Competitive Distributed Spectrum Access in QoS-Constrained Cognitive Radio Networks

Ziqiang Feng, Ian Wassell
Computer Laboratory
University of Cambridge, UK
Email: {zf232, ijw24}@cam.ac.uk

Abstract—**Dynamic spectrum access is an important issue in cognitive radio networks (CRNs) as secondary users (SUs) can benefit from accessing the vacant licensed channels of primary users (PUs). In this paper, we consider the problem of competitive distributed spectrum access in CRNs with quality of service (QoS) constraints. We first propose a distributed matching algorithm (DMA) to handle spectrum access in a QoS-constrained CRN. We then propose a fast distributed matching algorithm (FDMA) for competitive spectrum access in a large-scale CRN. The distributed algorithms for the PU and SU are given separately for practical implementation. The performance and complexity of both algorithms are analyzed and demonstrated via simulation results.**

I. INTRODUCTION

Developments such as machine-to-machine communication and large-scale wireless sensor networks are placing growing demands on limited spectrum resources. However, most licensed spectrum is underutilized and stays unoccupied for most of the time [1]. Cognitive radio (CR) has been considered as a method that can enhance spectrum utilization by allowing unlicensed users to utilize licensed vacant channels without interfering with the licensed users.

In cognitive radio networks (CRNs), secondary (unlicensed) users (SUs) are allowed to opportunistically gain access to vacant channels of primary (licensed) users (PUs). However, SUs must not interfere with the PUs' utilization of their licensed channels. Once a channel is detected as vacant, an SU can send a request to access the channel to the appropriate PU. The SU can transmit to the intended receiver through the channel until the PU needs access to the channel. Since SUs can use the vacant channels within the PUs' allocated band, dynamic spectrum access has been considered as an important issue in CRNs. Most previous works focus on maximizing the network throughput. In [2], the authors formulate the spectrum access problem as a non-cooperative game. They propose a price-based iterative water-filling algorithm to maximize each SU's sum-rate. A spectrum trading approach is proposed in [3]. Based on contract theory, the authors maximize the PU utility by designing an optimal contract between PU and SUs. These methods require frequent message exchanges between PUs and SUs, which is not practical for large-scale CRNs.

Matching theory is proposed as a promising method for distributed resource allocation in large-scale networks [4] since matching algorithms can be implemented in a distributed way without compromising their performance [5]. Furthermore, matching algorithms need fewer message exchanges and iterations to terminate compared with some price-based game-theoretic approaches.

In [6], the authors are the first to introduce matching theory to solve the channel access problem. Algorithms based on the Gale-Shapley stable marriage theorem are proposed and analyzed. These algorithms are one-to-one matching algorithms. The authors in [7] propose a many-to-one matching algorithm considering the characteristics of both SU and PU. The proposed algorithm matches multiple SUs to one PU based on the PU's quota of SUs. However, both papers assume that the number of available channels is no fewer than the number of SUs. This assumption does not hold in CRNs where SUs are densely deployed with a limited number of PUs. In addition, the authors in [6], [7] mainly consider a CRN without QoS constraints and focus on maximizing the network throughput. In this paper, we eliminate unnecessary assumptions and consider the competitive spectrum access problem in large-scale CRNs with QoS constraints. We also give practical distributed algorithms that can be separately implemented in PUs and SUs for the competitive spectrum access problem.

In this paper, we first describe the system model for the QoS-constrained CRN. Then we analyze the QoS requirements and design a utility function that can be used in different scenarios (e.g., a CRN with or without QoS constraints). Based on matching theory, we first propose a distributed matching algorithm (DMA) that can be implemented in practical CRNs. DMA exploits the channel state information (CSI) of the vacant channels and achieves results that are comparable with those of a centralized solution. We show that DMA generates the optimal stable matching with the highest total utility among all stable matchings. To improve the algorithm efficiency for large-scale CRNs, we propose a fast distributed matching algorithm (FDMA). Simulation results show that DMA generally achieves better result than FDMA. However, FDMA takes fewer iterations to terminate in a large-scale CRN than does DMA. Both algorithms can handle the QoS constraints in a CRN by properly modifying the utility function.

