Graph-Based Robust Resource Allocation for Cognitive Radio Networks

Lu Lu, *Student Member, IEEE*, Dawei He, Geoffrey Ye Li, *Fellow, IEEE*, and Xingxing Yu

***Abstract*—Cognitive radio (CR) technology is promising for next generation wireless networks. It allows unlicensed secondary users to use the licensed spectrum bands as long as they do not cause unacceptable interference to the primary users who own those bands. To efficiently allocate resources in CR networks, stable resource allocation based on graph theory is investigated, which takes all users' preferences into account. In this paper, we focus on improving robustness of the stable matching based resource allocation. A truncated scheme generating almost stable matchings is first investigated. Based on the properties of the truncated scheme, two types of edge-cutting algorithms, called *direct edge-cutting* (DEC) and *Gale-Shapley based edge-cutting* (GSEC), are developed to improve resource allocation robustness to the channel state information variation. To mitigate the problem that certain secondary users may not be able to find suitable resources after edge-cutting, *multi-stage* (MS) algorithms are then proposed. Numerical results show that the proposed algorithms are robust to the channel state information variation.**

***Index Terms*—Cognitive radio (CR) network, stable matching, robustness, almost stable, edge-cutting.**

I. Introduction

WITH dramatically increasing demands of high-data rate applications in wireless communications, more and more spectrum resources are needed. Currently, the spectrum bands are licensed to some operators, organizations or companies, leading to the underutilization of some spectrum bands. To improve the spectrum efficiency, *cognitive radio* (CR) technology has been developed [1], [2], where unlicensed users, also known as *secondary users* (SUs), are allowed to use the licensed spectrum bands as long as they do not generate unacceptable interference to the licensed users, also known as *primary users* (PUs). Due to coexistence of PUs and SUs, appropriate resource allocation is important to better utilize the limited spectrum resource.

For resource allocation in CR networks, schemes based on different objectives have been studied [3]. Various schemes have been developed to maximize throughput [4]–[6] and energy efficiency [7], [8] of SUs. Resource allocation can also take the competition feature among SUs into account [9]–[11], where game theory is commonly used. Besides considering the SUs' performance only, the activities of PUs can also be considered jointly to further optimize resource allocation [12], [13].

As a useful and convenient tool, graph theory based schemes have been studied. Here, we focus on using bipartite graphs where SUs and PUs are treated as two partite sets and a resource allocation result can been seen as a matching of the corresponding bipartite graph. To optimize the sum/average utility of SUs, Hungarian algorithm for finding a maximum matching has been used [7], [14]. We can also take both PUs' and SUs' preferences into account and allocate resources based on the stable matching algorithm [15] due to its advantages. First, it incorporates fairness issues. Second, efficient algorithms, such as the *Gale-Shapley* (GS) algorithm, can be used to find a solution with polynomial complexity. Stable matching based resource allocation has been studied in [16]–[19]. In [16], a one-to-many stable matching and its properties have been studied. In [17], stable matching based on queue- and channel-aware lists for cross-layer scheduling has been investigated. In [18], tight lower and upper bounds for stable allocation performance have been derived. Moreover, stable matching with auction has been studied in [19].

Existing schemes on stable matching mainly focus on the design based on a fixed system condition while the robustness to the variation of the system conditions is seldom investigated. In future networks [20], many other types of users, such as machines, will communicate along with human users, and such networks are usually have a large number of nodes. Consequently, it is desirable to design resource allocation schemes that provide robust results with respect to changes in user and channel conditions. Thus, robust resource allocation should be considered.

In this paper, we investigate robust resource allocation based on graph theory, in particular, stable matching. By taking both SUs and PUs preferences into account, resource allocation is first performed based on the GS algorithm for stable matchings. To improve the robustness of resource allocation, a truncated GS algorithm is studied. Such algorithm has first been proposed and compared to the maximum utility matching in [21]. We will investigate the relationship between the matching produced by the truncated GS algorithm and the optimal stable matching when they are used in resource allocation in CR systems. Upper bounds on the number of rounds to reach almost stable resource

Manuscript received April 17, 2014; revised November 05, 2014; accepted April 18, 2015. Date of publication May 13, 2015. This work was supported in part by the NSF under Grants 1247545 and 1443894. This work appeared in part in the *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP), Florence, Italy, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Hongwei Liu.

L. Lu and G. Y. Li are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250 USA (e-mail: lulu0528@gatech.edu; liye@ece.gatech.edu).

D. He and X. Yu are with the School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332-0250 USA (e-mail: dhe9@math.gatech.edu; yu@math.gatech.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2015.2432733

allocation will be derived. Motivated by our theoretical results, we develop two types of edge-cutting algorithms, called *direct edge-cutting* (DEC) and *Gale-Shapley based edge-cutting* (GSEC), to further improve the robustness of our resource allocation schemes. To mitigate the problem that some SUs may not be able to find suitable resources to transmit after edge-cutting, *multi-stage* (MS) algorithms are developed. Below is a summary of main contributions of this paper.

- We develop and investigate a stable resource allocation scheme based on the GS algorithm.
- We study an almost stable resource allocation based on the truncated GS algorithm. We provide bounds on the numbers of rounds for the truncated GS algorithm to achieve ϵ-stable and $(1+\epsilon)$-approximation of the stable resource allocation, which show that the maximum degree of the SU pairs is the most important factor on the performance.
- Motivated by our theoretical results, we developed three types of edge-cutting algorithms, including DEC, GSEC, and MS, to improve the robustness of the resource allocation. Numerical results show that the robustness can indeed be improved.

The rest of this paper is organized as follows. In Section II, we introduce the system model. In Section III, resource allocation based on stable matching is discussed. In Section IV, a truncated algorithm guaranteeing almost stable matching is investigated. Two types of edge-cutting algorithms are proposed in Section V. In Section VI, multi-stage algorithms are developed. Numerical results are provided in Section VII to show the impact of different system parameters on the performance of the developed algorithms. Finally, Section VIII concludes the paper.

II. System Model

We consider an underlay CR network with multiple channels/bands, where PUs have priorities to use N spectrum channels/bands while M SU pairs want to transmit simultaneously. All channels are modeled as Rayleigh block fading channels. Without loss of generality, we assume the j-th PU uses the j-th spectrum band. As shown in Fig. 1, the channel between the i-th SU pair on the j-th spectrum band and the interference channel from the i-th SU transmitter to the j-th PU receiver are denoted as $h_{i,j}$ and $g_{i,j}$, respectively. The transmit power of the i-th SU transmitter on the j-th spectrum band is $P_{i,j}$. The noise power on all spectrum bands is assumed to be the same, denoted as σ^2. A centralized system is assumed, where the control center with CSI allocates resources. Furthermore, the control center will only assign a frequency band to a SU pair when the corresponding CSI is available, including both CSI between the SU pairs and of the interference channel from the SU transmitter to the PU receiver. Correspondingly, the number of spectrum bands available for the i-th SU pair is denoted as $\Delta_{s,i}$ and the number of SU pairs available for the j-th spectrum band is denoted as $\Delta_{p,j}$. We consider the scenario that only one SU pair is allowed on each spectrum band and each SU pair can only access at most one spectrum band.

To protect the PUs, the interference power generated by the SU pairs on the j-th spectrum band should be below a given threshold, that is,

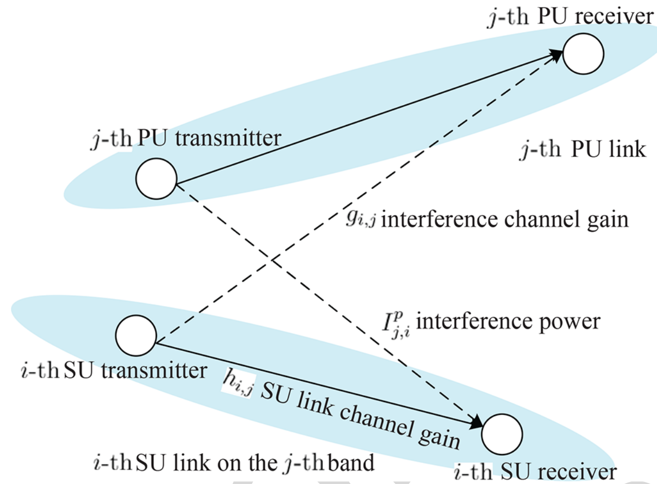

Fig. 1. System model.

$$I_{i,j} = P_{i,j}|g_{i,j}|^2 \leq I_{th}, \quad \forall i,j, \tag{1}$$

where I_{th} is the interference threshold[1]. Moreover, due to the amplifier capacity limit, each SU transmitter has a peak transmit power constraint, P, that is,

$$P_{i,j} \leq P, \quad \forall i. \tag{2}$$

If the i-th SU pair is assigned to the j-th spectrum band, its throughput can be expressed as

$$R_{i,j} = \log_2\left(1 + \frac{|h_{i,j}|^2 P_{i,j}}{\sigma^2 + I^p_{j,i}}\right), \tag{3}$$

where $I^p_{j,i}$ denotes the interference from the j-th PU transmitter to the i-th SU receiver. We assume that the interference powers, $I^p_{j,i}$, are known since the PUs' transmit powers remain the same with or without SU pairs' transmission and, hence, $I^p_{j,i}$ can be estimated in advance.

When performing resource allocation to the SU pairs, we also consider benefits to PUs by incorporating the concept of spectrum trading into the utility function design [13]. PUs will charge more from the SU pairs for providing better service. On the other hand, the performance of PUs will degrade if SU pairs generate strong interference. Therefore, while SU pairs improve their own performance by increasing their transmit powers, they should also get penalties for generating stronger interference. As in [22] and [23], to capture such features of both PUs and SU pairs, if the i-th SU pair uses the j-th spectrum band, the utility can be defined as a linear combination of the throughput of the i-th SU pair, $R_{i,j}$, and the interference generated to the j-th PU, $I_{i,j}$, expressed as

$$w_{i,j}(P_{i,j}) = c_s R_{i,j} - c_p I_{i,j}, \tag{4}$$

[1]Without loss of generality, we assume, for simplicity, that the interference threshold is the same on all the spectrum bands. The results can be directly extended to the system with different interference thresholds on different spectrum bands.

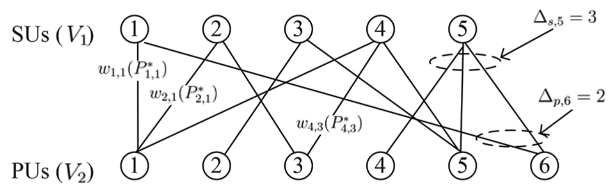

Fig. 2. Bipartite graph illustration with $M = 5$ and $N = 6$.

c_s and c_p are weight factors.

The transmit power, $P_{i,j}$, can be optimized to maximize the utility function in (4) subject to constraints (1) and (2). Since the utility function in (4) is a concave function of $P_{i,j}$, the optimal transmit power, $P^*_{i,j}$, is

$$P^*_{i,j} = \left(\min \left\{ \frac{c_s}{c_p |g_{i,j}|^2} - \frac{\sigma^2 + I^p_{j,i}}{|h_{i,j}|^2}, P, \frac{I_{th}}{|g_{i,j}|^2} \right\} \right)^+, \quad (5)$$

where $(x)^+ = \max\{0, x\}$. Then, if the i-th SU pair is allocated on the j-th spectrum band to transmit, the utility value of the i-th SU pair and the j-th PU can be expressed as $w_{i,j}(P^*_{i,j})$.

Even though (4) is used for resource allocation in this paper, the approaches developed here can easily be adapted for other utility functions.

III. Resource Allocation Based on Stable Matching

As indicated before, we will focus on resource allocation based on graph theory by taking preferences of both PUs and SU pairs into account. We first describe our resource allocation scheme based on stable matching [7] and then discuss its properties.

Stable matching based resource allocation can be described with the help of a *bipartite* graph, say G, with bipartition, V_1, V_2. See Fig. 2, where nodes in V_1 represent SU pairs and nodes in V_2 represent spectrum bands/PUs. As in Section II, the sizes of V_1, V_2 are denoted as M, N, respectively. An edge is put between the i-th SU pair in V_1 and the j-th PU in V_2 if the CSI of the i-th SU pair on the j-th spectrum band is known at the control center. Then, the common utility value of the i-th SU pair and the j-th PU, $w_{i,j}(P^*_{i,j})$, is assigned to the corresponding edge.

To define the preference lists for all SU pairs and PUs, we can regard G as a *complete* bipartite graph, i.e., every node in V_1 is connected to every node in V_2, by setting $w_{i,j}(P^*_{i,j}) = -\infty$ if the CSI of the i-th SU pair on the j-th spectrum band is not available. Furthermore, we assume that the numbers of spectrum bands/PUs and SU pairs are equal, i.e., $M = N$. If the number of SU pairs exceeds the number of spectrum bands, i.e., $M > N$, we can add $(M - N)$ virtual spectrum bands/PUs and edges between all these virtual PUs and all the SU pairs, and set the weights on these new edges to $-\infty$. Similar operation can be done when the number of spectrum bands exceeds the number of SU pairs.

The preference list for the i-th SU pair is defined as

$$\mathcal{L}^s_i = [j_1, \ldots, j_N], \quad (6)$$

where $j_1, \ldots, j_N$ is a permutation of $1, \ldots, N$ satisfying

$$w_{i,j_1}(P^*_{i,j_1}) \leq \cdots \leq w_{i,j_N}(P^*_{i,j_N}). \quad (7)$$

Similarly, the preference list for the j-th PU is defined as

$$\mathcal{L}^p_j = [i_1, \ldots, i_M], \quad (8)$$

where $i_1, \ldots, i_M$ is a permutation of $1, \ldots, M$ satisfying

$$w_{i_1,j}(P^*_{i_1,j}) \leq \cdots \leq w_{i_M,j}\left(P^*_{i_M,j}\right). \quad (9)$$

Our resource allocation procedure will depend on the preference lists of both SU pairs and PUs. To exploit stable matching from graph theory for resource allocation, more definitions are needed. A *matching* in a bipartite graph is a set of pairwise non-adjacent edges. Given a matching in a bipartite graph with partition sets V_1 and V_2, with V_1 (respectively, V_2) consisting of nodes represent SU pairs (respectively, PUs), if an SU pair is matched with a band/PU, we may allocate that band/PU to that SU pair. Hence, a matching in such a bipartite graph naturally gives a resource allocation in which each spectrum band has at most one SU pair and each SU pair accesses at most one spectrum band. We call a matching in a graph *perfect* if all the nodes in the graph are matched.

An edge in the bipartite graph, G, between the i-th SU pair and the j-th PU, denoted as (s_i, p_j), where s_i and p_j denote the i-th SU pair and the j-th PU, respectively, is said to be *unstable* relative to a matching $\mathcal{M}$ in G, if

1 $(s_i, p_j) \notin \mathcal{M}$,
2 s_i is unmatched by $\mathcal{M}$, or prefers p_j over its current match in $\mathcal{M}$, and
3 p_j is unmatched by $\mathcal{M}$, or prefers s_i over its current match in $\mathcal{M}$.

If there is no unstable edges relative to the matching, $\mathcal{M}$, then it is called *stable*.

Based on the above definition, it is desirable to find a resource allocation that corresponds to a stable matching. This can be done by using the GS algorithm [15]. In Table I, we give a version of the GS algorithm in which PUs propose to SU pairs (see Step 2). However, the roles of PUs and SU pairs can be exchanged. So the GS algorithm can be presented in which SU pairs propose to PUs. Here, we focus on the PU-proposing version of the GS algorithm as shown in Table I.

By applying the PU-proposing version of the GS algorithm in Table I, we have the following result (see [15]).

Lemma 1: For the complete bipartite graph in which the number of SU nodes equals the number of PU nodes, the PU-proposing version of the GS algorithm generates a perfect stable matching. Moreover, for this matching, each PU is matched to the best possibility among all stable matchings, and each SU pair is matched to the worst possibility among all stable matchings.

According to Lemma 1, we know that, if we apply the version of the GS algorithm in Table I, each PU will be matched to the best possibility among all stable matchings. That means, the weight in the resulting stable matching for each PU is maximum among all the stable matchings. Thus, the sum or average weight of the matching must be the maximum among all the stable matchings [15].

TABLE I
THE GALE-SHAPLEY PREFERENCE-BASED CHANNEL ALLOCATION ALGORITHM

```
1:  Initialize all the SU pairs and PUs to be free.
2:  Each PU asks its most preferred SU pair which has not
    rejected it yet.
3:  if Each SU pair receives exactly one request from the PUs.
    then
4:      Stop and output the results as the channel allocation
        result.
5:  else
6:      Every SU pair receiving more than one requests chooses
        the PU that is highest on its preference list and rejects
        the other PUs.
7:  end if
8:  if All PUs have proposed to all SU pairs. then
9:      Stop and output the results as the channel allocation
        result.
10: else
11:     Goto Step 2.
12: end if
```

Theorem 2: Let G be a complete bipartite graph with partition sets V_1, V_2 such that V_1 (respectively, V_2) consists of SU (respectively, PU) nodes. Define the weight on edges according to (4) and (5), and define preference lists of nodes according to (6)–(9). Then, the GS algorithm in Table I produces a stable resource allocation that has maximum sum or average weight among all possible stable resource allocations.

As we pointed out before, there are two versions of GS algorithms, the PU- and the SU-proposing ones. In general, the PU- and SU-proposing versions may provide different matching results. Next, we show that, for our studied scenario, the stable matching produced by the version in Table I is unique, regardless of PU- or SU-proposing algorithm is used. By Lemma 1, SU pairs are matched to the worst possibility among all stable matchings after applying the PU-proposing version of the GS algorithm. On the other hand, if we apply the SU-proposing version of the GS algorithm, SU pairs will be matched to the best possibility among all stable matchings [15]. Since the weights on the edges are the same for both SU-proposing and PU-proposing versions, the maximum sum weight among all possible stable matchings is uniquely determined no matter which version of the GS algorithm we use. Hence, the best possibility and the worst possibility for each SU pair among all stable matchings are the same. Thus, we have

Theorem 3: The stable matching produced in the Theorem 2 is unique.

