove the NP-hardness of JCA-CSP, we begin by defining a decision problem corresponding to JCA-CSP, referred to as JCA-CSP′, and then prove that JCA-CSP′ is NP-complete by reducing it to the 3-Dimensional Matching (3DM) problem, which is known to be NP-complete [5]. Therefore we can conclude that the optimization problem JCA-CSP is NP-hard. (1) JCA-CSP′. Denote by C, PT, and ST the set of the available channels, primary transmitters, and secondary transmitters, respectively, where $M = |PT| = |C| < |ST| = N$. We first partition ST into M disjoint subsets, denoted by $ST(1), ST(2), \cdots$, and $ST(M)$. Then let Λ be a subset of $C \times PT \times ST$, which consists of triples (x_l, y_l, z_l), where $x_l \in C$ is a channel, $y_l \in PT$ is a primary transmitter, $z_l \subset ST$ is a subset of ST. For example, if $y_l = PT_i$, then let $C_i = x_l$, and $PCS_i = z_l$, by Theorem 1 we can define a function of for the triple as $f'((x_l, y_l, z_l)) = \max f_i(C_i, PT_i, PCS_i)$, provided that $PT_i = y_l$. JCA-CSP′ determines whether there exists a three dimensional matching $\Lambda' \subseteq \Lambda$, such that $|\Lambda'| = |M|$, $\sum_{i=1}^{M} f'((x_i, y_i, z_i)) \geq \lambda$, where $(x_i, y_i, z_i) \in \Lambda'$, λ is a given real number, meanwhile for any two distinct triples (x_1, y_1, z_1) and (x_2, y_2, z_2) in Λ', $x_1 \neq x_2$, $y_1 \neq y_2$, and $z_1 \neq z_2$ must be met. (2) The 3DM problem. Let C, PT, and ST be finite, disjoint sets, where $|C| = |PT| = |ST| = M$. Let Λ be a subset of $C \times PT \times ST$, which consists of triples (x_l, y_l, z_l) such that $x_l \in C$, $y_l \in PT$, and $z_l \in ST$. The 3DM problem determines whether there exists a three dimensional matching $\Lambda' \subseteq \Lambda$, such that $|\Lambda'| = |M|$, and for any two distinct triples (x_1, y_1, z_1) and (x_2, y_2, z_2) in Λ', $x_1 \neq x_2$, $y_1 \neq y_2$, and $z_1 \neq z_2$ must be met. (3) We finally reduce JCA-CSP′ to the 3DM problem as follows. 1) Decrease the number of the nodes in ST to M, then in JCA-CSP′, we have $|C| = |PT| = |ST| = M$. Since $|ST| = M$, the third part in the triple of JCA-CSP′ has only one element. 2) Eliminate the constraints to $\sum_{i=1}^{M} f'((x_i, y_i, z_i))$, i.e., let $\lambda = -\infty$. We now have reduced JCA-CSP′ to the 3DM problem. Since the latter one is NP-complete, JCA-CSP′ is NP-complete too. Thus its optimization problem JCA-CSP is NP-hard. ■

IV. Centralize Genetic Algorithm

The Genetic Algorithm (GA) provides an efficient approach to solve the NP-hard problems [6], [7]. The standard GA is based on Darwinism where each possible solution of the optimization problem is modeled as an individual of a species that is characterized by its chromosome. A colony of individuals is considered as a generation of the species in its evolutionary chain. The GA iteratively evolves its generation via three essential steps: (1) selecting genetically superior individuals as parents from the current generation, (2) mating them to produce their offspring, and (3) occasionally mutating the offspring to avoid falling into local optima. The superiority of an individual is measured by its fitness, which is usually a function of the objective to optimize. To solve the JCA-CSP, in this section, we are inspired to propose a Centralized Genetic Algorithm, which is referred to as CGA.