The rest of the paper is organized as follows. In Section II, we describe the system model of the CRN and the QoS requirements. The problem is formulated in Section III. We propose our distributed spectrum access algorithms in Section IV. Simulation results and analysis are presented in
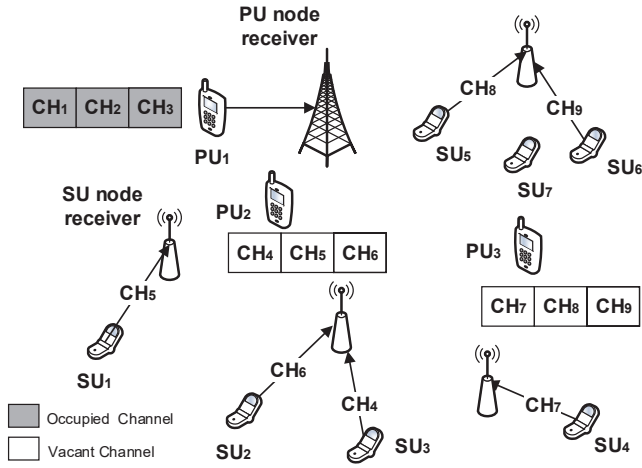
Fig. 1. Cognitive radio network

Section V. Finally, the paper is concluded in Section VI.

II. SYSTEM MODEL

We consider a QoS-constrained CRN consisting of M PUs and N SUs, as depicted in Fig. 1, where $M = 3$ and $N = 7$. We denote the set of PUs by $\mathcal{PU} = \{PU_1, PU_2, \ldots, PU_M\}$, and the set of SUs by $\mathcal{SU} = \{SU_1, SU_2, \ldots, SU_N\}$. Let K be the total number of licensed channels in the CRN ($K = 9$ in Fig. 1) where $\mathcal{CH} = \{CH_1, CH_2, \ldots, CH_K\}$ is the set of the licensed channels. Each $PU_i \in \mathcal{PU}$ has a set of K_i licensed channels available to it denoted by $\mathcal{CH}_i$, where $K_i = |\mathcal{CH}_i|$ and that it uses them all when communicating with the PU node receiver. Assuming that all PUs operate on different channels, then we know that $\mathcal{CH}_1, \mathcal{CH}_2, \ldots, \mathcal{CH}_M$ is a partition of $\mathcal{CH}$ and we have $K = \sum_{i=1}^{M} K_i$.

In CRNs, SUs are allowed to access the vacant licensed channels (i.e., those not in use by PUs) but must withdraw from the channels when a PU wishes to occupy the channel. SUs that are assigned vacant channels can communicate with corresponding SU node receivers under QoS constraints and thus improve the throughput of the network. In our model, we assume that each SU can only access one vacant channel of a PU at a time. The uplink and downlink communication of one SU and its SU node receiver share the same channel using a time division duplex (TDD) scheme. The TDD scheme is also utilized by the uplink and downlink communication of PUs and PU node receivers on their licensed channels.

A. QoS Requirements

In practical CRNs, SUs usually have different QoS requirements on for example data rate, delay and packet error rate. Let R_{SU_i}, D_{SU_i}, E_{SU_i} be the QoS requirements of SU_i's average data rate, delay and packet error rate, respectively. If SU_i occupies PU_k's vacant channel CH_j, the channel capacity $C_{i,j}$ can be written as

$$C_{i,j} = w_{i,j}\log_2\left(1 + \gamma_{i,j}\right), \quad (1)$$

where $w_{i,j}$ is the channel bandwidth and $\gamma_{i,j}$ is the signal to noise ratio (SNR) at SU_i's node receiver. Now $\gamma_{i,j}$ is given by

$$\gamma_{i,j} = \frac{p_i|h_{i,j}|^2}{d_i^\alpha \sigma^2}, \quad (2)$$

where p_i is the transmission power of SU_i, $|h_{i,j}|^2$ is the instantaneous fading channel gain between SU_i and its node receiver, d_i is the distance between SU_i and its receiver, α is the path loss exponent and σ^2 is the receive noise variance.