The conclusion on uniqueness in Theorem 3 is the same as in [24]; but the procedure used there is different from the above. Note that, for our scenario, this uniqueness conclusion implies that the same allocation results are obtained regardless of which version of GS algorithm is applied.

Note that Theorem 3 holds if (4) is replaced by a different utility function, as long as the preference lists of SU pairs and PUs are calculated from the weight function on the edges.

IV. ALMOST STABLE RESOURCE ALLOCATION

In the previous section, we proposed a stable resource allocation scheme based on the GS algorithm and discussed some of its properties. However, the stable matching results need not be robust to CSI variation. In fact, CSI variation of one channel may lead to a totally different stable matching result, which is not desirable in practice, especially when the number of nodes in the system is large. To measure the system robustness, we use the amount of variation of the resource allocation after CSI variation as a metric. Based on our definition, with CSI variation, the larger the resource allocation variation is, the less robust the system is to the CSI variation and vice versa.

To improve the robustness of the resource allocation scheme with respect to CSI variation, we consider a truncated GS algorithm. Instead of executing the GS algorithm fully to get a stable resource allocation, the truncated GS algorithm outputs a resource allocation result after executing a fixed number of rounds in the GS algorithm. Note that the GS algorithm, as in Table I, consists of a number of asking-accepting/rejecting rounds; during each round the PUs propose to the SU pairs and the SU pairs answer the proposals. Denote by T the number of rounds we execute the algorithm. From [21], based on the truncated GS algorithm, a change of CSI of one node will only affect the part of resource allocation within distance $2T$ from the node of the change. Comparing to the stable resource allocation, where a change of CSI of one node may affect all nodes in the system, allocation based on the truncated GS algorithm should be much more robust to the CSI variation.

The resulting resource allocation from the truncated GS algorithm may be unstable. To measure the stability of such resource allocation, we introduce the concept of ϵ-stable (or almost stable) matching. Let $\epsilon > 0$ be given. The matching, $\mathcal{M}$, in the graph, G, is said to be ϵ-*stable* if the number of unstable edges in G is at most $\epsilon|\mathcal{M}|$, where $|\mathcal{M}|$ is size of $\mathcal{M}$. Let Δ be the maximum degree of nodes (i.e., $\Delta = \max\{\Delta_{s,i}, \Delta_{p,j}\}$, $\forall i, j$, where $\Delta_{s,i}, \Delta_{p,j}$ are the numbers of edges at the i-th SU pair and the j-th PU, respectively). For the ϵ-stable matching, we have the following theorem from [21]:

Theorem 4: By executing at most $2 + \Delta^2/\epsilon$ rounds of the GS algorithm, we can find an ϵ-stable resource allocation.

The detailed proof for Theorem 4 can be found in [21]. Based on the procedure in [21], we can further bound the number of execution rounds by using $\Delta_s = \max\{\Delta_{s,i}\}$, the maximum degree among the nodes representing SU pairs only.

Theorem 5: By executing at most $2 + \Delta_s^2/\epsilon$ rounds of the GS algorithm, we can find an ϵ-stable resource allocation.

Our proof of Theorem 5 follows the same procedure as that of Theorem 4 by using the maximum degree among nodes for SU pairs, Δ_s, instead of the maximum degree of all nodes, Δ. It is clear that $\Delta_s \leq \Delta$. Hence, in practical scenarios, the bound in Theorem 5 is tighter than the bound in Theorem 4 from [21].

Besides the stability property, we also concern about the utility of resource allocation. Given one resource allocation, $\mathcal{M}$, its utility, denoted as $w_{\mathcal{M}}$, is defined as the sum utilities of PUs/SU pairs in $\mathcal{M}$. Let $\mathcal{S}$ denote the stable resource allocation produced by the GS algorithm, with $w_{\mathcal{S}}$ as the corresponding utility. A resource allocation, $\mathcal{M}$, is a $(1+\epsilon)$-*approximation* of the maximum-weight stable matching if its utility, $w_{\mathcal{M}}$, satisfies $(1+\epsilon)w_{\mathcal{M}} \geq w_{\mathcal{S}}$. For the truncated GS algorithm, we have the following conclusion regarding its utility and the detailed proof is in the Appendix.

Theorem 6: By executing at most $2+\Delta_s/\epsilon$ rounds of the GS algorithm, we can find a $(1+\epsilon)$-approximation of the maximum-weight stable resource allocation.

In [21], the result based on the truncated GS algorithm is compared with the maximum-weight resource allocation (instead of stable resource allocation). For the scenario studied by us, it is more reasonable to compare the result from truncation with other stable resource allocation.

The robustness of the resource allocation from the truncated GS algorithm is affected by T, the number of rounds the GS algorithm is executed. A change in the bipartite graph only affects the result in the radius-$2T$ neighbourhood of the changing point, instead of the whole resource allocation result [21]. Theorems 5 and 6 provide upper bounds on T to achieve ϵ-stability and $(1+\epsilon)$-approximation of the maximum-weight stable resource allocation, respectively. Both bounds depend only on ϵ and Δ_s. Given ϵ, decreasing Δ_s is expected to help decrease the required number of executing rounds, T. From these observations, several algorithms for decreasing Δ_s are developed in the next section in order to achieve robust designs.

V. Edge-Cutting for Robust Design

Based on our discussion in the previous section, the robustness of resource allocation obtained from the truncated GS algorithm depends on the number of rounds, say T, of the GS algorithm that we execute. Smaller T leads to higher robustness. Since the smallest T to achieve ϵ-stability or $(1+\epsilon)$-approximation depends on the instantaneous CSI, we instead focus on reducing the upper bounds on T.

From Theorems 5 and 6, the upper bounds on T to achieve ϵ-stability or $(1+\epsilon)$-approximation are related to Δ_s, the maximum number of available bands of any SU pair. Both bounds can be decreased by decreasing Δ_s. Thus, to improve robustness of resource allocation, a small Δ_s is preferable. It is possible that SU pairs only access CSI of a small number of PU bands, and Δ_s is small naturally. However, PUs, in order to improve their own utilities, may be willing to share their information to attract more SU pairs, as in the spectrum trading scenario [13]. In this case, the maximum number of bands available at SU pairs, Δ_s, may be large. Note that, the number of available bands for the i-th SU pair is the number of edges connected to it in the constructed bipartite graph.

For robust design, we propose edge-cutting algorithms to decrease the maximum number of edges connected to SU pairs, Δ_s. In other words, if the number of available bands at the i-th SU pair, $\Delta_{s,i}$, is larger than a given threshold, denoted as Δ_c^{max}, it will be asked to give up some bands before performing resource allocation. In the following, two different types of edge-cutting algorithms will be developed: *direct edge-cutting* (DEC) that cuts edges based on the preference lists, and *GS-Based Edge-Cutting* (GSEC) that cuts edges based on the GS algorithm.

A. Direct Edge-Cutting (DEC)

In this part, we propose ways to delete edges according to the preference lists of SU pairs and/or PUs. We will first describe a general approach that depends on a parameter $p \in [0,1]$, and we get two special approaches by letting $p = 0$ or 1.

For the edge, (s_i, p_j), between the i-th SU pair s_i and the j-th PU p_j, let $t^p_{i,j} = k$ if s_i is the k-th element on p_j's preference list, and $t^s_{i,j} = l$ if p_j is the l-th element in s_i's preference list. We set a preference value on the edge (s_i, p_j) as a convex combination of $t^p_{i,j}$ and $t^s_{i,j}$, which can be expressed as $t_{i,j} = pt^s_{i,j} + (1-p)t^p_{i,j}$, where $0 \le p \le 1$. We let each SU pair keep $\Delta_c^{\max}$ (a fixed constant) edges that have the highest preference values. So in the resulting graph, each SU pair is incident with precisely $\Delta_c^{\max}$ edges. We call this process *SP-preferred DEC* (SP-DEC) algorithm, as it can be based on the preference lists of both SUs and PUs. The weight factors can be adjusted according to the priorities in the preference lists of SU pairs and PUs. We consider two extreme cases: $p = 1$ and 0.

When $p = 1$, then SP-DEC keeps $\Delta_c^{\max}$ edges at each SU pair that is at the top of its preference list; this approach is called *SU-preferred DEC* (S-DEC) as it only takes SU's preference lists into consideration.

When $p = 0$, then SP-DEC keeps $\Delta_c^{\max}$ edges at each SU pair with the highest preference values (we can randomly choose the edges that have same preference value), and those edges occur at the top of PUs' preference lists; this approach is called *PU-preferred DEC* (P-DEC) as it takes PUs' preference lists into consideration.

B. GS-Based Edge-Cutting (GSEC)

The DEC procedures described above reduce the maximum number of edges connected to the SU pairs, and are easy to implement. However, they do not take the original GS matching process into account. Hence, the resource allocation results obtained from the GS algorithm after the DEC procedures may be quite different from the result obtained by applying the GS algorithm on the original graph. Moreover, it is possible that certain SU pairs with similar preference lists will compete with each other, so there is a chance that some of them are not allocated any resource.

For instance, consider a system with two PUs, p_1 and p_2, and two SU pairs, s_1 and s_2, where both p_1 and p_2 prefer s_1 to s_2, and both s_1 and s_2 prefer p_1 to p_2. Applying the GS algorithm in Table I, s_1 is assigned to p_1, and s_2 is assigned to p_2. However, the result can be different if we apply the GS algorithm after a DEC procedure. Set $\Delta_c^{\max} = 1$, so that the maximum number of edges connected to each SU pair is one. Applying S-DEC, we obtain a graph in which both s_1 and s_2 are connected to p_1, but not to p_2. Then applying the GS algorithm to the new graph, s_1 is allocated to the channel of p_1, but s_2 cannot transmit.

In order to use DEC to obtain a resource allocation result closer to the result obtained by applying the GS algorithm, we propose an edge-cutting algorithm based on the GS algorithm (GSEC algorithm). In general, the edges involved in the first few rounds of the GS algorithm will be kept. However, minor changes are made to satisfy the requirement on the maximum number of edges connected to each node representing an SU pair. In order to make sure the maximum number of edges connected to each SU pair is $\Delta_c^{\max}$, each SU pair can reject at most $\Delta_c^{\max} - 1$ edges during the GS process. We now describe the GSEC algorithm. See Table II.

First, a vector $\mathbf{d} \in \mathbb{N}^M$ is used in which the ith coordinate, d_i, records the number of edges rejected by the ith SU pair, s_i.

TABLE II
GS-BASED EDGE-CUTTING (GSEC) ALGORITHM

1: Initialize all SUs and PUs to be free. Define a vector $\mathbf{d} \in \mathbb{N}^M$ as an indicator with the i-th coordinate d_i recording the number of PUs rejected by s_i, the i-th SU pair. Set $d_i = 0$ for $i = 1, ..., M$. Define a vector $\mathbf{m} \in \mathbb{N}^M$ with the i-th element $m(s_i)$ recording the indices of the PUs connected to s_i. Initialize $m(s_i) = 0$ for $i = 1, ..., M$. Let $\mathcal{A}$ denote the indices of SU pairs which are available and initialize $\mathcal{A} = \{1, ..., M\}$. Define $\mathbf{E} \in \mathbb{N}^{M \times N}$ to record the edges that remain after edge-cutting. Set $e_{i,j} = 1$ if the edge between s_i and the j-th PU is kept. Initialize $\mathbf{E}$ to be a zero matrix.
2: Each PU asks its most preferred SU with indices in $\mathcal{A}$ that has not rejected it.
3: **for** $i \in \mathcal{A}$ **do**
4: Let $\mathcal{P}_i$ denote the set of the indices of those PUs which have proposed to s_i during this round.
5: **if** s_i is not proposed, $\mathcal{P}_i = \emptyset$ **then**
6: Do nothing.
7: **else if** s_i is only proposed by one PU, say the j-th PU and $m(s_i) = 0$ **then**
8: Accept the j-the PU and set $e_{i,j} = 1$.
9: **else**
10: **if** $m(s_i) \neq 0$ **then**
11: $\mathcal{P}_i = \mathcal{P}_i \cup m(s_i)$
12: **end if**
13: Accept the PU in $\mathcal{P}_i$ that is highest on s_i's preference list.

$$m(s_i) = \arg\max_{j \in \mathcal{P}_i} w_{i,j}(P^*_{i,j}).$$

Increase d_i by the number of PUs rejected by s_i during this round, that is $d_i = d_i + |\mathcal{P}_i| - 1$, where $|\mathcal{P}_i|$ denotes the number of elements in $\mathcal{P}_i$.
14: **if** $d_i \geq \Delta_c^{max} - 1$ **then**
15: $\mathcal{A} = \mathcal{A} - \{i\}$
16: **if** $d_i = \Delta_c^{max} - 1$ **then**
17: $e_{i,j} = 1$, for $j \in \mathcal{P}_i$.
18: **else**
19: Delete $d_i - (\Delta_c^{max} - 1)$ elements from $\mathcal{P}_i$ that are lowest on s_i's preference list. Then, set $e_{i,j} = 1$, for $j \in \mathcal{P}_i$.
20: **end if**
21: **end if**
22: **end if**
23: **end for**
24: **if** $\mathcal{A} = \emptyset$ **then**
25: Stop.
26: **else if** All PUs are matched or they have already proposed to all SUs in $\mathcal{A}$ **then**
27: For $i \in \mathcal{A}$, add $\Delta_c^{max} - 1 - d_i$ edges between s_i and those PUs at the top of s_i's preference list and then stop.
28: **else**
29: Go to step 3.
30: **end if**

Initially $d_i = 0$ for all i, and all SU pairs are involved. During each round of the GS algorithm in Table I, each PU asks its most preferred SU pair. At the end of a round, increase d_i by the number of edges/PUs rejected by s_i during this round, and update $\mathcal{P}_i$, the set of the indices of the PUs who have asked s_i during this round. There are three cases after executing a particular round of the GS algorithm: $d_i < \Delta_c^{\max} - 1$, $d_i = \Delta_c^{\max} - 1$, and $d_i > \Delta_c^{\max} - 1$.

If $d_i < \Delta_c^{\max} - 1$, then the number of PUs rejected by s_i has not reached the limit; in this case, keep all edges between s_i and the PUs with indices in $\mathcal{P}_i$, and s_i remains in the process for the next round.

If $d_i = \Delta_c^{\max} - 1$ then the number of PUs rejected by s_i reaches the limit; keep all edges between s_i and the PUs with indices in $\mathcal{P}_i$, but s_i is removed from the process.

Now assume $d_i > \Delta_c^{\max} - 1$. Then the number of PUs rejected by s_i exceeds the limit. So s_i will be removed from the process. To meet the requirement that $d_i \leq \Delta_s - 1$, we remove $d_i - (\Delta_c^{\max} - 1)$ PUs with indices in $\mathcal{P}_i$. There are different ways to remove PUs with indices in $\mathcal{P}_i$. A natural way is to remove those $d_i - (\Delta_c^{\max} - 1)$ PUs with the lowest ranks in the preference list of s_i. Thus, the process keeps the PUs on the top of s_i's preference list, and it is independent of other PUs and SU pairs.

The GSEC process stops when all SU pairs are removed from the process or when the GS algorithm stops. It is possible that the GS algorithm stops before all SU pairs are removed. In this case, all PUs and SU pairs are matched. If this case occurs, $\Delta_c^{\max} - 1 - d_i$ edges can be added for each s_i that is still in the process.

Therefore, when the GSEC process stops, each SU pair will have $\Delta_c^{\max}$ edges. The detailed algorithm is given in Table II. Note that, during the process of the GSEC, a matching will be generated as well. This matching will be the same as the one if we execute the GS algorithm on the bipartite graph from the GSEC. Thus, we can use the matching generated from the GSEC directly for resource allocation without running the GS algorithm again.

To illustrate the difference between GSEC and DEC, we apply GSEC algorithm to the example at the beginning of this subsection. Both PUs, p_1 and p_2, ask the first SU pair, s_1, during the first round. Since the s_1 can only keep $\Delta_c^{\max} = 1$ edge and it prefers p_1 over p_2, it keeps only the edge (s_1, p_1), and s_1 leaves the process. In the second round, the second PU, p_2, asks the second SU, s_2, and keeps the edge (s_2, p_2). The GS algorithm stops. (Since both s_1 and s_2 have $\Delta_c^{\max} = 1$ edges, no extra edges need to be added.) Based on the graph after GSEC, s_1 will be allocated to the channel of p_1 while s_2 will be on the channel of p_2. This resource allocation result is the same as the original GS algorithm while the maximum number of edges connected to each SU pair is one after the GSEC. Therefore, resource allocation based on the GSEC process might give a result that is closer to the original GS algorithm while reducing the maximum number of edges connected to individual SU pairs.

Remark 1: Essentially, both DEC and GSEC shorten the preference lists of SU pairs and PUs. Any change on the deleted edges has no impact on the resource allocation results. This also explains why the edge-cutting algorithms improve robustness of the resource allocation.

Remark 2: After edge-cutting, some of SU pairs may not be able to find suitable channels. Consider the example at the beginning of Section V-B, only the first SU pair can transmit after DEC while both of them can transmit without edge-cutting. In the next section, a multi-stage edge-cutting algorithm will be proposed to mitigate this problem.

Remark 3: In addition to improving the robustness of resource allocation, edge-cutting also reduces the computational complexity.

VI. Multi-Stage Edge-Cutting

In the previous section, we proposed two different types of edge-cutting algorithms (DEC and GSEC) to reduce the maximum number of channels available at individual SU pairs after edge-cutting, $\Delta_c^{\max}$, to improve the robustness of resource allocation with respect to the CSI variation. We expect that resource allocation results are more robust when $\Delta_c^{\max}$ is made smaller. However, if $\Delta_c^{\max}$ is too small (so the number of channels available at each SU pair is small), some SU pairs may not be able to find suitable channels to transmit. To mitigate this problem, we propose a *multi-stage* (MS) edge-cutting algorithm.