Definition 2: Joint Chromosome, Colony. Denote by $\mathbf{JC} = [\mathbf{A}, \mathbf{B}]_{M \times (N+1)}$ a joint chromosome that uniquely characterizes an individual. As shown in Fig. 2, it is composed of two parts: the network potential cooperation matrix $\mathbf{A} = (a_{ij})_{M \times N}$ and a vector $\mathbf{B} = (\beta_i)_{M \times 1}$, where $\mathbf{B}$ is the vector format of the network channel allocation matrix $\mathbf{\Theta} = (\theta_{ik})_{M \times M}$, $\forall i, k \in [1, M]$, $\beta_i = k$ iff $\theta_{ik} = 1$, $\beta_i \neq k$ iff $\theta_{ik} = 0$. It is preferable to use the vector $\mathbf{B}$ instead of matrix $\mathbf{\Theta}$, since from the perspective of data structure, it is easier to deal with a vector rather than a matrix. In the following, for convenience, we use the term 'joint chromosome' instead of individual. Then a number of joint chromosomes can form a colony, denoted by CN.

Definition 3 (Chromosome Mask): Initially, CGA has to generate an original colony that consists of several legal joint chromosomes without violating the network topology. To do so, we define a Chromosome Mask, denoted by a binary matrix $\mathbf{CM} = (m_{ij})_{M \times N}$, to act as a pool to assist generating legal joint chromosomes, where $m_{ij} = 1$ on condition that the Euclidean distance between PT_i and ST_j is no more than the maximum transmission radius of PT_i, i.e., $d(PT_i, ST_j) \leq D_i^p$, otherwise $m_{ij} = 0$. In other words, $m_{ij} = 1$ indicates that ST_j is geographically qualified to use PT_i's channel and can be added to PCS_i.

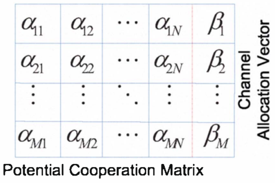

$$\begin{array}{ccccc} \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1N} & \beta_1 \\ \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2N} & \beta_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_{M1} & \alpha_{M2} & \cdots & \alpha_{MN} & \beta_M \end{array}$$

Fig. 2. A joint chromosome.

Definition 4 (Fitness function): To measure the superiority of a joint chromosome $\mathbf{JC}$, by Theorems 1 and 2, we can define its fitness function as follows.

$$\begin{aligned} F(\mathbf{JC}) = F([\mathbf{A}, \mathbf{B}]) &= \sum\nolimits_{i=1}^{M} \max f_i(C_i, PT_i, PCS_i) \\ &= \sum\nolimits_{i=1}^{M} (\max(\ln(R_i^p) + \sum\nolimits_{ST_j \in PCS_i} \ln(R_j^s))) \end{aligned}$$

We now present algorithm CGA, which comprises four steps.

Step 1: Colony Initialization. Initially, we randomly generates an original colony CN that consists of Q joint chromosomes. If using the denotations in MATLAB, each $\mathbf{JC} = [\mathbf{A}, \mathbf{B}]$ in CN should satisfy the constraints as follows.

$$\begin{aligned} &1.\quad \mathbf{BITAND}(\mathbf{CM}, \mathbf{A}) = \mathbf{A} \\ &2.\quad \mathbf{SUM}(\mathbf{A}(i, :)) \geq 1, \forall i \in [1, M] \\ &3.\quad \mathbf{SUM}(\mathbf{A}(:, j)) = 1, \forall j \in [1, N] \\ &4.\quad \mathbf{B}(i) \neq \mathbf{B}(k), \forall i \neq k \in [1, M] \end{aligned} \qquad (15)$$

Constraint (15)-1 guarantees that $\mathbf{A}$ does not violate the network topology, where function $\mathbf{BITAND}$ returns the bitwise AND of $\mathbf{CM}$ and $\mathbf{A}$. Constraints (15)-2,3,4 corresponds to constraints (8)-(11), where functions $\mathbf{SUM}(\mathbf{A}(i, :))$ and $\mathbf{SUM}(\mathbf{A}(:, j))$ return the sum of the ith column and the jth row of $\mathbf{A}$, respectively, $\mathbf{B}(i)$ denotes the ith element in $\mathbf{B}$.