We consider the average data rate and the delay as being the most important QoS requirements in our model. According to the QoS requirements of SU_i, We define CH_j and its corresponding PU_k (if $CH_j \in \mathcal{CH}_k$) are acceptable to the QoS requirement for SU_i if and only if

$$C_{i,j} \geq R_{SU_i}, \quad (3)$$

where R_{SU_i} is the minimum rate for SU_i and

$$D_{SU_i} \geq \frac{L_{SU_i}}{P_{CH_j}C_{i,j}\left(1 - E_{SU_i}\right)}, \quad (4)$$

where P_{CH_j} is the probability that CH_j stays vacant within the delay requirement D_{SU_i}, and L_{SU_i} is total number of bits in the transmission packet. From (3) and (4), we have

$$Q_{i,j} = \max\left(R_{SU_i}, \frac{L_{SU_i}}{P_{CH_j}D_{SU_i}\left(1 - E_{SU_i}\right)}\right) \leq C_{i,j}, \quad (5)$$

where $Q_{i,j}$ is defined as SU_i's QoS requirement over PU_k's vacant channel CH_j.

B. Utility Function

In order to measure the network performance, we define the utility function $\Phi_{i,j}$ when channel CH_j is assigned to SU_i as

$$\Phi_{i,j} = C_{i,j} - \lambda Q_{i,j} \geq 0, \quad (6)$$

where λ is the coefficient to adjust the weight of $Q_{i,j}$ in the utility function. Note that $\lambda = 1$ indicates a balanced QoS requirement in (5). We can increase λ for stricter QoS requirements or decrease it to relax the QoS constraints in the CRN.

We use a $N \times K$ matrix Π as a matching matrix of the CRN. We set its element $\pi_{i,j}$ to be 1 when channel CH_j is assigned to SU_i and 0 otherwise.

We aim at maximizing the network performance without violating the SUs' QoS requirements. Thus, the network utility maximization problem can be written as

$$\max_{\Pi} \sum_{CH_j \in \mathcal{CH}} \sum_{SU_i \in \mathcal{SU}} \Phi_{i,j}$$
$$s.t.$$
$$\sum_{CH_j \in \mathcal{CH}} \pi_{i,j} \leq 1, \quad \forall i \in \{1, 2, \ldots, N\}$$
$$\sum_{SU_i \in \mathcal{SU}} \pi_{i,j} \leq 1, \quad \forall j \in \{1, 2, \ldots, K\} \quad (7)$$
$$\sum_{CH_j \in \mathcal{CH}_k} \sum_{SU_i \in \mathcal{SU}} \pi_{i,j} \leq K_k, \quad \forall k \in \{1, 2, \ldots, M\}.$$

III. PROBLEM FORMULATION

The Hungarian algorithm (HA) [8] is a centralized method that can solve the utility maximization problem in (7). However, in large-scale and densely deployed wireless networks, a distributed method is more practical. We first give the definitions of matching and stable matching, then we show how we can solve (7) using distributed matching algorithms.

A. Matching Definition

Definition 1. We define a matching $\mathcal{M}$ as a function: $\mathcal{SU} \rightarrow \mathcal{CH} \cup \{\emptyset\} \times \mathcal{PU} \cup \{\emptyset\}$, $\mathcal{PU} \rightarrow \mathcal{SU} \cup \{\emptyset\}$ and $\mathcal{CH} \rightarrow \mathcal{SU} \cup \{\emptyset\}$, such that

1) $\mathcal{M}(SU_i) = (CH_j, PU_k) \in \mathcal{CH} \cup \{\emptyset\} \times \mathcal{PU} \cup \{\emptyset\}$, $CH_j \in \mathcal{CH}_k$ and $|\mathcal{M}(SU_i)| \in \{0, 1\}$.
2) $\mathcal{M}(CH_j) = SU_i \in \mathcal{SU} \cup \{\emptyset\}$, and $|\mathcal{M}(CH_j)| \in \{0, 1\}$, where $\mathcal{M}(CH_j) = SU_i \Leftrightarrow \mathcal{M}(SU_i) = (CH_j, PU_k)$, $CH_j \in \mathcal{CH}_k$.
3) $\mathcal{M}(PU_k) \subset \mathcal{SU} \cup \{\emptyset\}$, and $|\mathcal{M}(PU_k)| \leq K_k$, where if $\mathcal{M}(CH_j) = SU_i \in \mathcal{SU}$ and $CH_j \in \mathcal{CH}_k$, then $SU_i \in \mathcal{M}(PU_k), \forall SU_i \in \mathcal{SU}$.