The basic idea of the MS algorithm is to operate the GS algorithm after DEC or GSEC algorithms several times and to adjust the edge-cutting result during the procedure to increase the number of SU pairs that are allocated channels to transmit. The procedure is called MS algorithm, briefly described as follows. First, the DEC or the GSEC algorithm is operated on the original graph and then the GS algorithm. (This is the same as the procedure in Section V.) We then output the corresponding edge-cutting result only for the SU pairs who have been assigned channels. Remove the matched SU pairs, the corresponding matched PU nodes, and the edges among them from the procedure. Denote the sets of remaining SU pairs and PUs as $\mathcal{S}_r$ and $\mathcal{P}_r$, respectively. Note that no SU pair in $\mathcal{S}_r$ was asked by any PUs in the previous stage since if one SU pair was asked, it was assigned a channel and would not be in $\mathcal{S}_r$. Thus, the nodes in $\mathcal{S}_r$ has no impact on the previous stage. Then, we can ignore all previous procedures related to nodes in $\mathcal{S}_r$ and conduct the edge-cutting and the resource allocation again based on the subgraph of the original graph constituting of all nodes in $\mathcal{S}_r \bigcup \mathcal{P}_r$, and corresponding edges. Through this way, the maximum number of edges connected to each SU pair involving in the MS algorithm is still $\Delta_c^{\max}$. Repeat the procedure until all nodes are removed. The detailed procedure is given in Table III.

Moreover, like the GSEC algorithm, the MS algorithm will generate a matching during the process. This matching will be the same as the one if we execute the GS algorithm on the output bipartite graph from the MS algorithm and thus, can be used directly for resource allocation.

In general, by using the MS algorithm, the number of SU pairs which can transmit increases while the maximum number of edges connected to each SU pair kept in the graph is still $\Delta_c^{\max}$.

TABLE III
Multi-Stage (MS) Edge-Cutting Algorithm

1: Initialize the input graph (V_1^i, V_2^i, E^i) as $V_1^i \leftarrow V_1$, $V_2^i \leftarrow V_2$ and $E^i \leftarrow E$, where V_1, V_2, and E denote all SU pairs, PUs, and all edges among them, respectively, from the original graph. Initialize the output graph (V_1^o, V_2^o, E^o) to be empty.
2: The DEC or GSEC algorithm is conducted on (V_1^i, V_2^i, E^i) to reduce the number of edges connected to individual SU pairs. The resulting graph is denoted as (V_1^e, V_2^e, E^e), where V_1^e, V_2^e, and E^e denote the SU pairs, PUs and the remaining edges among SU pairs and PUs after edge-cutting on (V_1^i, V_2^i, E^i), respectively.
3: The GS algorithm is operated on (V_1^e, V_2^e, E^e).
4: **if** Each SU pair is allocated a channel for transmission, **then**
5:    Stop the algorithm and set the output graph (V_1^o, V_2^o, E^o) as $V_1^o \leftarrow V_1^o \cup V_1^e$, $V_2^o \leftarrow V_2^o \cup V_2^e$, and $E^o \leftarrow E^o \cup E^e$
6: **else**
7:    Let $\mathcal{S}_r$ denote the set of indices of those SU pairs which are not yet assigned a channel, and $\mathcal{P}_r$ denote the set of indices of those PUs which have no matched SU pair. Update $V_1^o \leftarrow V_1^o \cup (V_1^e - \mathcal{S}_r)$, $V_2^o \leftarrow V_2^o \cup (V_2^e - \mathcal{P}_r)$, and add edges among $V_1^e - \mathcal{S}_r$ and $V_2^e - \mathcal{P}_r$ in E^e to E^o.
8:    Update $V_1^i \leftarrow \mathcal{S}_r$, $V_2^i \leftarrow \mathcal{P}_r$, and E^i as the edges among $\mathcal{S}_r$ and $\mathcal{P}_r$ based on the original graph (V_1, V_2, E). The preference list of each member of $V_1^i \cup V_2^i$ will simply be the restriction of the original preference list to $V_1^i \cup V_2^i$.
9:    **if** All nodes have empty preference list, **then**
10:       Stop the algorithm.
11:    **else**
12:       Go to item 2.
13:    **end if**
14: **end if**

VII. Numerical Results

In this section, numerical results are presented to show the performance of the proposed algorithms in the application scenario where the number of users is large and the number of channels with CSI variation is small. Here, we consider the case with 200 SU pairs and 200 PUs, i.e., $M = N = 200$ and assume all CSI is known. Results are averaged after 20 000 trials. For each trial, algorithms are executed once based on the original CSI and then, conducted second time by changing CSI of 5 SU pairs. The utility gap shown in Figs. 3(b), 4(b), and 5(b) is defined as $(w_\mathcal{S} - w)/w_\mathcal{S}$, where $w_\mathcal{S}$ is the sum utility based on the GS algorithm and w is the utility based on the proposed algorithm. The SU allocation variation is defined as the number of SU pairs with different allocation results after CSI change. The SU allocation variation saving in Figs. 3(a), 4(a), and 5(a) is the relative value compared to the GS algorithm. We also show the allocation difference between each proposed algorithm and the GS algorithm, which is defined as the number of SU pairs with different allocation results between a proposed algorithm and the GS algorithm. For a resource allocation scheme, smaller resource allocation variation means higher robustness to the CSI variation. For all results, we set the *signal-to-noise ratios* (SNRs) between any SU pair and any interference channel from SU transmitter to the PU receiver as 0 dB and -10 dB, respectively. Interference threshold is -10 dB and the maximum transmit power is 10 dB.

A. Single-Stage Truncation or Edge-Cutting

Figs. 3(a) and (b) show the impact of the maximum number of available bands for each SU pair, $\Delta_c^{\max}$, on allocation variation saving and utility, where $\epsilon = 0.1$. Fig. 3(a) shows that the reduction in $\Delta_c^{\max}$ increases the robustness of resource allocation for all edge-cutting algorithms. Even though smaller $\Delta_c^{\max}$ leads to a larger utility gap from the GS algorithm as shown in Fig. 3(b), the saving on SU allocation variation is significant compared to that of the utility gap.

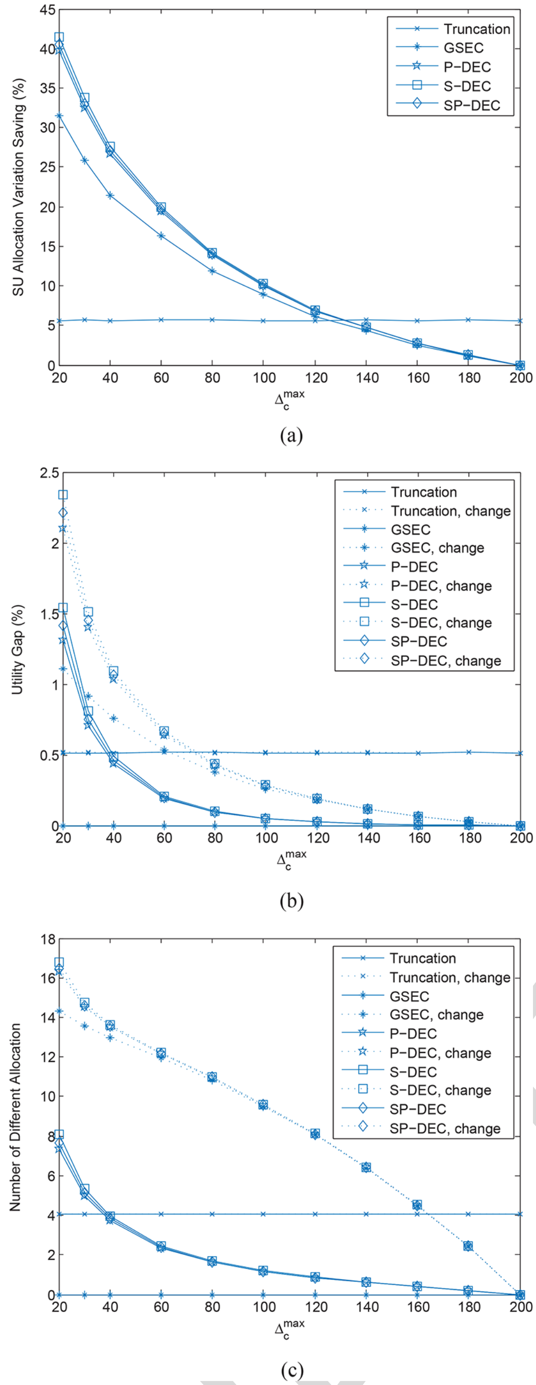

Fig. 3. Performance for algorithms with truncated or edge-cutting. (a) SU allocation variation saving. (b) Utility gap. (c) Resource allocation difference.

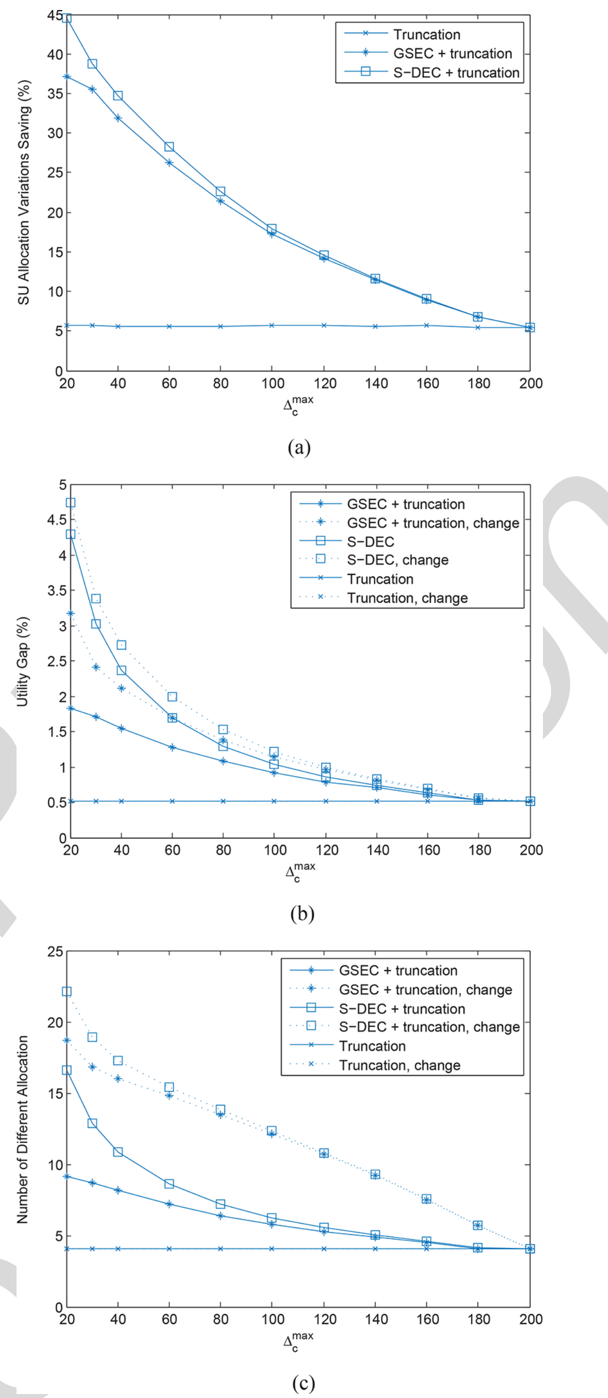

Fig. 4. Performance for algorithms with truncation and edge-cutting. (a) SU allocation variation saving. (b) Utility gap. (c) Resource allocation difference.

Next, let us compare the performance of the truncated GS algorithm and the edge-cutting algorithms. Fig. 3(a) shows that when $\Delta_c^{\max}$ is small, the edge-cutting algorithms may provide higher resource allocation robustness than the truncated GS algorithm. When $\Delta_c^{\max}$ is large, the truncated GS algorithm can provide more robust results. For the utility gap in Fig. 3(b), except for the GSEC before CSI change, it is larger for edge-cutting algorithms than for the truncated GS algorithm when $\Delta_c^{\max}$ is small, and the other way around when $\Delta_c^{\max}$ is large. Executing the GSEC algorithm before CSI change can provide a negligible performance gap from the GS algorithm. This is due to the fact the edges kept by the GSEC algorithm are based on the GS algorithm, and almost all edges involved in the GS algorithm are kept. The advantage cannot be maintained after CSI changes. But, the GSEC algorithm still provides smaller utility gap than DEC algorithms. Different DEC algorithms have similar performance.

As we discussed in Section V-B, it is expected that resource allocation results from GSEC are closer to those from the GS algorithm than those from DEC, which is illustrated in Fig. 3(c). GSEC provides the same resource allocation result as the GS

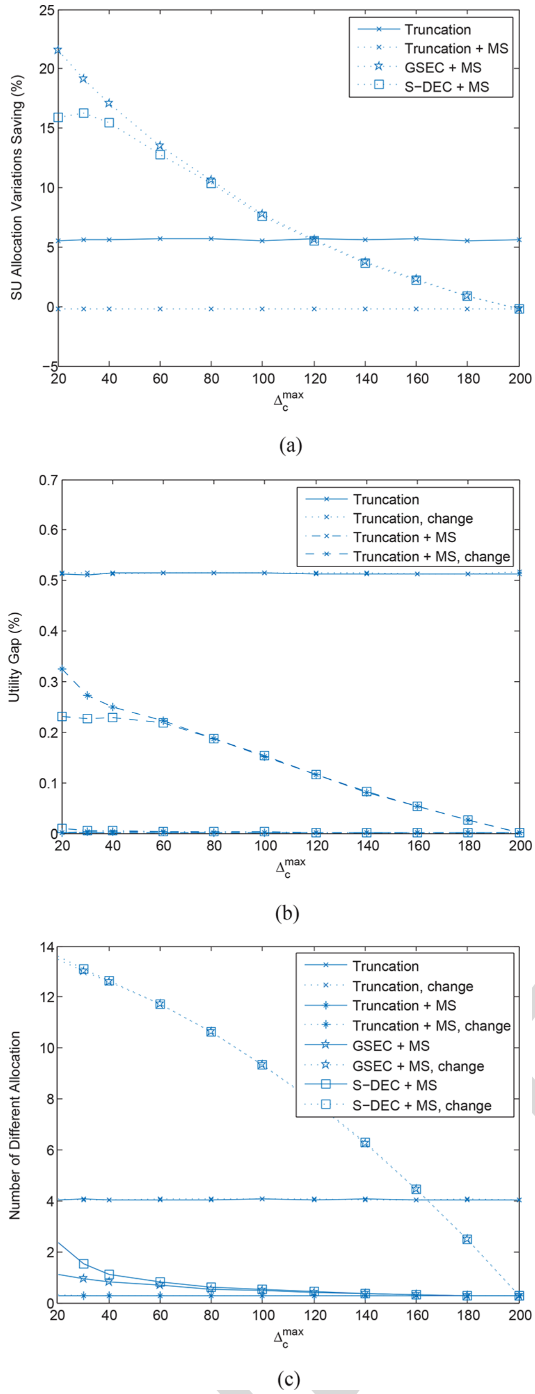

Fig. 5. Performance for different multi-stage algorithms. (a) SU allocation variation saving. (b) Utility gap. (c) Resource allocation difference.

algorithm even when the maximum available channels allowed at each SU pair is only 10% of all channels, i.e., $\Delta_c^{\max} = 20$. However, the DEC algorithms always provide resource allocation results that are different from those from the GS algorithm. This advantage of GSEC is only significant for the case before CSI change. After CSI change, the performance gap becomes smaller since GSEC is operated based on the CSI before change. Comparing the results in Fig. 3(a), GSEC loses certain level of robustness in order to get resource allocation results closer to those from the GS algorithm. This is the case because the more the algorithm depends on the original graph, the more sensitive it is to change on the graph (i.e., the CSI variations).

B. Single-Stage Truncation and Edge-Cutting

Besides operating truncation or edge-cutting separately, these operations can be combined. For example, we can fist apply edge-cutting algorithms on the original graph. We then run the GS algorithm on the new graph, and stop when we obtain a resource allocation that is ϵ-stable or is an $(1+\epsilon)$-approximation compared with the results on the original graph. Figs. 4(a) and (b) show the SU allocation variation saving and utility performance, respectively. The results based on truncation only are also provided for comparison reason. Moreover, all three DEC algorithms have similar performance, and only results based on S-DEC are provided here. As $\Delta_c^{\max}$ increases, the performance curves of algorithms with edge-cutting and truncation converge to that of the truncated only algorithm, instead of to that of the original GS algorithm. Compared with the results in Figs. 3(a) and (b), utilizing both edge-cutting and truncation can result in higher SU allocation variation saving accompanied a larger utility gap. The resource allocation difference is shown in Fig. 4(c). For all edge-cutting algorithms with truncation, the number of different resource allocations is larger compared to that from the edge-cutting only case and the performance curve converges to the truncation only case as $\Delta_c^{\max}$ increases.

C. Multi-Stage Truncation and Edge-Cutting

In this part, the performance of the MS algorithm is presented. Figs. 5(a) and (b) show the utility gap and SU allocation variation saving, respectively. With the help of the MS algorithm, the utility gap between the MS algorithm and the original GS algorithm is significantly reduced compared with the single-stage algorithm. The gap based on the CSI before the change is negligible while there is a small gap, less than 0.5%, after the CSI change. However, the SU allocation variation saving is smaller compared with the corresponding single-stage algorithms. Comparing results based on different edge-cutting algorithms, the utility gap for MS GSEC is slightly larger than the MS DEC algorithms while it can have higher SU allocation variation saving. This is different from the single-stage case. For MS GSEC, it can allocate more SU pairs in the first stage and the impact of the latter stages is smaller compared to the other DEC algorithms. Thus, it can keep higher SU allocation variation saving than the single-stage algorithm while the utility gap is larger as well.