We now show how a joint chromosome that satisfies constraint (15) can be generated. Note that we can easily obtain $\mathbf{B}$ of a $\mathbf{JC}$ by randomly permutating vector $[1, 2, \cdots, M]'$, which ensures that all the elements in $\mathbf{B}$ are distinct, so that constraint (15)-4 is met. However, it is no easy to obtain $\mathbf{A}$ of a $\mathbf{JC}$ that must satisfy constraints (15)-1,2,3 simultaneously. To do so, we devise an algorithm, which is described as follows. (1) Initially, let $\mathbf{A}$ be a zero matrix. We rank the columns in $\mathbf{CM}$ according to the total number of '1's each column owns. Columns with more '1's are assigned higher ranks. Then for each column ranked from the lowest to the highest in $\mathbf{CM}$, we randomly find a position in this column that has an '1'. Then at the identical position in $\mathbf{A}$, if there is no other '1's in the row to which this position belongs, we assign this position in

$\mathbf{A}$ an '1'. (2) Note that M rows in $\mathbf{A}$ are assigned '1's after the previous step, however, there are still $N-M$ rows in $\mathbf{A}$ that remain un-assigned, i.e., are still zero vectors. According to the indexes of these rows in $\mathbf{A}$, we can find $N-M$ corresponding rows in $\mathbf{CM}$. We then randomly choose a position that has an '1' in each of these $N-M$ rows in $\mathbf{CM}$. Finally, assume these positions in $\mathbf{CM}$ are chosen, we assign the identical chosen positions in $\mathbf{A}$ with '1's, accordingly.

Step 2: Protective Crossover. Then we use a protective crossover strategy for parent joint chromosomes in CN to generate their offspring. Specifically, as shown in Fig. 3, assume $\mathbf{JC} = [\mathbf{A}, \mathbf{B}]$ and $\mathbf{JC}' = [\mathbf{A}', \mathbf{B}']$ are now selected as parents. (1) We randomly select two rows in $\mathbf{A}$ and $\mathbf{A}'$ with identical index, e.g., L_1, and then mating $\mathbf{A}$ and $\mathbf{A}'$ by exchanging these two rows. Denote by $\widetilde{\mathbf{A}}$ and $\widetilde{\mathbf{A}'}$ the result matrixes, we further check whether the result satisfy constraints (15)-2 and (15)-3. If the constraints are met, the crossover is allowed, otherwise $\mathbf{A}$ and $\mathbf{A}'$ maintain unchanged. (2) Similarly, we randomly select two elements in $\mathbf{B}$ and $\mathbf{B}'$ with identical index, e.g., L_2, and exchange them at first. We observe that the result vectors, denoted by $\widetilde{\mathbf{B}}$ and $\widetilde{\mathbf{B}'}$, may violate constraint (15)-4, since there may exist two identical elements in each of them. We further perform a 'protective crossover' operation for $\widetilde{\mathbf{B}}$ and $\widetilde{\mathbf{B}'}$, respectively. Take $\widetilde{\mathbf{B}}$ as an example, if $\exists x \neq L_2 \in [1, M]$ in $\widetilde{\mathbf{B}}$ such that $\beta_x = \beta'_{L_2} = \mu_2$, then let $\beta_x = \beta_{L_2} = \mu_1$, so that the elements in $\widetilde{\mathbf{B}}$ are distinct. As a result of (1) and (2), we obtain two offspring, denoted by $\widetilde{\mathbf{JC}} = [\widetilde{\mathbf{A}}, \widetilde{\mathbf{B}}]$ and $\widetilde{\mathbf{JC}'} = [\widetilde{\mathbf{A}'}, \widetilde{\mathbf{B}'}]$. By this strategy, we generate a new colony CN' that consists of $2Q$ offspring.