For $SU_i \in \mathcal{SU}$, if $\mathcal{M}(SU_i) = (CH_j, PU_k)$ we say SU_i is matched with CH_j and PU_k. Otherwise, we say SU_i is unmatched. For $CH_j \in \mathcal{CH}$, if $\mathcal{M}(CH_j) = SU_i$ and $CH_j \in \mathcal{CH}_k$, we say CH_j and PU_k is matched with SU_i. Otherwise, CH_j is unmatched. For $PU_k \in \mathcal{PU}$, we let $\mathcal{M}(PU_k)$ denote the set of SUs matched with PU_k. PU_k is undersubscribed, full or oversubscribed according to whether $|\mathcal{M}(PU_k)|$ is less than, equal to, or greater than K_k, respectively.

B. Stable Matching

We now give the definition of blocking pairs and stable matching as follows:

Definition 2. We define a blocking pair (SU_i, CH_j) that blocks a matching $\mathcal{M}$ if:

1) CH_j is acceptable to SU_i (i.e., SU_i prefers to be matched with CH_j rather than staying unmatched).
2) Either SU_i is unmatched or SU_i prefers (CH_j, PU_k) to $\mathcal{M}(SU_i)$, where PU_k is the PU providing CH_j.
3) Either CH_j is unmatched, or CH_j is matched and PU_k prefers SU_i to $\mathcal{M}(CH_j)$.

A matching is stable if it admits no blocking pair.

From the above definitions, we know that each stable matching $\mathcal{M}$ is equal to a valid matching matrix $\Pi_{\mathcal{M}}$. Thus the optimal stable matching $\mathcal{M}^*$ that yields the maximum utility is at least a suboptimal solution $\Pi_{\mathcal{M}^*}$ for the problem in (7). In the next section, we show that our algorithms can get the optimal stable matching with reasonable complexity.

IV. DISTRIBUTED SPECTRUM ACCESS SCHEME AND MATCHING ALGORITHM

A. Distributed Spectrum Access Scheme

Considering the CRN in Fig. 1, the SU node receiver first detects the vacant channels in its vicinity. If there is a channel available (e.g., CH_4-CH_6), the SU node receiver sends a
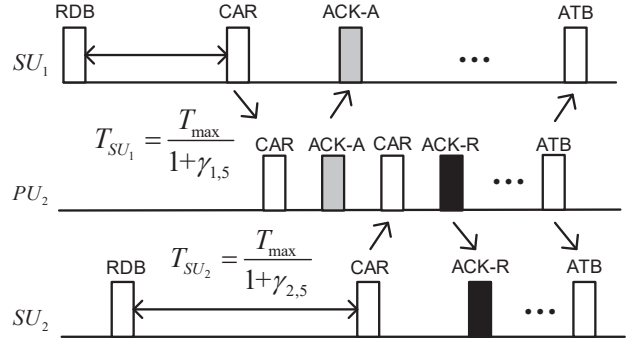

Fig. 2. Typical message exchanges of distributed spectrum access scheme

ready for data beacon (RDB) along with the channel state information (CSI) to the SU (e.g., SU_1). After receiving the RDB, the SU builds a preference list of available channels based on the CSI (i.e., the utility of each channel) and waits for $T_{SU} = \frac{T_{\max}}{1+\gamma_{\max}}$, where $T_{\max} > 0$ is the maximum waiting time and $\gamma_{\max}$ is the largest SNR among all channels. Then the SU sends a channel access request (CAR) message to the PU that has the best channel on its preference list. Since different SUs have different SNR at their receivers, the waiting period can eliminate most collisions of CAR messages emanating from different SUs to a particular PU. Other inevitable contention can be handled by methods such as binary exponential backoff as used in CSMA/CA [9]. After receiving a CAR from an SU, the PU assigns the requested channel to the SU if the SU has larger utility than the current channel holder and refuses the current holder, or refuses the SU otherwise. The PU's decision is sent to the SU by an acknowledgement (ACK) message. If a channel is assigned to an SU, an SU will wait until it receives an algorithm terminate beacon (ATB) or a refusal ACK message from its current matched PU. If an SU receives an ATB, it begins to transmit data to its receiver through the authorized channel. Otherwise, it sends a CAR message to apply for another available channel. The ATB is sent by any PU if it doesn't receive any CAR for a certain period T_{PU}.