Moreover, Fig. 5(c) shows the allocation difference between the MS algorithm and the original GS algorithm. Comparing with the results in Fig. 4(c), the number of allocation difference based on the MS algorithms is smaller than the one from the single-stage algorithm.

VIII. Conclusion

In this paper, we have studied robust resource allocation for CR networks. First, we develop a resource allocation scheme based on stable matching, which takes both SU pairs' and PUs' preferences into account. We then discuss the properties of almost stable resource allocation, which is robust to the CSI variation compared with stable resource allocation. Based on the properties of almost stable resource allocation, we propose the DEC and GSEC algorithms to further improve the robustness of the resource allocation scheme. The MS algorithms are then discussed to compensate for possible losses from the DEC and GSEC algorithms. Numerical results show that our proposed schemes are robust to the CSI variations.

Appendix A
Proof of Theorem 6

The proof of Theorem 6 follows an argument in [21]. Let T denote the number of rounds we execute the GS algorithm. For $0 \le n \le T$, let $\mathcal{M}_n$ denote the matching at the end of n-th round. The i-th SU pair is denoted as s_i, and the j-th PU is denoted by p_j. Define w as a weight function on edge. So $w_{s_i,p_j}(P^*_{i,j})$ is the weight of edge between s_i and p_j. If s_i is matched at the end of the n-th round, we use $m_n(s_i)$ to denote the PU matched to s_i. Let $C_n(p_j)$ denote the set of those SU pairs which have not rejected p_j at the end of n-th round (which is called the candidate set of p_j).

During any round, each PU removes at most one SU node from its candidate set, since each PU proposes to at most one SU pair during any round. The edge (s_i, p_j) is called *lost* if s_i is removed from the candidate set of p_j. We use L_n to denote the set of the lost edges by the end of the n-th round. For $\mathcal{M}_n$, the matching at the end of the n-th round, we define the weight of s_i as $w_n(s_i) = w_{s_i,m_n(s_i)}$ (if s_i is matched in $\mathcal{M}_n$) and $w_n(s_i) = 0$ (if s_i is unmatched in $\mathcal{M}_n$), where $w_{s_i,m_n(s_i)} := w_{s_i,m_n(s_i)}(P^*_{s_i,m_n(s_i)})$, see (4) and (5). Then, the total weight of the matching $\mathcal{M}_n$ can be expressed as

$$w_{\mathcal{M}_n} = \sum_{s_i \in V_1} w_n(s_i), \tag{10}$$

where V_1 denotes the set of SU nodes. Moreover, the total weight of the edges in L_n, denoted as w_{L_n}, is defined as the sum of weights of the edges in L_n.

We define the potential of p_j, denoted as $f_n(p_j)$, as follows. If p_j is matched in $\mathcal{M}_n$ or $C_n(p_j) = \emptyset$ then set $f_n(p_j) = 0$. Otherwise, set

$$f_n(p_j) = \max_{s_i \in C_n(p_j)} w_{s_i,p_j}, \tag{11}$$

which is the maximum weight of edges connecting p_j to the SU pairs in $C_n(p_j)$. The total potential of the matching $\mathcal{M}_n$ at the end of n-th round is the sum of the potentials of all PUs, that is,

$$f_{\mathcal{M}_n} = \sum_{p_j \in V_2} f_n(p_j), \tag{12}$$

where V_2 denotes the set of PU nodes.

Lemma 7: For $n \ge 2$, $f_{\mathcal{M}_n} \le w_{L_n} - w_{L_{n-1}}$.

Proof: Let p_j denote the j-th PU node. If $f_n(p_j) > 0$ then p_j is not matched by $\mathcal{M}_n$ and $C_n(p_j) \ne \emptyset$. Thus, during the n-th round, p_j is rejected by some SU pair, say s_k (the k-th SU node), and the edge (s_k, p_j) is lost. So $(s_k, p_j) \in L_n - L_{n-1}$. Note that p_j prefers s_k over other SU pairs in $C_n(p_j)$, which means, $f_n(p_j) \le w_{s_k,p_j}$. Hence,

$$f_{\mathcal{M}_n} = \sum_{p_j \in V_2} f_n(p_j) \le \sum_{p_j \in V_2} w_{s_k,p_j} \le w_{L_n} - w_{L_{n-1}},$$

and the lemma is proved. ■

Lemma 8: For $n \ge 2$, $f_{\mathcal{M}_n} \le f_{\mathcal{M}_{n-1}}$.

Proof: Since $f_{\mathcal{M}_n} = \sum_{p_j \in V_2} f_n(p_j)$ and $f_{\mathcal{M}_{n-1}} = \sum_{p_j \in V_2} f_{n-1}(p_j)$, we compare $f_n(p_j)$ and $f_{n-1}(p_j)$ for each $p_j \in V_2$.

If p_j is matched in both $\mathcal{M}_{n-1}$ and $\mathcal{M}_n$, then by definition, $f_n(p_j) = f_{n-1}(p_j) = 0$.

If p_j is matched in $\mathcal{M}_n$, but not in $\mathcal{M}_{n-1}$, then by definition, $f_n(p_j) \le f_{n-1}(p_j)$.

Now assume that p_j is matched to some $s_i \in V_1$ in $\mathcal{M}_{n-1}$ but p_j is not matched in $\mathcal{M}_n$. Then $f_{n-1}(p_j) = 0$ and $f_n(p_j) = w_{s_k,p_j}$, where s_k is such that $w_{s_k,p_j} = \max_{s_t \in C_n(p_j)} w_{s_t,p_j}$. Moreover, since in the n-th round, s_i must reject p_j and accept another PU, say $p_{j'}$ with $j \ne j'$, we have $f_{n-1}(p_{j'}) = w_{s_i,p_{j'}}$ and $f_n(p_{j'}) = 0$. Thus,

$$\begin{aligned} f_n(p_j) - f_{n-1}(p_j) &= w_{s_k,p_j} \le w_{s_i,p_j} \le w_{s_i,p_{j'}} \\ &= f_{n-1}(p_{j'}) - f_n(p_{j'}), \end{aligned}$$

which is equivalent to

$$f_n(p_j) + f_n(p_{j'}) \le f_{n-1}(p_j) + f_{n-1}(p_{j'}).$$

By summing up over all PUs, we have $f_{M_n} \le f_{M_{n-1}}$. ■

Lemma 9: For $n \ge 2$, $w_{L_n} \ge (n-1) f_{M_n}$.

Proof: By Lemmas 7 and 8, $w_{L_n} = \sum_{t=2}^{n} (w_{L_t} - w_{L_{t-1}}) \ge \sum_{t=2}^{n} f_{M_t} \ge (n-1) f_{M_n}$. ■

Lemma 10: For all $n \ge 2$, $w_{L_n} \le (\Delta_s - 1) w_{M_n}$.

Proof: Let s_i denote the i-th SU pair and p_j denote the j-th PU. If $(s_i, p_j) \in L_n$ (i.e., lost by the end of the n-th round), then s_i is matched to a better choice, $m_n(s_i)$, by the end of the n-th round. That is, $w_{s_i,p_j} < w_{s_i,m_n(s_i)}$. Since s_i can lose at most $\Delta_s - 1$ edges, the total weight of all lost edges adjacent to s_i is at most $(\Delta_s - 1) \cdot w_{s_i,m_n(s_i)}$. Hence, by summing over all SU pairs, we obtain $w_{L_n} \le (\Delta_s - 1) w_{M_n}$. ■

By Lemmas 9 and 10, we have

Lemma 11: For any real $\gamma > 0$ and integer $n \ge 1 + (\Delta_s - 1)/\gamma$, $f_{M_n} \le \gamma w_{M_n}$.

Next, we present the proof of Theorem 6.

Proof: Let $\mathcal{S}$ be the unique stable matching produced by the GS algorithm. We use s_i to denote the i-th SU pair and p_j to denote the j-th PU. Suppose $(s_i, p_j) \in \mathcal{S}$.

Note that when p_j is matched to s_k in $\mathcal{M}_n$ for some k (possibly $k = i$), p_j prefers s_k over s_i and $w_{s_i,p_j} \le w_{s_k,p_j} = w_n(p_j)$. Also note that when p_j is not matched in $\mathcal{M}_n$, then p_j has not asked s_i, $s_i \in C_n(p_j)$, and $w_{s_i,p_j} \le f_n(p_j)$.

Hence,

$$w_{s_i,p_j} \le f_n(p_j) + w_n(p_j)$$

and therefore

$$\begin{aligned} w_{\mathcal{S}} &= \sum_{(s_i,p_j)\in\mathcal{S}} w_{s_i,p_j} \\ &\le \sum_{p_j \in V_2} (f_n(p_j) + w_n(p_j)) \\ &= f_{\mathcal{M}_n} + w_{\mathcal{M}_n}. \end{aligned}$$

From Lemma 11, we conclude that

$$w_{\mathcal{S}} \le (1+\epsilon) w_{M_n}$$

whenever $n \ge 1 + (\Delta_s - 1)/\epsilon$.

Thus, Theorem 6 follows by choosing $T = \lfloor 2 + \Delta_s/\epsilon \rfloor$. ■

REFERENCES

[1] J. Mitola and G. Q. Maguire, "Cognitive radio: Making software radios more personal," *IEEE Personal Commun.*, vol. 6, no. 4, p. 13, Aug. 1999.

[2] L. Lu, X. Zhou, U. Onunkwo, and G. Y. Li, "Ten years of research in spectrum sensing and sharing in cognitive radio," *EURASIP J. Wireless Commun. Netw.* vol. 28, Jan. 2012 [Online]. Available: http://jwcn.eurasipjournals.com/content/2012/1/28, [Online]. Available:

[3] E. Z. Tragos, S. Zeadally, A. G. Fragkiadakis, and V. A. Siris, "Spectrum assignment in cognitive radio networks: A comprehensive survey," *IEEE Commun. Survey Tutorials*, vol. 15, no. 3, pp. 1108–1135, 2013.

[4] L. Musavian and S. Aissa, "Capacity and power allocation for spectrum-sharing communications in fading channels," *IEEE Trans. Wireless Commun.*, vol. 8, no. 1, pp. 148–156, Jan. 2009.

[5] V. Asghari and S. Aissa, "Adaptive rate and power transmission in spectrum-sharing systems," *IEEE Trans. Wireless Commun.*, vol. 9, no. 10, pp. 3272–3280, Oct. 2010.

[6] X. Zhou, G. Y. Li, D. Li, D. Wang, and A. C. K. Soong, "Probabilistic resource allocation for opportunistic spectrum access," *IEEE Trans. Wireless Commun.*, vol. 9, no. 9, pp. 2870–2879, Sep. 2010.

[7] L. Lu, D. He, X. Yu, and G. Y. Li, "Energy efficient resource allocation for cognitive radio networks," presented at the IEEE Global Telecommun. Conf. (GLOBECOM'13), Atlanta, Dec. 2013.

[8] C. Xiong, L. Lu, and G. Y. Li, "Energy-efficient spectrum access in cognitive radios," *IEEE J. Sel. Area Commun.*, vol. 32, no. 3, pp. 550–562, Mar. 2014.

[9] Y.-E. Lin, K.-H. Liu, and H.-Y. Hsieh, "Design of power control protocols for spectrum sharing in cognitive radio networks: A game-theoretic perspective," presented at the IEEE Int. Conf. Commun. (ICC), Cape Town, May 2010.

[10] S. Buzzi and D. Saturnino, "A game-theoretic approach to energy-efficient power control and receiver design in cognitive CDMA wireless networks," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 1, pp. 137–150, Feb. 2011.

[11] Z. Tian, G. Leus, and V. Lottici, "Joint dynamic resource allocation and waveform adaptation for cognitive networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 2, pp. 443–454, Feb. 2011.

[12] S. K. Jayaweera, G. Vazquez-Vilar, and C. Mosquera, "Dynamic spectrum leasing: A new paradigm for spectrum sharing in cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 5, pp. 2328–2339, Jun. 2010.

[13] C. Xiong, G. Y. Li, L. Lu, D. Feng, Z. Ding, and H. Mitchell, "Spectrum trading for efficient spectrum utilization," *EAI Trans. Wireless Spectrum*, 2014, submitted for publication.

[14] D. Feng, L. Lu, Y. Yuan-Wu, G. Y. Li, G. Feng, and S. Li, "Device-to-device communications underlaying cellular networks," *IEEE Trans. Commun.*, vol. 61, no. 8, pp. 3541–3551, Aug. 2013.

[15] D. B. West, *Introduction to Graph Theory*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice Hall, 2001.

[16] E. A. Jorswieck, "Stable matchings for resource allocation in wireless networks," presented at the IEEE Int. Conf. Digital Signal Process. (DSP), Corfu, Jul. 2011.

[17] B. Holfeld, R. Mochaourab, and T. Wirth, "Stable matching for adaptive cross-layer scheduling in the LTE downlink," presented at the Veh. Technol. Conf. (VTC), Dresden, Jun. 2013.

[18] A. Leshem, E. Zehavi, and Y. Yaffe, "Multichannel opportunistic carrier sensing for stable channel access control in cognitive radio systems," *IEEE J. Sel. Area Commun.*, vol. 30, no. 1, pp. 82–95, Jan. 2012.

[19] S. Bayat, R. H. Y. Louie, B. Vucetic, and Y. Li, "Dynamic decentralised algorithms for cognitive radio relay networks with multiple primary and secondary users utilising matching theory," *Trans. Emerging Tel. Tech.*, vol. 24, pp. 486–502, May 2013.

[20] A. Osseiran, V. Braun, T. Hidekazu, P. Marsch, H. Schotten, H. Tullberg, M. A. Uusitalo, and M. Schellman, "The foundation of the mobile and wireless communications system for 2020 and beyond challenges, enablers and technology solutions," presented at the Proc. IEEE Veh. Technol. Conf. (VTC), Dresden, Jun. 2013.

[21] P. Floreen, P. Kaski, V. Polishchuk, and J. Suomela, "Almost stable matchings by truncating the Gale-Shapley algorithm," *Algorithmica*, vol. 28, no. 1, pp. 102–118, 2010.

[22] M. Xiao, N. B. Shroff, and E. K. P. Chong, "A utility-based power-control scheme in wireless cellular systems," *IEEE/ACM Trans. Netw.*, vol. 11, no. 2, pp. 210–221, Apr. 2003.

[23] T. Alpcan, T. Basar, and R. Srikant, "CDMA uplink power control as a noncooperative game," *Wireless Netw.*, vol. 8, no. 6, pp. 659–670, Nov. 2002.

[24] Y. Y. A. Leshem and E. Zehavi, "Stable matching for channel access control in cognitive radio systems," in *Proc. Int. Workshop Cogn. Inf. Process. (CIP)*, Elba, Jun. 2010, pp. 470–475.

Lu Lu (S'09) received her B.S.E. degree and M.S.E. degree from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2007 and 2010, respectively. She then got her licentiate degree from Royal Institute of Technology (KTH) in 2011. She is currently working toward the Ph.D. degree with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA. Her research interests include MIMO, cooperative communications, and cognitive radio networks.

Dawei He received his B.S. degree and M.S. degree from East China Normal University (ECNU), Shanghai, China, in 2008 and 2011, respectively. Now he is pursuing his Ph.D. degree in the School of Mathematics, Georgia Institute of Technology, Atlanta, GA, USA. He is interested in graph theory, combinatorics, and algorithm.

Geoffrey Ye Li (S'93–M'95–SM'97–F'06) received his B.S.E. and M.S.E. degrees in 1983 and 1986, respectively, from the Department of Wireless Engineering, Nanjing Institute of Technology, Nanjing, China, and his Ph.D. degree in 1994 from the Department of Electrical Engineering, Auburn University, Alabama.

He was a Teaching Assistant and then a Lecturer with Southeast University, Nanjing, China, from 1986 to 1991, a Research and Teaching Assistant with Auburn University, Alabama, from 1991 to 1994, and a Post-Doctoral Research Associate with the University of Maryland at College Park, Maryland, from 1994 to 1996. He was with AT&T Labs—Research at Red Bank, New Jersey, as a Senior and then a Principal Technical Staff Member from 1996 to 2000. Since 2000, he has been with the School of Electrical and Computer Engineering at the Georgia Institute of Technology as an Associate Professor and then a Full Professor. He is also holding a Cheung Kong Scholar title at the University of Electronic Science and Technology of China since 2006.

His general research interests include statistical signal processing and communications, with emphasis on cross-layer optimization for spectral- and energy-efficient networks, cognitive radios and opportunistic spectrum access, and practical issues in LTE systems. In these areas, he has published over 300 refereed journal and conference papers in addition to 26 granted patents. His publications have been cited around 20 000 times and he has been listed as the *World's Most Influential Scientific Mind*, also known as a *Highly-Cited Researcher*, by Thomson Reuters. He has been involved in editorial activities for over 10 technical journals for the IEEE Communications and Signal Processing Societies. He organized and chaired many international conferences, including technical program vice-chair of *IEEE ICC'03*, technical program co-chair of *IEEE SPAWC'11*, general chair of *IEEE GlobalSIP'14* and technical program co-chair of *IEEE VTC'16 (Spring)*. He has been awarded *IEEE Fellow* for his contributions to *signal processing for wireless communications* since 2006. He won 2010 *Stephen O. Rice Prize Paper Award* and *2013 WTC Wireless Recognition Award* from the IEEE Communications Society. He also received 2013 *James Evans Avant Garde Award* and 2014 *Jack Neubauer Memorial Award* from the *IEEE Vehicular Technology Society*. Recently, he won 2015 *Distinguished Faculty Achievement Award from the School of Electrical and Computer Engineering, Georgia Tech*.