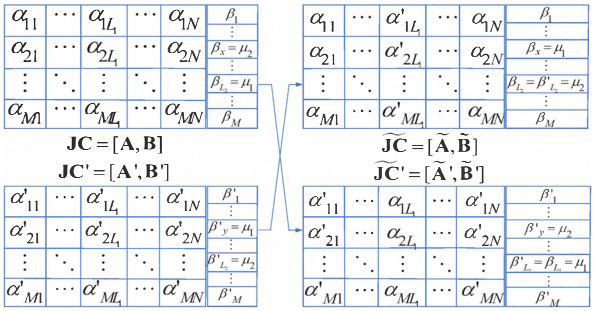

α_{11} ⋯ α_{1L_1} ⋯ α_{1N} β_1
α_{21} ⋯ α_{2L_1} ⋯ α_{2N} $\beta_x=\mu_2$
$\beta_{L_2}=\mu_1$
α_{M1} ⋯ α_{ML_1} ⋯ α_{MN} β_M
$\mathbf{JC}=[\mathbf{A},\mathbf{B}]$
$\mathbf{JC}'=[\mathbf{A}',\mathbf{B}']$
α'_{11} ⋯ α'_{1L_1} ⋯ α'_{1N} β'_1
α'_{21} ⋯ α'_{2L_1} ⋯ α'_{2N} $\beta'_y=\mu_1$
$\beta'_{L_2}=\mu_2$
α'_{M1} ⋯ α'_{ML_1} ⋯ α'_{MN} β'_M
α_{11} ⋯ α'_{1L_1} ⋯ α_{1N} β_1
α_{21} ⋯ α'_{2L_1} ⋯ α_{2N} $\beta_x=\mu_1$
$\beta_{L_2}=\beta'_{L_2}=\mu_2$
α_{M1} ⋯ α'_{ML_1} ⋯ α_{MN} β_M
$\widetilde{\mathbf{JC}}=[\widetilde{\mathbf{A}},\widetilde{\mathbf{B}}]$
$\widetilde{\mathbf{JC}'}=[\widetilde{\mathbf{A}'},\widetilde{\mathbf{B}'}]$
α'_{11} ⋯ α_{1L_1} ⋯ α'_{1N} β'_1
α'_{21} ⋯ α_{2L_1} ⋯ α'_{2N} $\beta'_y=\mu_2$
$\beta'_{L_2}=\beta_{L_2}=\mu_1$
α'_{M1} ⋯ α_{ML_1} ⋯ α'_{MN} β'_M

Fig. 3. Protective crossover for $\mathbf{JC}$ and $\mathbf{JC}'$.

Step 3: Protective Mutation. Next, we use a protective mutation strategy to mutate each offspring in CN'. Specifically, for each offspring $\mathbf{JC} = [\mathbf{A}, \mathbf{B}]$ in CN', (1) we mutate each column in $\mathbf{A}$ with a probability P_m. If a column is chosen to mutate, we reassign the only '1' in this column to another position randomly. After performing mutation to all the columns, we further check whether the result matrix meets constraint (15). If so, the mutation to all the columns are allowed, otherwise $\mathbf{A}$ maintain unchanged. (2) As for $\mathbf{B}$, we perform protective mutation for each of its element with P_m too. e.g., if $\beta_y = \mu_1$ is chosen to mutate, then β_y is reassigned a value that is randomly drawn from $[1, M]$, e.g., we now have $\beta_y = \mu_2$. The 'protective' means that if $\exists x \neq y$ such that $\beta_x = \mu_2$, then we let $\beta_x = \mu_1$ so as to satisfy constraint (15)-4. By this strategy, all the offspring in CN' are mutated, as a result, we have a new colony CN''.

Step 4: Selection. Given colonies CN'' and CN, i.e., $3Q$ joint chromosomes, we now select the best Q joint chromosomes in terms of their fitness from $CN'' \cup CN$. The chosen joint chromosomes then form an original colony for the next evolution process. We terminate the algorithm as the changing rate of the average fitness of all the joint chromosomes in the original colony becomes zero, otherwise go to Step 2 and continue the evolution process.