We show typical message exchanges of the distributed spectrum access scheme in Fig. 2. Let ACK-A and ACK-R denote ACK (accept) and ACK (refuse), respectively. Based on the scenario in Fig. 1, we assume both SU_1 and SU_2 apply for CH_5 of PU_2 and $\gamma_{1,5} > \gamma_{2,5}$. After receiving RDB from its node receiver, SU_1 waits for $T_{SU_1} = \frac{T_{\max}}{1+\gamma_{1,5}}$ and sends its CAR of CH_5 to PU_2. Since it is the first request, PU_2 sends an ACK-A to SU_1. After receiving RDB from its node receiver and waiting for $T_{SU_2} = \frac{T_{\max}}{1+\gamma_{2,5}}$, SU_2 sends its CAR of CH_5 to PU_2 and receives an ACK-R from PU_2 since we assume SU_1 has larger utility than SU_2 for PU_2. After being refused, SU_2 keeps sending CARs to PUs for other available channels until it receives an ACK-A or ATB. Note that although the waiting time can eliminate most contention among different SUs to the same PU, there is still a possibility of collisions. If an SU does not receive an ACK in a given time, it assumes

its CAR collided with other CARs. The SU will wait for a time period based on binary exponential backoff as described in CSMA/CA before retransmiting its CAR.

B. Distributed Matching Algorithm

In this section, we describe our distributed matching algorithm (DMA) for the distributed spectrum access scheme. We give the DMA of the SU side and the PU side as follows:

Algorithm 1 Distributed Matching Algorithm of SU

Initialization for SU_i:
Build SU_i's preference list $list(CH)$ through message exchanges based on the utility function, $state(SU_i) = free$, $state(ATB) = false$, $\mathcal{M}(SU_i) = (\emptyset, \emptyset)$.
while $state(ATB) = false$ **do**
  if $state(SU_i) = free$ and $list(CH) \neq \emptyset$ **then**
    $CH_j = best(list(CH))$, wait for T_{SU_i},
    $send(CAR, SU_i, CH_j)$ to PU_k where $CH_j \in \mathcal{CH}_k$.
  else if $get(MSG, CH_j, PU_k) = true$ **then**
    if $MSG = ACK(accept)$ **then**
      $\mathcal{M}(SU_i) = (CH_j, PU_k)$, $state(SU_i) = occupied$.
    else if $MSG = ACK(refuse)$ **then**
      Remove CH_j from $list(CH)$, $\mathcal{M}(SU_i) = (\emptyset, \emptyset)$,
      $state(SU_i) = free$.
    else if $MSG = ATB$ **then**
      $state(ATB) = true$.
    end if
  end if
end while

Algorithm 2 Distributed Matching Algorithm of PU

Initialization for PU_k:
$state(CH_j \in \mathcal{CH}_k) = free$, $\mathcal{M}(CH_j \in \mathcal{CH}_k) = \emptyset$, $\mathcal{M}(PU_k) = \emptyset$, $timer(PU_k) = T_{PU}$.
while $timer(PU_k) > 0$ **do**
  $timer(PU_k)$ counts down.
  if $get(CAR, SU_i, CH_j) = true$ **then**
    $timer(PU_k) = T_{PU}$.
    if $state(CH_j) = free$ **then**
      $send(ACK(accept), CH_j, PU_k)$, $state(CH_j) = occupied$, $\mathcal{M}(CH_j) = SU_i$, add SU_i to $\mathcal{M}(PU_k)$.
    else if SU_i has larger utility than $\mathcal{M}(CH_j)$ **then**
      $send(ACK(accept), CH_j, PU_k)$ to SU_i.
      $send(ACK(refuse), CH_j, PU_k)$ to $\mathcal{M}(CH_j)$.
      Remove $\mathcal{M}(CH_j)$ from $\mathcal{M}(PU_k)$, $\mathcal{M}(CH_j) = SU_i$, add SU_i to $\mathcal{M}(PU_k)$.
    else
      $send(ACK(refuse), CH_j, PU_k)$ to SU_i.
    end if
  end if
end while
Broadcast $send(ATB, CH_j \in \mathcal{CH}_k, PU_k)$ to all SUs.