Xingxing Yu received his B.S. (1983) and M.S. (1986) degrees in mathematics from East China Normal University, and Ph.D. (1990) degree in mathematics from Vanderbilt University. He joined the faculty of Georgia Institute of Technology in 1990, and is currently a professor in the School of Mathematics. His research interests are in discrete mathematics and its applications. He has served on the editorial board of several professional journals.

Graph-Based Robust Resource Allocation for Cognitive Radio Networks

Lu Lu, *Student Member, IEEE*, Dawei He, Geoffrey Ye Li, *Fellow, IEEE*, and Xingxing Yu

***Abstract*—Cognitive radio** **(CR) technology is promising for next generation wireless networks. It allows unlicensed secondary users to use the licensed spectrum bands as long as they do not cause unacceptable interference to the primary users who own those bands. To efficiently allocate resources in CR networks, stable resource allocation based on graph theory is investigated, which takes all users' preferences into account. In this paper, we focus on improving robustness of the stable matching based resource allocation. A truncated scheme generating almost stable matchings is first investigated. Based on the properties of the truncated scheme, two types of edge-cutting algorithms, called *direct edge-cutting* (DEC) and *Gale-Shapley based edge-cutting* (GSEC), are developed to improve resource allocation robustness to the channel state information variation. To mitigate the problem that certain secondary users may not be able to find suitable resources after edge-cutting, *multi-stage* (MS) algorithms are then proposed. Numerical results show that the proposed algorithms are robust to the channel state information variation.**

***Index Terms*—Cognitive radio (CR) network, stable matching, robustness, almost stable, edge-cutting.**

I. Introduction

With dramatically increasing demands of high-data rate applications in wireless communications, more and more spectrum resources are needed. Currently, the spectrum bands are licensed to some operators, organizations or companies, leading to the underutilization of some spectrum bands. To improve the spectrum efficiency, *cognitive radio* (CR) technology has been developed [1], [2], where unlicensed users, also known as *secondary users* (SUs), are allowed to use the licensed spectrum bands as long as they do not generate unacceptable interference to the licensed users, also known as *primary users* (PUs). Due to coexistence of PUs and SUs, appropriate resource allocation is important to better utilize the limited spectrum resource.

Manuscript received April 17, 2014; revised November 05, 2014; accepted April 18, 2015. Date of publication May 13, 2015. This work was supported in part by the NSF under Grants 1247545 and 1443894. This work appeared in part in the *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP), Florence, Italy, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Hongwei Liu.

L. Lu and G. Y. Li are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250 USA (e-mail: lulu0528@gatech.edu; liye@ece.gatech.edu).

D. He and X. Yu are with the School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332-0250 USA (e-mail: dhe9@math.gatech.edu; yu@math.gatech.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2015.2432733

For resource allocation in CR networks, schemes based on different objectives have been studied [3]. Various schemes have been developed to maximize throughput [4]–[6] and energy efficiency [7], [8] of SUs. Resource allocation can also take the competition feature among SUs into account [9]–[11], where game theory is commonly used. Besides considering the SUs' performance only, the activities of PUs can also be considered jointly to further optimize resource allocation [12], [13].

As a useful and convenient tool, graph theory based schemes have been studied. Here, we focus on using bipartite graphs where SUs and PUs are treated as two partite sets and a resource allocation result can been seen as a matching of the corresponding bipartite graph. To optimize the sum/average utility of SUs, Hungarian algorithm for finding a maximum matching has been used [7], [14]. We can also take both PUs' and SUs' preferences into account and allocate resources based on the stable matching algorithm [15] due to its advantages. First, it incorporates fairness issues. Second, efficient algorithms, such as the *Gale-Shapley* (GS) algorithm, can be used to find a solution with polynomial complexity. Stable matching based resource allocation has been studied in [16]–[19]. In [16], a one-to-many stable matching and its properties have been studied. In [17], stable matching based on queue- and channel-aware lists for cross-layer scheduling has been investigated. In [18], tight lower and upper bounds for stable allocation performance have been derived. Moreover, stable matching with auction has been studied in [19].

Existing schemes on stable matching mainly focus on the design based on a fixed system condition while the robustness to the variation of the system conditions is seldom investigated. In future networks [20], many other types of users, such as machines, will communicate along with human users, and such networks are usually have a large number of nodes. Consequently, it is desirable to design resource allocation schemes that provide robust results with respect to changes in user and channel conditions. Thus, robust resource allocation should be considered.

In this paper, we investigate robust resource allocation based on graph theory, in particular, stable matching. By taking both SUs and PUs preferences into account, resource allocation is first performed based on the GS algorithm for stable matchings. To improve the robustness of resource allocation, a truncated GS algorithm is studied. Such algorithm has first been proposed and compared to the maximum utility matching in [21]. We will investigate the relationship between the matching produced by the truncated GS algorithm and the optimal stable matching when they are used in resource allocation in CR systems. Upper bounds on the number of rounds to reach almost stable resource allocation will be derived. Motivated by our theoretical results, we develop two types of edge-cutting algorithms, called *direct edge-cutting* (DEC) and *Gale-Shapley based edge-cutting* (GSEC), to further improve the robustness of our resource allocation schemes. To mitigate the problem that some SUs may not be able to find suitable resources to transmit after edge-cutting, *multi-stage* (MS) algorithms are developed. Below is a summary of main contributions of this paper.

- We develop and investigate a stable resource allocation scheme based on the GS algorithm.
- We study an almost stable resource allocation based on the truncated GS algorithm. We provide bounds on the numbers of rounds for the truncated GS algorithm to achieve ϵ-stable and $(1+\epsilon)$-approximation of the stable resource allocation, which show that the maximum degree of the SU pairs is the most important factor on the performance.
- Motivated by our theoretical results, we developed three types of edge-cutting algorithms, including DEC, GSEC, and MS, to improve the robustness of the resource allocation. Numerical results show that the robustness can indeed be improved.

The rest of this paper is organized as follows. In Section II, we introduce the system model. In Section III, resource allocation based on stable matching is discussed. In Section IV, a truncated algorithm guaranteeing almost stable matching is investigated. Two types of edge-cutting algorithms are proposed in Section V. In Section VI, multi-stage algorithms are developed. Numerical results are provided in Section VII to show the impact of different system parameters on the performance of the developed algorithms. Finally, Section VIII concludes the paper.

II. System Model

We consider an underlay CR network with multiple channels/bands, where PUs have priorities to use N spectrum channels/bands while M SU pairs want to transmit simultaneously. All channels are modeled as Rayleigh block fading channels. Without loss of generality, we assume the j-th PU uses the j-th spectrum band. As shown in Fig. 1, the channel between the i-th SU pair on the j-th spectrum band and the interference channel from the i-th SU transmitter to the j-th PU receiver are denoted as $h_{i,j}$ and $g_{i,j}$, respectively. The transmit power of the i-th SU transmitter on the j-th spectrum band is $P_{i,j}$. The noise power on all spectrum bands is assumed to be the same, denoted as σ^2. A centralized system is assumed, where the control center with CSI allocates resources. Furthermore, the control center will only assign a frequency band to a SU pair when the corresponding CSI is available, including both CSI between the SU pairs and of the interference channel from the SU transmitter to the PU receiver. Correspondingly, the number of spectrum bands available for the i-th SU pair is denoted as $\Delta_{s,i}$ and the number of SU pairs available for the j-th spectrum band is denoted as $\Delta_{p,j}$. We consider the scenario that only one SU pair is allowed on each spectrum band and each SU pair can only access at most one spectrum band.

To protect the PUs, the interference power generated by the SU pairs on the j-th spectrum band should be below a given threshold, that is,

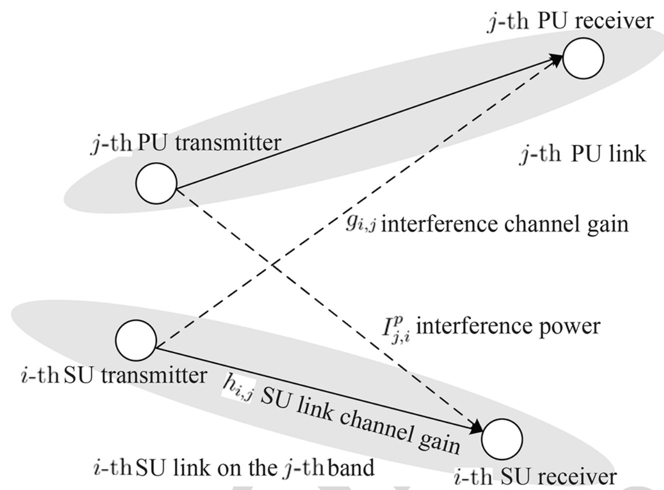

Fig. 1. System model.

$$I_{i,j} = P_{i,j}|g_{i,j}|^2 \leq I_{th}, \quad \forall i,j, \tag{1}$$

where I_{th} is the interference threshold[1]. Moreover, due to the amplifier capacity limit, each SU transmitter has a peak transmit power constraint, P, that is,

$$P_{i,j} \leq P, \quad \forall i. \tag{2}$$

If the i-th SU pair is assigned to the j-th spectrum band, its throughput can be expressed as

$$R_{i,j} = \log_2\left(1 + \frac{|h_{i,j}|^2 P_{i,j}}{\sigma^2 + I^p_{j,i}}\right), \tag{3}$$

where $I^p_{j,i}$ denotes the interference from the j-th PU transmitter to the i-th SU receiver. We assume that the interference powers, $I^p_{j,i}$, are known since the PUs' transmit powers remain the same with or without SU pairs' transmission and, hence, $I^p_{j,i}$ can be estimated in advance.

When performing resource allocation to the SU pairs, we also consider benefits to PUs by incorporating the concept of spectrum trading into the utility function design [13]. PUs will charge more from the SU pairs for providing better service. On the other hand, the performance of PUs will degrade if SU pairs generate strong interference. Therefore, while SU pairs improve their own performance by increasing their transmit powers, they should also get penalties for generating stronger interference. As in [22] and [23], to capture such features of both PUs and SU pairs, if the i-th SU pair uses the j-th spectrum band, the utility can be defined as a linear combination of the throughput of the i-th SU pair, $R_{i,j}$, and the interference generated to the j-th PU, $I_{i,j}$, expressed as

$$w_{i,j}(P_{i,j}) = c_s R_{i,j} - c_p I_{i,j}, \tag{4}$$

[1]Without loss of generality, we assume, for simplicity, that the interference threshold is the same on all the spectrum bands. The results can be directly extended to the system with different interference thresholds on different spectrum bands.

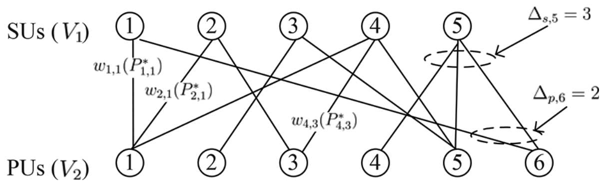

Fig. 2. Bipartite graph illustration with $M = 5$ and $N = 6$.

c_s and c_p are weight factors.

The transmit power, $P_{i,j}$, can be optimized to maximize the utility function in (4) subject to constraints (1) and (2). Since the utility function in (4) is a concave function of $P_{i,j}$, the optimal transmit power, $P^*_{i,j}$, is

$$P^*_{i,j} = \left(\min \left\{ \frac{c_s}{c_p |g_{i,j}|^2} - \frac{\sigma^2 + I^p_{j,i}}{|h_{i,j}|^2}, P, \frac{I_{th}}{|g_{i,j}|^2} \right\} \right)^+, \tag{5}$$

where $(x)^+ = \max\{0, x\}$. Then, if the i-th SU pair is allocated on the j-th spectrum band to transmit, the utility value of the i-th SU pair and the j-th PU can be expressed as $w_{i,j}(P^*_{i,j})$.

Even though (4) is used for resource allocation in this paper, the approaches developed here can easily be adapted for other utility functions.

III. Resource Allocation Based on Stable Matching

As indicated before, we will focus on resource allocation based on graph theory by taking preferences of both PUs and SU pairs into account. We first describe our resource allocation scheme based on stable matching [7] and then discuss its properties.

Stable matching based resource allocation can be described with the help of a *bipartite* graph, say G, with bipartition, V_1, V_2. See Fig. 2, where nodes in V_1 represent SU pairs and nodes in V_2 represent spectrum bands/PUs. As in Section II, the sizes of V_1, V_2 are denoted as M, N, respectively. An edge is put between the i-th SU pair in V_1 and the j-th PU in V_2 if the CSI of the i-th SU pair on the j-th spectrum band is known at the control center. Then, the common utility value of the i-th SU pair and the j-th PU, $w_{i,j}(P^*_{i,j})$, is assigned to the corresponding edge.

To define the preference lists for all SU pairs and PUs, we can regard G as a *complete* bipartite graph, i.e., every node in V_1 is connected to every node in V_2, by setting $w_{i,j}(P^*_{i,j}) = -\infty$ if the CSI of the i-th SU pair on the j-th spectrum band is not available. Furthermore, we assume that the numbers of spectrum bands/PUs and SU pairs are equal, i.e., $M = N$. If the number of SU pairs exceeds the number of spectrum bands, i.e., $M > N$, we can add $(M - N)$ virtual spectrum bands/PUs and edges between all these virtual PUs and all the SU pairs, and set the weights on these new edges to $-\infty$. Similar operation can be done when the number of spectrum bands exceeds the number of SU pairs.

The preference list for the i-th SU pair is defined as

$$\mathcal{L}^s_i = [j_1, \ldots, j_N], \tag{6}$$

where $j_1, \ldots, j_N$ is a permutation of $1, \ldots, N$ satisfying

$$w_{i,j_1}(P^*_{i,j_1}) \leq \cdots \leq w_{i,j_N}(P^*_{i,j_N}). \tag{7}$$

Similarly, the preference list for the j-th PU is defined as

$$\mathcal{L}^p_j = [i_1, \ldots, i_M], \tag{8}$$

where $i_1, \ldots, i_M$ is a permutation of $1, \ldots, M$ satisfying

$$w_{i_1,j}(P^*_{i_1,j}) \leq \cdots \leq w_{i_M,j}\left(P^*_{i_M,j}\right). \tag{9}$$

Our resource allocation procedure will depend on the preference lists of both SU pairs and PUs. To exploit stable matching from graph theory for resource allocation, more definitions are needed. A *matching* in a bipartite graph is a set of pairwise non-adjacent edges. Given a matching in a bipartite graph with partition sets V_1 and V_2, with V_1 (respectively, V_2) consisting of nodes represent SU pairs (respectively, PUs), if an SU pair is matched with a band/PU, we may allocate that band/PU to that SU pair. Hence, a matching in such a bipartite graph naturally gives a resource allocation in which each spectrum band has at most one SU pair and each SU pair accesses at most one spectrum band. We call a matching in a graph *perfect* if all the nodes in the graph are matched.

An edge in the bipartite graph, G, between the i-th SU pair and the j-th PU, denoted as (s_i, p_j), where s_i and p_j denote the i-th SU pair and the j-th PU, respectively, is said to be *unstable* relative to a matching $\mathcal{M}$ in G, if

1. $(s_i, p_j) \notin \mathcal{M}$,
2. s_i is unmatched by $\mathcal{M}$, or prefers p_j over its current match in $\mathcal{M}$, and
3. p_j is unmatched by $\mathcal{M}$, or prefers s_i over its current match in $\mathcal{M}$.

If there is no unstable edges relative to the matching, $\mathcal{M}$, then it is called *stable*.

Based on the above definition, it is desirable to find a resource allocation that corresponds to a stable matching. This can be done by using the GS algorithm [15]. In Table I, we give a version of the GS algorithm in which PUs propose to SU pairs (see Step 2). However, the roles of PUs and SU pairs can be exchanged. So the GS algorithm can be presented in which SU pairs propose to PUs. Here, we focus on the PU-proposing version of the GS algorithm as shown in Table I.

By applying the PU-proposing version of the GS algorithm in Table I, we have the following result (see [15]).

Lemma 1: For the complete bipartite graph in which the number of SU nodes equals the number of PU nodes, the PU-proposing version of the GS algorithm generates a perfect stable matching. Moreover, for this matching, each PU is matched to the best possibility among all stable matchings, and each SU pair is matched to the worst possibility among all stable matchings.

According to Lemma 1, we know that, if we apply the version of the GS algorithm in Table I, each PU will be matched to the best possibility among all stable matchings. That means, the weight in the resulting stable matching for each PU is maximum among all the stable matchings. Thus, the sum or average weight of the matching must be the maximum among all the stable matchings [15].

TABLE I
THE GALE-SHAPLEY PREFERENCE-BASED CHANNEL ALLOCATION ALGORITHM

1: Initialize all the SU pairs and PUs to be free.
2: Each PU asks its most preferred SU pair which has not rejected it yet.
3: **if** Each SU pair receives exactly one request from the PUs. **then**
4: Stop and output the results as the channel allocation result.
5: **else**
6: Every SU pair receiving more than one requests chooses the PU that is highest on its preference list and rejects the other PUs.
7: **end if**
8: **if** All PUs have proposed to all SU pairs. **then**
9: Stop and output the results as the channel allocation result.
10: **else**
11: Goto Step 2.
12: **end if**

Theorem 2: Let G be a complete bipartite graph with partition sets V_1, V_2 such that V_1 (respectively, V_2) consists of SU (respectively, PU) nodes. Define the weight on edges according to (4) and (5), and define preference lists of nodes according to (6)–(9). Then, the GS algorithm in Table I produces a stable resource allocation that has maximum sum or average weight among all possible stable resource allocations.