V. Performance Evaluation

In this section, we evaluate CGA against algorithm Brutal Search, referred to as BS. Above all, BS is described as follows. First, we partition ST (N secondary transmitters) into M disjoint sets. Denote by $S(N, M)$ the total number of combination for such partition, which is also known as the Stirling number of the second kind [8] and can be iteratively calculated as $S(N, M) = S(N-1, M-1) + M \times S(N-1, M)$. By calculation, we have $S(N, M) = (1/M!)\sum_{i=1}^{M}(-1)^{M-i}C_M^i(i)^N$. Then, match these M disjoint sets with M primary transmitters in PT, we obtain $M!S(N, M)$ different matchings. Finally, match each of these matchings with M available channels, we finally have $M!M!S(N, M) = M!\sum_{i=1}^{M}(-1)^{M-i}C_M^i(i)^N$ possible solutions for the optimization problem JCA-CSP. BS then checks these possible solutions one by one and selects an optimal joint chromosome that has the maximum fitness. The network size of a CCRN is defined by a triple $NS(M, N, D)$, where $M = |PT|$, $N = |ST|$, and D is the maximum transmission radius of the user (assume identical for all users). For each topology under a given $NS(M, N, D)$, we generate a uniform distribution for M primary transmitters and N secondary transmitters in a unit square region. For each transmitter, we randomly locate its corresponding receiver within the maximum transmission radius D. $\forall i \in [1, M], \forall j \in [1, N]$, we assume $P_i^p/N_0 = P_j^s/N_0 = 10$, the weight of ST_j, i.e., ω_j, is randomly drawn from [1,10]. Furthermore $g_{ij}^k = d_{ij}^{-\gamma}|\xi_{ij}^k|^2$, where $d_{ij}^{-\gamma}$ is the path loss between PT_i and ST_j with path loss exponent $\gamma = 2$, ξ_{ij}^k is a Gaussian random variable with normal distribution $\mathcal{N}(0, 1)$ [9]. The mutation probability in the CGA is $P_m = 0.01$, the number of joint chromosomes in the original colony is $Q = 30$. All simulations are done by using MATLAB.

A. CGA *vs. Brutal Search* (BS)

We begin the simulation by a demonstration, referred to as DEMO, to verify the correctness of CGA, in which we randomly generate a topology under the network size $NS(4, 10, 0.7)$. Fig. 4 is channel allocation and cooperation set partition result delivered by CGA, where the circle nodes denote the primary transmitters, the square ones denote the secondary transmitters, the numbers in the parenthesis are the channels that have been allocated to the primary transmitters whose identities numbers are inside the square brackets beside

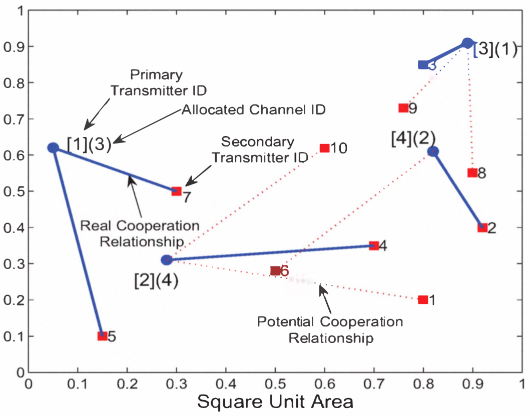


Fig. 4. The result delivered by CGA to DEMO.

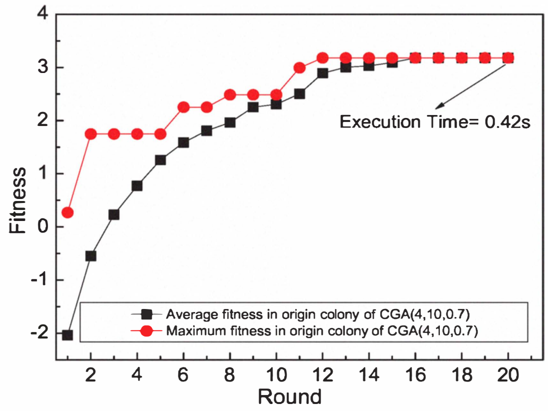


Fig. 5. The evolution process of CGA in DEMO.

the parenthesis. For simplicity, we omit to display all the receivers. In addition, the dashed lines and solid lines indicate the potential and real cooperation relationships between the primary and secondary transmitters, respectively. The evolution process is shown in Fig. 5, where CGA took 0.42s to conducted 20 rounds of evolution and finally converged to fitness 3.1787. DEMO verifies that CGA converges rapidly and can find a channel and cooperation set for each primary transmitter successfully.