As we pointed out before, there are two versions of GS algorithms, the PU- and the SU-proposing ones. In general, the PU- and SU-proposing versions may provide different matching results. Next, we show that, for our studied scenario, the stable matching produced by the version in Table I is unique, regardless of PU- or SU-proposing algorithm is used. By Lemma 1, SU pairs are matched to the worst possibility among all stable matchings after applying the PU-proposing version of the GS algorithm. On the other hand, if we apply the SU-proposing version of the GS algorithm, SU pairs will be matched to the best possibility among all stable matchings [15]. Since the weights on the edges are the same for both SU-proposing and PU-proposing versions, the maximum sum weight among all possible stable matchings is uniquely determined no matter which version of the GS algorithm we use. Hence, the best possibility and the worst possibility for each SU pair among all stable matchings are the same. Thus, we have

Theorem 3: The stable matching produced in the Theorem 2 is unique.

The conclusion on uniqueness in Theorem 3 is the same as in [24]; but the procedure used there is different from the above. Note that, for our scenario, this uniqueness conclusion implies that the same allocation results are obtained regardless of which version of GS algorithm is applied.

Note that Theorem 3 holds if (4) is replaced by a different utility function, as long as the preference lists of SU pairs and PUs are calculated from the weight function on the edges.

IV. ALMOST STABLE RESOURCE ALLOCATION

In the previous section, we proposed a stable resource allocation scheme based on the GS algorithm and discussed some of its properties. However, the stable matching results need not be robust to CSI variation. In fact, CSI variation of one channel may lead to a totally different stable matching result, which is not desirable in practice, especially when the number of nodes in the system is large. To measure the system robustness, we use the amount of variation of the resource allocation after CSI variation as a metric. Based on our definition, with CSI variation, the larger the resource allocation variation is, the less robust the system is to the CSI variation and vice versa.

To improve the robustness of the resource allocation scheme with respect to CSI variation, we consider a truncated GS algorithm. Instead of executing the GS algorithm fully to get a stable resource allocation, the truncated GS algorithm outputs a resource allocation result after executing a fixed number of rounds in the GS algorithm. Note that the GS algorithm, as in Table I, consists of a number of asking-accepting/rejecting rounds; during each round the PUs propose to the SU pairs and the SU pairs answer the proposals. Denote by T the number of rounds we execute the algorithm. From [21], based on the truncated GS algorithm, a change of CSI of one node will only affect the part of resource allocation within distance $2T$ from the node of the change. Comparing to the stable resource allocation, where a change of CSI of one node may affect all nodes in the system, allocation based on the truncated GS algorithm should be much more robust to the CSI variation.

The resulting resource allocation from the truncated GS algorithm may be unstable. To measure the stability of such resource allocation, we introduce the concept of ϵ-stable (or almost stable) matching. Let $\epsilon > 0$ be given. The matching, $\mathcal{M}$, in the graph, G, is said to be ϵ-*stable* if the number of unstable edges in G is at most $\epsilon|\mathcal{M}|$, where $|\mathcal{M}|$ is size of $\mathcal{M}$. Let Δ be the maximum degree of nodes (i.e., $\Delta = \max\{\Delta_{s,i}, \Delta_{p,j}\}$, $\forall i, j$, where $\Delta_{s,i}, \Delta_{p,j}$ are the numbers of edges at the i-th SU pair and the j-th PU, respectively). For the ϵ-stable matching, we have the following theorem from [21]:

Theorem 4: By executing at most $2 + \Delta^2/\epsilon$ rounds of the GS algorithm, we can find an ϵ-stable resource allocation.

The detailed proof for Theorem 4 can be found in [21]. Based on the procedure in [21], we can further bound the number of execution rounds by using $\Delta_s = \max\{\Delta_{s,i}\}$, the maximum degree among the nodes representing SU pairs only.

Theorem 5: By executing at most $2 + \Delta_s^2/\epsilon$ rounds of the GS algorithm, we can find an ϵ-stable resource allocation.

Our proof of Theorem 5 follows the same procedure as that of Theorem 4 by using the maximum degree among nodes for SU pairs, Δ_s, instead of the maximum degree of all nodes, Δ. It is clear that $\Delta_s \leq \Delta$. Hence, in practical scenarios, the bound in Theorem 5 is tighter than the bound in Theorem 4 from [21].

Besides the stability property, we also concern about the utility of resource allocation. Given one resource allocation, $\mathcal{M}$, its utility, denoted as $w_{\mathcal{M}}$, is defined as the sum utilities of PUs/SU pairs in $\mathcal{M}$. Let $\mathcal{S}$ denote the stable resource allocation produced by the GS algorithm, with $w_{\mathcal{S}}$ as the corresponding utility. A resource allocation, $\mathcal{M}$, is a $(1+\epsilon)$-*approximation* of the maximum-weight stable matching if its utility, $w_{\mathcal{M}}$, satisfies $(1+\epsilon)w_{\mathcal{M}} \geq w_{\mathcal{S}}$. For the truncated GS algorithm, we have the following conclusion regarding its utility and the detailed proof is in the Appendix.

Theorem 6: By executing at most $2+\Delta_s/\epsilon$ rounds of the GS algorithm, we can find a $(1+\epsilon)$-approximation of the maximum-weight stable resource allocation.

In [21], the result based on the truncated GS algorithm is compared with the maximum-weight resource allocation (instead of stable resource allocation). For the scenario studied by us, it is more reasonable to compare the result from truncation with other stable resource allocation.

The robustness of the resource allocation from the truncated GS algorithm is affected by T, the number of rounds the GS algorithm is executed. A change in the bipartite graph only affects the result in the radius-$2T$ neighbourhood of the changing point, instead of the whole resource allocation result [21]. Theorems 5 and 6 provide upper bounds on T to achieve ϵ-stability and $(1+\epsilon)$-approximation of the maximum-weight stable resource allocation, respectively. Both bounds depend only on ϵ and Δ_s. Given ϵ, decreasing Δ_s is expected to help decrease the required number of executing rounds, T. From these observations, several algorithms for decreasing Δ_s are developed in the next section in order to achieve robust designs.

V. Edge-Cutting for Robust Design

Based on our discussion in the previous section, the robustness of resource allocation obtained from the truncated GS algorithm depends on the number of rounds, say T, of the GS algorithm that we execute. Smaller T leads to higher robustness. Since the smallest T to achieve ϵ-stability or $(1+\epsilon)$-approximation depends on the instantaneous CSI, we instead focus on reducing the upper bounds on T.

From Theorems 5 and 6, the upper bounds on T to achieve ϵ-stability or $(1+\epsilon)$-approximation are related to Δ_s, the maximum number of available bands of any SU pair. Both bounds can be decreased by decreasing Δ_s. Thus, to improve robustness of resource allocation, a small Δ_s is preferable. It is possible that SU pairs only access CSI of a small number of PU bands, and Δ_s is small naturally. However, PUs, in order to improve their own utilities, may be willing to share their information to attract more SU pairs, as in the spectrum trading scenario [13]. In this case, the maximum number of bands available at SU pairs, Δ_s, may be large. Note that, the number of available bands for the i-th SU pair is the number of edges connected to it in the constructed bipartite graph.

For robust design, we propose edge-cutting algorithms to decrease the maximum number of edges connected to SU pairs, Δ_s. In other words, if the number of available bands at the i-th SU pair, $\Delta_{s,i}$, is larger than a given threshold, denoted as Δ_c^{max}, it will be asked to give up some bands before performing resource allocation. In the following, two different types of edge-cutting algorithms will be developed: *direct edge-cutting* (DEC) that cuts edges based on the preference lists, and *GS-Based Edge-Cutting* (GSEC) that cuts edges based on the GS algorithm.

A. Direct Edge-Cutting (DEC)

In this part, we propose ways to delete edges according to the preference lists of SU pairs and/or PUs. We will first describe a general approach that depends on a parameter $p \in [0,1]$, and we get two special approaches by letting $p=0$ or 1.

For the edge, (s_i, p_j), between the i-th SU pair s_i and the j-th PU p_j, let $t_{i,j}^p = k$ if s_i is the k-th element on p_j's preference list, and $t_{i,j}^s = l$ if p_j is the l-th element in s_i's preference list. We set a preference value on the edge (s_i, p_j) as a convex combination of $t_{i,j}^p$ and $t_{i,j}^s$, which can be expressed as $t_{i,j} = pt_{i,j}^s + (1-p)t_{i,j}^p$, where $0 \le p \le 1$. We let each SU pair keep $\Delta_c^{\max}$ (a fixed constant) edges that have the highest preference values. So in the resulting graph, each SU pair is incident with precisely $\Delta_c^{\max}$ edges. We call this process *SP-preferred DEC* (SP-DEC) algorithm, as it can be based on the preference lists of both SUs and PUs. The weight factors can be adjusted according to the priorities in the preference lists of SU pairs and PUs. We consider two extreme cases: $p=1$ and 0.

When $p=1$, then SP-DEC keeps $\Delta_c^{\max}$ edges at each SU pair that is at the top of its preference list; this approach is called *SU-preferred DEC* (S-DEC) as it only takes SU's preference lists into consideration.

When $p=0$, then SP-DEC keeps $\Delta_c^{\max}$ edges at each SU pair with the highest preference values (we can randomly choose the edges that have same preference value), and those edges occur at the top of PUs' preference lists; this approach is called *PU-preferred DEC* (P-DEC) as it takes PUs' preference lists into consideration.

B. GS-Based Edge-Cutting (GSEC)

The DEC procedures described above reduce the maximum number of edges connected to the SU pairs, and are easy to implement. However, they do not take the original GS matching process into account. Hence, the resource allocation results obtained from the GS algorithm after the DEC procedures may be quite different from the result obtained by applying the GS algorithm on the original graph. Moreover, it is possible that certain SU pairs with similar preference lists will compete with each other, so there is a chance that some of them are not allocated any resource.

For instance, consider a system with two PUs, p_1 and p_2, and two SU pairs, s_1 and s_2, where both p_1 and p_2 prefer s_1 to s_2, and both s_1 and s_2 prefer p_1 to p_2. Applying the GS algorithm in Table I, s_1 is assigned to p_1, and s_2 is assigned to p_2. However, the result can be different if we apply the GS algorithm after a DEC procedure. Set $\Delta_c^{\max}=1$, so that the maximum number of edges connected to each SU pair is one. Applying S-DEC, we obtain a graph in which both s_1 and s_2 are connected to p_1, but not to p_2. Then applying the GS algorithm to the new graph, s_1 is allocated to the channel of p_1, but s_2 cannot transmit.

In order to use DEC to obtain a resource allocation result closer to the result obtained by applying the GS algorithm, we propose an edge-cutting algorithm based on the GS algorithm (GSEC algorithm). In general, the edges involved in the first few rounds of the GS algorithm will be kept. However, minor changes are made to satisfy the requirement on the maximum number of edges connected to each node representing an SU pair. In order to make sure the maximum number of edges connected to each SU pair is $\Delta_c^{\max}$, each SU pair can reject at most $\Delta_c^{\max}-1$ edges during the GS process. We now describe the GSEC algorithm. See Table II.

First, a vector $\mathbf{d} \in \mathbb{N}^M$ is used in which the ith coordinate, d_i, records the number of edges rejected by the ith SU pair, s_i.

TABLE II
GS-BASED EDGE-CUTTING (GSEC) ALGORITHM

1: Initialize all SUs and PUs to be free. Define a vector $\mathbf{d} \in \mathbb{N}^M$ as an indicator with the i-th coordinate d_i recording the number of PUs rejected by s_i, the i-th SU pair. Set $d_i = 0$ for $i = 1, ..., M$. Define a vector $\mathbf{m} \in \mathbb{N}^M$ with the i-th element $m(s_i)$ recording the indices of the PUs connected to s_i. Initialize $m(s_i) = 0$ for $i = 1, ..., M$. Let $\mathcal{A}$ denote the indices of SU pairs which are available and initialize $\mathcal{A} = \{1, ..., M\}$. Define $\mathbf{E} \in \mathbb{N}^{M \times N}$ to record the edges that remain after edge-cutting. Set $e_{i,j} = 1$ if the edge between s_i and the j-th PU is kept. Initialize $\mathbf{E}$ to be a zero matrix.
2: Each PU asks its most preferred SU with indices in $\mathcal{A}$ that has not rejected it.
3: **for** $i \in \mathcal{A}$ **do**
4: Let $\mathcal{P}_i$ denote the set of the indices of those PUs which have proposed to s_i during this round.
5: **if** s_i is not proposed, $\mathcal{P}_i = \emptyset$ **then**
6: Do nothing.
7: **else if** s_i is only proposed by one PU, say the j-th PU and $m(s_i) = 0$ **then**
8: Accept the j-the PU and set $e_{i,j} = 1$.
9: **else**
10: **if** $m(s_i) \neq 0$ **then**
11: $\mathcal{P}_i = \mathcal{P}_i \cup m(s_i)$
12: **end if**
13: Accept the PU in $\mathcal{P}_i$ that is highest on s_i's preference list.

$$m(s_i) = \arg\max_{j \in \mathcal{P}_i} w_{i,j}(P^*_{i,j}).$$

Increase d_i by the number of PUs rejected by s_i during this round, that is $d_i = d_i + |\mathcal{P}_i| - 1$, where $|\mathcal{P}_i|$ denotes the number of elements in $\mathcal{P}_i$.
14: **if** $d_i \geq \Delta_c^{max} - 1$ **then**
15: $\mathcal{A} = \mathcal{A} - \{i\}$
16: **if** $d_i = \Delta_c^{max} - 1$ **then**
17: $e_{i,j} = 1$, for $j \in \mathcal{P}_i$.
18: **else**
19: Delete $d_i - (\Delta_c^{max} - 1)$ elements from $\mathcal{P}_i$ that are lowest on s_i's preference list. Then, set $e_{i,j} = 1$, for $j \in \mathcal{P}_i$.
20: **end if**
21: **end if**
22: **end if**
23: **end for**
24: **if** $\mathcal{A} = \emptyset$ **then**
25: Stop.
26: **else if** All PUs are matched or they have already proposed to all SUs in $\mathcal{A}$ **then**
27: For $i \in \mathcal{A}$, add $\Delta_c^{max} - 1 - d_i$ edges between s_i and those PUs at the top of s_i's preference list and then stop.
28: **else**
29: Go to step 3.
30: **end if**

Initially $d_i = 0$ for all i, and all SU pairs are involved. During each round of the GS algorithm in Table I, each PU asks its most preferred SU pair. At the end of a round, increase d_i by the number of edges/PUs rejected by s_i during this round, and update $\mathcal{P}_i$, the set of the indices of the PUs who have asked s_i during this round. There are three cases after executing a particular round of the GS algorithm: $d_i < \Delta_c^{\max} - 1$, $d_i = \Delta_c^{\max} - 1$, and $d_i > \Delta_c^{\max} - 1$.

If $d_i < \Delta_c^{\max} - 1$, then the number of PUs rejected by s_i has not reached the limit; in this case, keep all edges between s_i and the PUs with indices in $\mathcal{P}_i$, and s_i remains in the process for the next round.

If $d_i = \Delta_c^{\max} - 1$ then the number of PUs rejected by s_i reaches the limit; keep all edges between s_i and the PUs with indices in $\mathcal{P}_i$, but s_i is removed from the process.

Now assume $d_i > \Delta_c^{\max} - 1$. Then the number of PUs rejected by s_i exceeds the limit. So s_i will be removed from the process. To meet the requirement that $d_i \leq \Delta_s - 1$, we remove $d_i - (\Delta_c^{\max} - 1)$ PUs with indices in $\mathcal{P}_i$. There are different ways to remove PUs with indices in $\mathcal{P}_i$. A natural way is to remove those $d_i - (\Delta_c^{\max} - 1)$ PUs with the lowest ranks in the preference list of s_i. Thus, the process keeps the PUs on the top of s_i's preference list, and it is independent of other PUs and SU pairs.

The GSEC process stops when all SU pairs are removed from the process or when the GS algorithm stops. It is possible that the GS algorithm stops before all SU pairs are removed. In this case, all PUs and SU pairs are matched. If this case occurs, $\Delta_c^{\max} - 1 - d_i$ edges can be added for each s_i that is still in the process.

Therefore, when the GSEC process stops, each SU pair will have $\Delta_c^{\max}$ edges. The detailed algorithm is given in Table II. Note that, during the process of the GSEC, a matching will be generated as well. This matching will be the same as the one if we execute the GS algorithm on the bipartite graph from the GSEC. Thus, we can use the matching generated from the GSEC directly for resource allocation without running the GS algorithm again.

To illustrate the difference between GSEC and DEC, we apply GSEC algorithm to the example at the beginning of this subsection. Both PUs, p_1 and p_2, ask the first SU pair, s_1, during the first round. Since the s_1 can only keep $\Delta_c^{\max} = 1$ edge and it prefers p_1 over p_2, it keeps only the edge (s_1, p_1), and s_1 leaves the process. In the second round, the second PU, p_2, asks the second SU, s_2, and keeps the edge (s_2, p_2). The GS algorithm stops. (Since both s_1 and s_2 have $\Delta_c^{\max} = 1$ edges, no extra edges need to be added.) Based on the graph after GSEC, s_1 will be allocated to the channel of p_1 while s_2 will be on the channel of p_2. This resource allocation result is the same as the original GS algorithm while the maximum number of edges connected to each SU pair is one after the GSEC. Therefore, resource allocation based on the GSEC process might give a result that is closer to the original GS algorithm while reducing the maximum number of edges connected to individual SU pairs.

Remark 1: Essentially, both DEC and GSEC shorten the preference lists of SU pairs and PUs. Any change on the deleted edges has no impact on the resource allocation results. This also explains why the edge-cutting algorithms improve robustness of the resource allocation.

Remark 2: After edge-cutting, some of SU pairs may not be able to find suitable channels. Consider the example at the beginning of Section V-B, only the first SU pair can transmit after DEC while both of them can transmit without edge-cutting. In the next section, a multi-stage edge-cutting algorithm will be proposed to mitigate this problem.

Remark 3: In addition to improving the robustness of resource allocation, edge-cutting also reduces the computational complexity.

VI. Multi-Stage Edge-Cutting

In the previous section, we proposed two different types of edge-cutting algorithms (DEC and GSEC) to reduce the maximum number of channels available at individual SU pairs after edge-cutting, $\Delta_c^{\max}$, to improve the robustness of resource allocation with respect to the CSI variation. We expect that resource allocation results are more robust when $\Delta_c^{\max}$ is made smaller. However, if $\Delta_c^{\max}$ is too small (so the number of channels available at each SU pair is small), some SU pairs may not be able to find suitable channels to transmit. To mitigate this problem, we propose a *multi-stage* (MS) edge-cutting algorithm.

The basic idea of the MS algorithm is to operate the GS algorithm after DEC or GSEC algorithms several times and to adjust the edge-cutting result during the procedure to increase the number of SU pairs that are allocated channels to transmit. The procedure is called MS algorithm, briefly described as follows. First, the DEC or the GSEC algorithm is operated on the original graph and then the GS algorithm. (This is the same as the procedure in Section V.) We then output the corresponding edge-cutting result only for the SU pairs who have been assigned channels. Remove the matched SU pairs, the corresponding matched PU nodes, and the edges among them from the procedure. Denote the sets of remaining SU pairs and PUs as $\mathcal{S}_r$ and $\mathcal{P}_r$, respectively. Note that no SU pair in $\mathcal{S}_r$ was asked by any PUs in the previous stage since if one SU pair was asked, it was assigned a channel and would not be in $\mathcal{S}_r$. Thus, the nodes in $\mathcal{S}_r$ has no impact on the previous stage. Then, we can ignore all previous procedures related to nodes in $\mathcal{S}_r$ and conduct the edge-cutting and the resource allocation again based on the subgraph of the original graph constituting of all nodes in $\mathcal{S}_r \bigcup \mathcal{P}_r$, and corresponding edges. Through this way, the maximum number of edges connected to each SU pair involving in the MS algorithm is still $\Delta_c^{\max}$. Repeat the procedure until all nodes are removed. The detailed procedure is given in Table III.

Moreover, like the GSEC algorithm, the MS algorithm will generate a matching during the process. This matching will be the same as the one if we execute the GS algorithm on the output bipartite graph from the MS algorithm and thus, can be used directly for resource allocation.

In general, by using the MS algorithm, the number of SU pairs which can transmit increases while the maximum number of edges connected to each SU pair kept in the graph is still $\Delta_c^{\max}$.

TABLE III
Multi-Stage (MS) Edge-Cutting Algorithm

1: Initialize the input graph (V_1^i, V_2^i, E^i) as $V_1^i \leftarrow V_1$, $V_2^i \leftarrow V_2$ and $E^i \leftarrow E$, where V_1, V_2, and E denote all SU pairs, PUs, and all edges among them, respectively, from the original graph. Initialize the output graph (V_1^o, V_2^o, E^o) to be empty.
2: The DEC or GSEC algorithm is conducted on (V_1^i, V_2^i, E^i) to reduce the number of edges connected to individual SU pairs. The resulting graph is denoted as (V_1^e, V_2^e, E^e), where V_1^e, V_2^e, and E^e denote the SU pairs, PUs and the remaining edges among SU pairs and PUs after edge-cutting on (V_1^i, V_2^i, E^i), respectively.
3: The GS algorithm is operated on (V_1^e, V_2^e, E^e).
4: **if** Each SU pair is allocated a channel for transmission, **then**
5:    Stop the algorithm and set the output graph (V_1^o, V_2^o, E^o) as $V_1^o \leftarrow V_1^o \cup V_1^e$, $V_2^o \leftarrow V_2^o \cup V_2^e$, and $E^o \leftarrow E^o \cup E^e$
6: **else**
7:    Let $\mathcal{S}_r$ denote the set of indices of those SU pairs which are not yet assigned a channel, and $\mathcal{P}_r$ denote the set of indices of those PUs which have no matched SU pair. Update $V_1^o \leftarrow V_1^o \cup (V_1^e - \mathcal{S}_r)$, $V_2^o \leftarrow V_2^o \cup (V_2^e - \mathcal{P}_r)$, and add edges among $V_1^e - \mathcal{S}_r$ and $V_2^e - \mathcal{P}_r$ in E^e to E^o.
8:    Update $V_1^i \leftarrow \mathcal{S}_r$, $V_2^i \leftarrow \mathcal{P}_r$, and E^i as the edges among $\mathcal{S}_r$ and $\mathcal{P}_r$ based on the original graph (V_1, V_2, E). The preference list of each member of $V_1^i \cup V_2^i$ will simply be the restriction of the original preference list to $V_1^i \cup V_2^i$.
9:    **if** All nodes have empty preference list, **then**
10:       Stop the algorithm.
11:    **else**
12:       Go to item 2.
13:    **end if**
14: **end if**

VII. Numerical Results

In this section, numerical results are presented to show the performance of the proposed algorithms in the application scenario where the number of users is large and the number of channels with CSI variation is small. Here, we consider the case with 200 SU pairs and 200 PUs, i.e., $M = N = 200$ and assume all CSI is known. Results are averaged after 20 000 trials. For each trial, algorithms are executed once based on the original CSI and then, conducted second time by changing CSI of 5 SU pairs. The utility gap shown in Figs. 3(b), 4(b), and 5(b) is defined as $(w_\mathcal{S} - w)/w_\mathcal{S}$, where $w_\mathcal{S}$ is the sum utility based on the GS algorithm and w is the utility based on the proposed algorithm. The SU allocation variation is defined as the number of SU pairs with different allocation results after CSI change. The SU allocation variation saving in Figs. 3(a), 4(a), and 5(a) is the relative value compared to the GS algorithm. We also show the allocation difference between each proposed algorithm and the GS algorithm, which is defined as the number of SU pairs with different allocation results between a proposed algorithm and the GS algorithm. For a resource allocation scheme, smaller resource allocation variation means higher robustness to the CSI variation. For all results, we set the *signal-to-noise ratios* (SNRs) between any SU pair and any interference channel from SU transmitter to the PU receiver as 0 dB and -10 dB, respectively. Interference threshold is -10 dB and the maximum transmit power is 10 dB.

A. Single-Stage Truncation or Edge-Cutting

Figs. 3(a) and (b) show the impact of the maximum number of available bands for each SU pair, $\Delta_c^{\max}$, on allocation variation saving and utility, where $\epsilon = 0.1$. Fig. 3(a) shows that the reduction in $\Delta_c^{\max}$ increases the robustness of resource allocation for all edge-cutting algorithms. Even though smaller $\Delta_c^{\max}$ leads to a larger utility gap from the GS algorithm as shown in Fig. 3(b), the saving on SU allocation variation is significant compared to that of the utility gap.

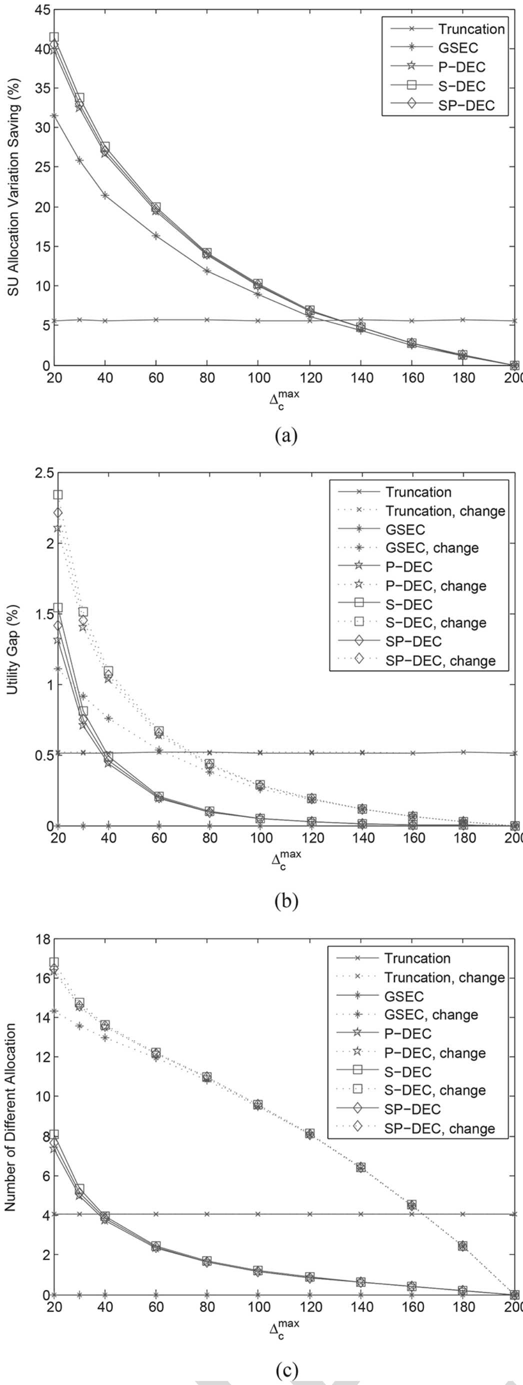

Fig. 3. Performance for algorithms with truncated or edge-cutting. (a) SU allocation variation saving. (b) Utility gap. (c) Resource allocation difference.

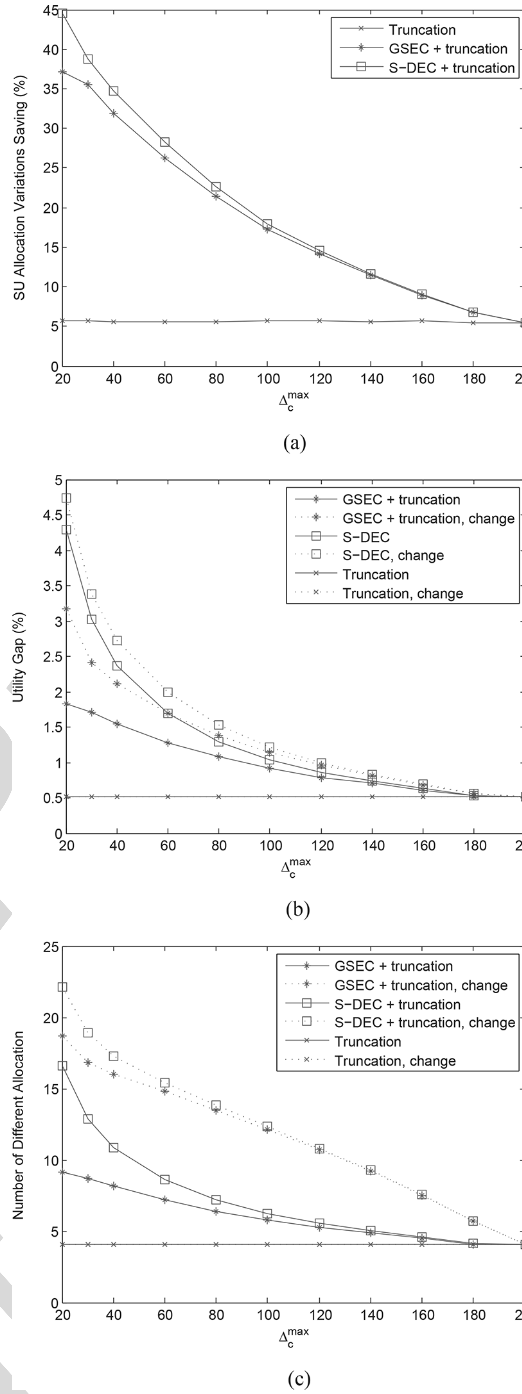

Fig. 4. Performance for algorithms with truncation and edge-cutting. (a) SU allocation variation saving. (b) Utility gap. (c) Resource allocation difference.

Next, let us compare the performance of the truncated GS algorithm and the edge-cutting algorithms. Fig. 3(a) shows that when $\Delta_c^{\max}$ is small, the edge-cutting algorithms may provide higher resource allocation robustness than the truncated GS algorithm. When $\Delta_c^{\max}$ is large, the truncated GS algorithm can provide more robust results. For the utility gap in Fig. 3(b), except for the GSEC before CSI change, it is larger for edge-cutting algorithms than for the truncated GS algorithm when $\Delta_c^{\max}$ is small, and the other way around when $\Delta_c^{\max}$ is large. Executing the GSEC algorithm before CSI change can provide a negligible performance gap from the GS algorithm. This is due to the fact the edges kept by the GSEC algorithm are based on the GS algorithm, and almost all edges involved in the GS algorithm are kept. The advantage cannot be maintained after CSI changes. But, the GSEC algorithm still provides smaller utility gap than DEC algorithms. Different DEC algorithms have similar performance.

As we discussed in Section V-B, it is expected that resource allocation results from GSEC are closer to those from the GS algorithm than those from DEC, which is illustrated in Fig. 3(c). GSEC provides the same resource allocation result as the GS

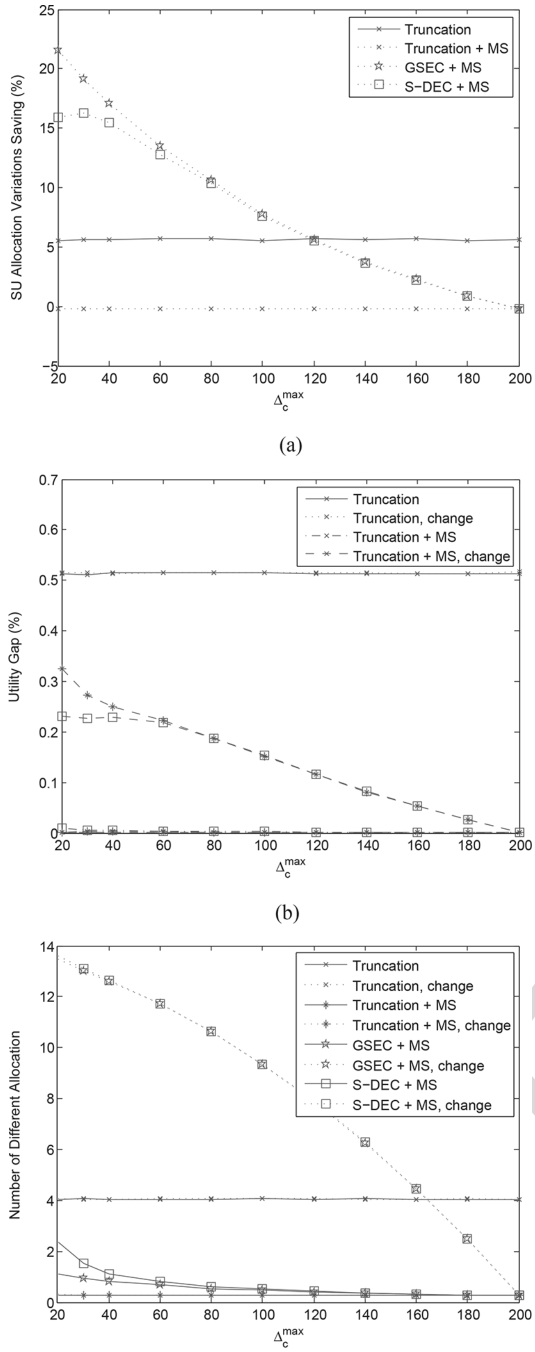

Fig. 5. Performance for different multi-stage algorithms. (a) SU allocation variation saving. (b) Utility gap. (c) Resource allocation difference.

algorithm even when the maximum available channels allowed at each SU pair is only 10% of all channels, i.e., $\Delta_c^{\max} = 20$. However, the DEC algorithms always provide resource allocation results that are different from those from the GS algorithm. This advantage of GSEC is only significant for the case before CSI change. After CSI change, the performance gap becomes smaller since GSEC is operated based on the CSI before change. Comparing the results in Fig. 3(a), GSEC loses certain level of robustness in order to get resource allocation results closer to those from the GS algorithm. This is the case because the more the algorithm depends on the original graph, the more sensitive it is to change on the graph (i.e., the CSI variations).

B. Single-Stage Truncation and Edge-Cutting

Besides operating truncation or edge-cutting separately, these operations can be combined. For example, we can fist apply edge-cutting algorithms on the original graph. We then run the GS algorithm on the new graph, and stop when we obtain a resource allocation that is ϵ-stable or is an $(1+\epsilon)$-approximation compared with the results on the original graph. Figs. 4(a) and (b) show the SU allocation variation saving and utility performance, respectively. The results based on truncation only are also provided for comparison reason. Moreover, all three DEC algorithms have similar performance, and only results based on S-DEC are provided here. As $\Delta_c^{\max}$ increases, the performance curves of algorithms with edge-cutting and truncation converge to that of the truncated only algorithm, instead of to that of the original GS algorithm. Compared with the results in Figs. 3(a) and (b), utilizing both edge-cutting and truncation can result in higher SU allocation variation saving accompanied a larger utility gap. The resource allocation difference is shown in Fig. 4(c). For all edge-cutting algorithms with truncation, the number of different resource allocations is larger compared to that from the edge-cutting only case and the performance curve converges to the truncation only case as $\Delta_c^{\max}$ increases.

C. Multi-Stage Truncation and Edge-Cutting

In this part, the performance of the MS algorithm is presented. Figs. 5(a) and (b) show the utility gap and SU allocation variation saving, respectively. With the help of the MS algorithm, the utility gap between the MS algorithm and the original GS algorithm is significantly reduced compared with the single-stage algorithm. The gap based on the CSI before the change is negligible while there is a small gap, less than 0.5%, after the CSI change. However, the SU allocation variation saving is smaller compared with the corresponding single-stage algorithms. Comparing results based on different edge-cutting algorithms, the utility gap for MS GSEC is slightly larger than the MS DEC algorithms while it can have higher SU allocation variation saving. This is different from the single-stage case. For MS GSEC, it can allocate more SU pairs in the first stage and the impact of the latter stages is smaller compared to the other DEC algorithms. Thus, it can keep higher SU allocation variation saving than the single-stage algorithm while the utility gap is larger as well.

Moreover, Fig. 5(c) shows the allocation difference between the MS algorithm and the original GS algorithm. Comparing with the results in Fig. 4(c), the number of allocation difference based on the MS algorithms is smaller than the one from the single-stage algorithm.