We then compare CGA with BS under different topologies of various network sizes to verify the effectiveness of CGA. Specifically, we set the network sizes as $NS(4, 8, 0.7)$, $NS(4, 9, 0.7)$, $NS(4, 10, 0.7)$, and $NS(4, 11, 0.7)$, respectively. For each network size, 50 topologies are generated, and for each topology each algorithm runs 50 times, therefore the value shown in Fig. 6 is the mean of 2500 values. This figure shows that under the given networks, CGA performs as well as 92% of the optimal solution delivered by BS on average, meanwhile the average execution time of CGA is only 0.42s. The execution times of BS, however, is hundreds to thousands of seconds, and we omit to express them in the figure.

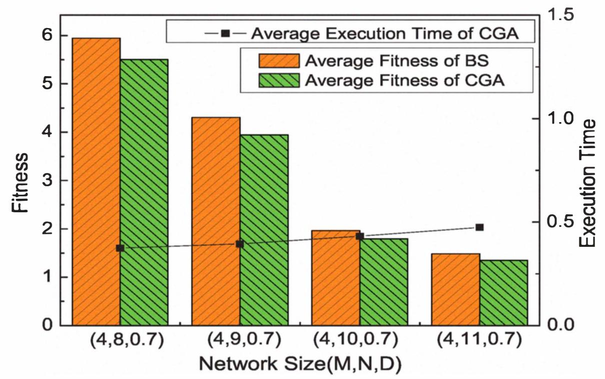


Fig. 6. The fitness and execution time delivered by CGA and BS under various small-scale networks.

VI. Conclusions

In this paper we have studied the joint channel allocation and cooperation set partition problem in CCRNs. We formulated the problem as a 0-1 non-linear programming model. To solve this NP-hard problem, we proposed a suboptimal Centralized Genetic Algorithm (CGA). Extensive experiments by simulations demonstrated the correctness and effectiveness of CGA. It must be mentioned that the proposed algorithms can be extended to solve a generation of JCA-CSP problems in computer networks, wireless networks, and even in the real world, i.e., there are M tasks, N men, and M different kinds of tools, $M < N$. N men are partitioned into M disjoint groups, so that in each group, several men perform one task using a tool. The output of each group can be the linear or non-linear function of the men, the tool, and the task. The problem aims to find a men-task-tool matching, such that the sum of the output of all the groups is maximized. In future, we would like to devise the distributed algorithm for the problem so that it can be used in practical scenarios.

References

[1] J. Zhang and Q. Zhang. Stackelberg game for utility-based cooperative cognitive radio networks. *Proc. Mobihoc'09*, ACM, pp.23-32, 2009.

[2] O. Simeone, I. Stanojev, S. Savazzi, Y. Bar-Ness, U. Spagnolini, and R. Pickholtz. Spectrum leasing to cooperating secondary ad hoc networks. *IEEE Journal on Selected Areas in Communications*, Vol.26, No.1, pp.203-213, 2008.

[3] J. N. Laneman, D. N. C. Tse, and G. W. Wornell. Cooperative diversity in wireless networks: efficient protocols and outage behavior. *IEEE Trans. on Information Theory*, Vol.50, No.12, pp.3062-3080, 2004.

[4] L. Tassiulas and S. Sarkar. Maxmin fair scheduling in wireless ad hoc networks. *IEEE Journal on Selected Areas in Communications*, Vol.23, No.1, pp.163-173, 2005.

[5] V. Kann. Maximum bounded 3-dimensional matching is MAX SNP-complete. *Information Processing Letters*, Vol.49, No.6, pp.309-318, 1994.

[6] H. Kim and K. G. Shin. Asymmetry-aware real-time distributed joint resource allocation in IEEE 802.22 WRANs. *Proc. INFOCOM'10*, IEEE, pp.543-551, 2010.

[7] W. Cai, M. Chen, T. Hara, L. Shu, and T. Kwon. A Genetic approach to multiple mobile agents itinerary planning in wireless sensor networks. *ACM Springer Mobile Networks and Applications (MONET)*, 2010.

[8] F. S. Roberts and B. Tesman. Applied Combinatorics: Second edition. New Jersey, Pearson Prentice Hall, Upper Saddle River, 2004.

[9] Y. Wu and D. H. K. Tsang. Distributed power allocation algorithm for spectrum sharing cognitive radio networks with QoS Guarantee. *Proc. INFOCOM'09*, IEEE, pp.981-989, 2009.