VIII. Conclusion

In this paper, we have studied robust resource allocation for CR networks. First, we develop a resource allocation scheme based on stable matching, which takes both SU pairs' and PUs' preferences into account. We then discuss the properties of almost stable resource allocation, which is robust to the CSI variation compared with stable resource allocation. Based on the properties of almost stable resource allocation, we propose the DEC and GSEC algorithms to further improve the robustness of the resource allocation scheme. The MS algorithms are then discussed to compensate for possible losses from the DEC and GSEC algorithms. Numerical results show that our proposed schemes are robust to the CSI variations.

Appendix A
Proof of Theorem 6

The proof of Theorem 6 follows an argument in [21]. Let T denote the number of rounds we execute the GS algorithm. For $0 \le n \le T$, let $\mathcal{M}_n$ denote the matching at the end of n-th round. The i-th SU pair is denoted as s_i, and the j-th PU is denoted by p_j. Define w as a weight function on edge. So $w_{s_i,p_j}(P^*_{i,j})$ is the weight of edge between s_i and p_j. If s_i is matched at the end of the n-th round, we use $m_n(s_i)$ to denote the PU matched to s_i. Let $C_n(p_j)$ denote the set of those SU pairs which have not rejected p_j at the end of n-th round (which is called the candidate set of p_j).

During any round, each PU removes at most one SU node from its candidate set, since each PU proposes to at most one SU pair during any round. The edge (s_i, p_j) is called *lost* if s_i is removed from the candidate set of p_j. We use L_n to denote the set of the lost edges by the end of the n-th round. For $\mathcal{M}_n$, the matching at the end of the n-th round, we define the weight of s_i as $w_n(s_i) = w_{s_i,m_n(s_i)}$ (if s_i is matched in $\mathcal{M}_n$) and $w_n(s_i) = 0$ (if s_i is unmatched in $\mathcal{M}_n$), where $w_{s_i,m_n(s_i)} := w_{s_i,m_n(s_i)}(P^*_{s_i,m_n(s_i)})$, see (4) and (5). Then, the total weight of the matching $\mathcal{M}_n$ can be expressed as

$$w_{\mathcal{M}_n} = \sum_{s_i \in V_1} w_n(s_i), \tag{10}$$

where V_1 denotes the set of SU nodes. Moreover, the total weight of the edges in L_n, denoted as w_{L_n}, is defined as the sum of weights of the edges in L_n.

We define the potential of p_j, denoted as $f_n(p_j)$, as follows. If p_j is matched in $\mathcal{M}_n$ or $C_n(p_j) = \emptyset$ then set $f_n(p_j) = 0$. Otherwise, set

$$f_n(p_j) = \max_{s_i \in C_n(p_j)} w_{s_i,p_j}, \tag{11}$$

which is the maximum weight of edges connecting p_j to the SU pairs in $C_n(p_j)$. The total potential of the matching $\mathcal{M}_n$ at the end of n-th round is the sum of the potentials of all PUs, that is,

$$f_{\mathcal{M}_n} = \sum_{p_j \in V_2} f_n(p_j), \tag{12}$$

where V_2 denotes the set of PU nodes.

Lemma 7: For $n \ge 2$, $f_{\mathcal{M}_n} \le w_{L_n} - w_{L_{n-1}}$.

Proof: Let p_j denote the j-th PU node. If $f_n(p_j) > 0$ then p_j is not matched by $\mathcal{M}_n$ and $C_n(p_j) \ne \emptyset$. Thus, during the n-th round, p_j is rejected by some SU pair, say s_k (the k-th SU node), and the edge (s_k, p_j) is lost. So $(s_k, p_j) \in L_n - L_{n-1}$. Note that p_j prefers s_k over other SU pairs in $C_n(p_j)$, which means, $f_n(p_j) \le w_{s_k,p_j}$. Hence,

$$f_{\mathcal{M}_n} = \sum_{p_j \in V_2} f_n(p_j) \le \sum_{p_j \in V_2} w_{s_k,p_j} \le w_{L_n} - w_{L_{n-1}},$$

and the lemma is proved. ■

Lemma 8: For $n \ge 2$, $f_{\mathcal{M}_n} \le f_{\mathcal{M}_{n-1}}$.

Proof: Since $f_{\mathcal{M}_n} = \sum_{p_j \in V_2} f_n(p_j)$ and $f_{\mathcal{M}_{n-1}} = \sum_{p_j \in V_2} f_{n-1}(p_j)$, we compare $f_n(p_j)$ and $f_{n-1}(p_j)$ for each $p_j \in V_2$.

If p_j is matched in both $\mathcal{M}_{n-1}$ and $\mathcal{M}_n$, then by definition, $f_n(p_j) = f_{n-1}(p_j) = 0$.

If p_j is matched in $\mathcal{M}_n$, but not in $\mathcal{M}_{n-1}$, then by definition, $f_n(p_j) \le f_{n-1}(p_j)$.

Now assume that p_j is matched to some $s_i \in V_1$ in $\mathcal{M}_{n-1}$ but p_j is not matched in $\mathcal{M}_n$. Then $f_{n-1}(p_j) = 0$ and $f_n(p_j) = w_{s_k,p_j}$, where s_k is such that $w_{s_k,p_j} = \max_{s_t \in C_n(p_j)} w_{s_t,p_j}$. Moreover, since in the n-th round, s_i must reject p_j and accept another PU, say $p_{j'}$ with $j \ne j'$, we have $f_{n-1}(p_{j'}) = w_{s_i,p_{j'}}$ and $f_n(p_{j'}) = 0$. Thus,

$$\begin{aligned} f_n(p_j) - f_{n-1}(p_j) &= w_{s_k,p_j} \le w_{s_i,p_j} \le w_{s_i,p_{j'}} \\ &= f_{n-1}(p_{j'}) - f_n(p_{j'}), \end{aligned}$$

which is equivalent to

$$f_n(p_j) + f_n(p_{j'}) \le f_{n-1}(p_j) + f_{n-1}(p_{j'}).$$

By summing up over all PUs, we have $f_{M_n} \le f_{M_{n-1}}$. ■

Lemma 9: For $n \ge 2$, $w_{L_n} \ge (n-1) f_{M_n}$.

Proof: By Lemmas 7 and 8, $w_{L_n} = \sum_{t=2}^{n} (w_{L_t} - w_{L_{t-1}}) \ge \sum_{t=2}^{n} f_{M_t} \ge (n-1) f_{M_n}$. ■

Lemma 10: For all $n \ge 2$, $w_{L_n} \le (\Delta_s - 1) w_{M_n}$.

Proof: Let s_i denote the i-th SU pair and p_j denote the j-th PU. If $(s_i, p_j) \in L_n$ (i.e., lost by the end of the n-th round), then s_i is matched to a better choice, $m_n(s_i)$, by the end of the n-th round. That is, $w_{s_i,p_j} < w_{s_i,m_n(s_i)}$. Since s_i can lose at most $\Delta_s - 1$ edges, the total weight of all lost edges adjacent to s_i is at most $(\Delta_s - 1) \cdot w_{s_i,m_n(s_i)}$. Hence, by summing over all SU pairs, we obtain $w_{L_n} \le (\Delta_s - 1) w_{M_n}$. ■

By Lemmas 9 and 10, we have

Lemma 11: For any real $\gamma > 0$ and integer $n \ge 1 + (\Delta_s - 1)/\gamma$, $f_{M_n} \le \gamma w_{M_n}$.

Next, we present the proof of Theorem 6.

Proof: Let $\mathcal{S}$ be the unique stable matching produced by the GS algorithm. We use s_i to denote the i-th SU pair and p_j to denote the j-th PU. Suppose $(s_i, p_j) \in \mathcal{S}$.

Note that when p_j is matched to s_k in $\mathcal{M}_n$ for some k (possibly $k = i$), p_j prefers s_k over s_i and $w_{s_i,p_j} \le w_{s_k,p_j} = w_n(p_j)$. Also note that when p_j is not matched in $\mathcal{M}_n$, then p_j has not asked s_i, $s_i \in C_n(p_j)$, and $w_{s_i,p_j} \le f_n(p_j)$.

Hence,

$$w_{s_i,p_j} \leq f_n(p_j) + w_n(p_j)$$

and therefore

$$\begin{aligned} w_{\mathcal{S}} &= \sum_{(s_i,p_j)\in\mathcal{S}} w_{s_i,p_j} \\ &\leq \sum_{p_j\in V_2} (f_n(p_j) + w_n(p_j)) \\ &= f_{\mathcal{M}_n} + w_{\mathcal{M}_n}. \end{aligned}$$

From Lemma 11, we conclude that

$$w_{\mathcal{S}} \leq (1+\epsilon) w_{M_n}$$

whenever $n \geq 1 + (\Delta_s - 1)/\epsilon$.

Thus, Theorem 6 follows by choosing $T = \lfloor 2 + \Delta_s/\epsilon \rfloor$. ∎

REFERENCES

[1] J. Mitola and G. Q. Maguire, "Cognitive radio: Making software radios more personal," *IEEE Personal Commun.*, vol. 6, no. 4, p. 13, Aug. 1999.

[2] L. Lu, X. Zhou, U. Onunkwo, and G. Y. Li, "Ten years of research in spectrum sensing and sharing in cognitive radio," *EURASIP J. Wireless Commun. Netw.* vol. 28, Jan. 2012 [Online]. Available: http://jwcn.eurasipjournals.com/content/2012/1/28, [Online]. Available:

[3] E. Z. Tragos, S. Zeadally, A. G. Fragkiadakis, and V. A. Siris, "Spectrum assignment in cognitive radio networks: A comprehensive survey," *IEEE Commun. Survey Tutorials*, vol. 15, no. 3, pp. 1108–1135, 2013.

[4] L. Musavian and S. Aissa, "Capacity and power allocation for spectrum-sharing communications in fading channels," *IEEE Trans. Wireless Commun.*, vol. 8, no. 1, pp. 148–156, Jan. 2009.

[5] V. Asghari and S. Aissa, "Adaptive rate and power transmission in spectrum-sharing systems," *IEEE Trans. Wireless Commun.*, vol. 9, no. 10, pp. 3272–3280, Oct. 2010.

[6] X. Zhou, G. Y. Li, D. Li, D. Wang, and A. C. K. Soong, "Probabilistic resource allocation for opportunistic spectrum access," *IEEE Trans. Wireless Commun.*, vol. 9, no. 9, pp. 2870–2879, Sep. 2010.

[7] L. Lu, D. He, X. Yu, and G. Y. Li, "Energy efficient resource allocation for cognitive radio networks," presented at the IEEE Global Telecommun. Conf. (GLOBECOM'13), Atlanta, Dec. 2013.

[8] C. Xiong, L. Lu, and G. Y. Li, "Energy-efficient spectrum access in cognitive radios," *IEEE J. Sel. Area Commun.*, vol. 32, no. 3, pp. 550–562, Mar. 2014.

[9] Y.-E. Lin, K.-H. Liu, and H.-Y. Hsieh, "Design of power control protocols for spectrum sharing in cognitive radio networks: A game-theoretic perspective," presented at the IEEE Int. Conf. Commun. (ICC), Cape Town, May 2010.

[10] S. Buzzi and D. Saturnino, "A game-theoretic approach to energy-efficient power control and receiver design in cognitive CDMA wireless networks," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 1, pp. 137–150, Feb. 2011.

[11] Z. Tian, G. Leus, and V. Lottici, "Joint dynamic resource allocation and waveform adaptation for cognitive networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 2, pp. 443–454, Feb. 2011.

[12] S. K. Jayaweera, G. Vazquez-Vilar, and C. Mosquera, "Dynamic spectrum leasing: A new paradigm for spectrum sharing in cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 5, pp. 2328–2339, Jun. 2010.

[13] C. Xiong, G. Y. Li, L. Lu, D. Feng, Z. Ding, and H. Mitchell, "Spectrum trading for efficient spectrum utilization," *EAI Trans. Wireless Spectrum*, 2014, submitted for publication.

[14] D. Feng, L. Lu, Y. Yuan-Wu, G. Y. Li, G. Feng, and S. Li, "Device-to-device communications underlaying cellular networks," *IEEE Trans. Commun.*, vol. 61, no. 8, pp. 3541–3551, Aug. 2013.

[15] D. B. West, *Introduction to Graph Theory*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice Hall, 2001.

[16] E. A. Jorswieck, "Stable matchings for resource allocation in wireless networks," presented at the IEEE Int. Conf. Digital Signal Process. (DSP), Corfu, Jul. 2011.

[17] B. Holfeld, R. Mochaourab, and T. Wirth, "Stable matching for adaptive cross-layer scheduling in the LTE downlink," presented at the Veh. Technol. Conf. (VTC), Dresden, Jun. 2013.

[18] A. Leshem, E. Zehavi, and Y. Yaffe, "Multichannel opportunistic carrier sensing for stable channel access control in cognitive radio systems," *IEEE J. Sel. Area Commun.*, vol. 30, no. 1, pp. 82–95, Jan. 2012.

[19] S. Bayat, R. H. Y. Louie, B. Vucetic, and Y. Li, "Dynamic decentralised algorithms for cognitive radio relay networks with multiple primary and secondary users utilising matching theory," *Trans. Emerging Tel. Tech.*, vol. 24, pp. 486–502, May 2013.

[20] A. Osseiran, V. Braun, T. Hidekazu, P. Marsch, H. Schotten, H. Tullberg, M. A. Uusitalo, and M. Schellman, "The foundation of the mobile and wireless communications system for 2020 and beyond challenges, enablers and technology solutions," presented at the Proc. IEEE Veh. Technol. Conf. (VTC), Dresden, Jun. 2013.

[21] P. Floreen, P. Kaski, V. Polishchuk, and J. Suomela, "Almost stable matchings by truncating the Gale-Shapley algorithm," *Algorithmica*, vol. 28, no. 1, pp. 102–118, 2010.

[22] M. Xiao, N. B. Shroff, and E. K. P. Chong, "A utility-based power-control scheme in wireless cellular systems," *IEEE/ACM Trans. Netw.*, vol. 11, no. 2, pp. 210–221, Apr. 2003.

[23] T. Alpcan, T. Basar, and R. Srikant, "CDMA uplink power control as a noncooperative game," *Wireless Netw.*, vol. 8, no. 6, pp. 659–670, Nov. 2002.

[24] Y. Y. A. Leshem and E. Zehavi, "Stable matching for channel access control in cognitive radio systems," in *Proc. Int. Workshop Cogn. Inf. Process. (CIP)*, Elba, Jun. 2010, pp. 470–475.

Lu Lu (S'09) received her B.S.E. degree and M.S.E. degree from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2007 and 2010, respectively. She then got her licentiate degree from Royal Institute of Technology (KTH) in 2011. She is currently working toward the Ph.D. degree with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA. Her research interests include MIMO, cooperative communications, and cognitive radio networks.

Dawei He received his B.S. degree and M.S. degree from East China Normal University (ECNU), Shanghai, China, in 2008 and 2011, respectively. Now he is pursuing his Ph.D. degree in the School of Mathematics, Georgia Institute of Technology, Atlanta, GA, USA. He is interested in graph theory, combinatorics, and algorithm.

Geoffrey Ye Li (S'93–M'95–SM'97–F'06) received his B.S.E. and M.S.E. degrees in 1983 and 1986, respectively, from the Department of Wireless Engineering, Nanjing Institute of Technology, Nanjing, China, and his Ph.D. degree in 1994 from the Department of Electrical Engineering, Auburn University, Alabama.

He was a Teaching Assistant and then a Lecturer with Southeast University, Nanjing, China, from 1986 to 1991, a Research and Teaching Assistant with Auburn University, Alabama, from 1991 to 1994, and a Post-Doctoral Research Associate with the University of Maryland at College Park, Maryland, from 1994 to 1996. He was with AT&T Labs—Research at Red Bank, New Jersey, as a Senior and then a Principal Technical Staff Member from 1996 to 2000. Since 2000, he has been with the School of Electrical and Computer Engineering at the Georgia Institute of Technology as an Associate Professor and then a Full Professor. He is also holding a Cheung Kong Scholar title at the University of Electronic Science and Technology of China since 2006.

His general research interests include statistical signal processing and communications, with emphasis on cross-layer optimization for spectral- and energy-efficient networks, cognitive radios and opportunistic spectrum access, and practical issues in LTE systems. In these areas, he has published over 300 refereed journal and conference papers in addition to 26 granted patents. His publications have been cited around 20 000 times and he has been listed as the *World's Most Influential Scientific Mind*, also known as a *Highly-Cited Researcher*, by Thomson Reuters. He has been involved in editorial activities for over 10 technical journals for the IEEE Communications and Signal Processing Societies. He organized and chaired many international conferences, including technical program vice-chair of *IEEE ICC'03*, technical program co-chair of *IEEE SPAWC'11*, general chair of *IEEE GlobalSIP'14* and technical program co-chair of *IEEE VTC'16 (Spring)*. He has been awarded *IEEE Fellow* for his contributions to *signal processing for wireless communications* since 2006. He won 2010 *Stephen O. Rice Prize Paper Award* and *2013 WTC Wireless Recognition Award* from the IEEE Communications Society. He also received 2013 *James Evans Avant Garde Award* and 2014 *Jack Neubauer Memorial Award* from the *IEEE Vehicular Technology Society*. Recently, he won 2015 *Distinguished Faculty Achievement Award from the School of Electrical and Computer Engineering, Georgia Tech*.

Xingxing Yu received his B.S. (1983) and M.S. (1986) degrees in mathematics from East China Normal University, and Ph.D. (1990) degree in mathematics from Vanderbilt University. He joined the faculty of Georgia Institute of Technology in 1990, and is currently a professor in the School of Mathematics. His research interests are in discrete mathematics and its applications. He has served on the editorial board of several professional journals.