We then prove that DMA will always terminate and generate an optimal stable matching $\mathcal{M}^*$ that maximizes the total utility of the network. We first give the following lemmas.

Lemma 1. *DMA terminates with a matching in $O(NK)$.*

Proof. Since SU_i will remove CH_j from its preference list once it is refused, it cannot apply to CH_j more than once. The total CARs an SU can send is limited by the length of its preference list. Once there is no SU sending a CAR, the DMA terminates with an ATB broadcast by some PU. Based on the definition of matching in Definition 1, it is clear that DMA terminates with a matching in $O(NK)$ iterations. □

Lemma 2. *For any SU_i, if CH_j is deleted from SU_i's preference list during the execution of DMA, (SU_i, CH_j) cannot block the matching generated by DMA.*

Proof. Suppose CH_j is deleted from SU_i's preference list and (SU_i, CH_j) blocks the matching generated by DMA. From Algorithm 1 and 2, we know that CH_j will be deleted from SU_i's preference list if and only if PU_k that holds CH_j sends an ACK-R to SU_i. This means that $\mathcal{M}(CH_j) \neq \emptyset$ and $\mathcal{M}(CH_j)$ is better than SU_i. The above property holds until the DMA terminates. From the definition of blocking pairs in Definition 2, we know that the above property contradicts Condition 3 of blocking pairs since CH_j is matched and PU_k prefers $\mathcal{M}(CH_j)$ to SU_i. Therefore, (SU_i, CH_j) cannot be a blocking pair of the matching generated by DMA. □

We then present the following theorems based on the previous lemmas and definitions.

Theorem 1. *The DMA terminates in $O(NK)$ iterations with a stable matching.*

Proof. From Lemma 1, we know that the DMA terminates in $O(NK)$ iterations. Suppose a blocking pair (SU_i, CH_j) exists in the generated matching. We show that CH_j will be deleted by SU_i during the execution of the DMA which contradicts Lemma 2. We assume SU_i is matched with CH_t where $t \neq j$. CH_t must be the best channel in SU_i's preference list. However, since (SU_i, CH_j) is a blocking pair, SU_i must prefer CH_j to CH_t. Therefore, CH_j has to be deleted, which contradicts Lemma 2. Therefore, (SU_i, CH_j) cannot be a blocking pair. Since no blocking pair exists, the generated matching is stable. □

Theorem 2. *The DMA generates the optimal stable matching in which each SU is matched with the best channel and PU that it can have in any stable matching.*

Proof. Let $\mathcal{M}$ be the stable matching generated from the DMA where SU_i is matched with CH_j. Suppose there is another stable matching $\mathcal{M}'$ where SU_i is matched with CH_t and prefer CH_t than CH_j. Then SU_i must be refused by CH_t in $\mathcal{M}$. Suppose the refusal is caused by the matching between SU_r and CH_t. Then for CH_t, SU_r must be better than SU_i. Without loss of generality, we assume this is the first refusal in the execution of the DMA. Thus, for SU_r there is no channel better than CH_t since SU_r hasn't been refused before. In $\mathcal{M}'$, SU_r prefers CH_t to its matched channel and CH_t prefers SU_r to SU_i. Thus, (SU_r, CH_t) is a blocking pair of $\mathcal{M}'$, which contradicts that $\mathcal{M}'$ is a stable matching. □

C. Fast Distributed Matching Algorithm

In order to reduce the message exchanges in DMA, we further propose a fast distributed matching algorithm (FDMA). In FDMA, each SU sends CAR to the PU that has the highest average CSI (i.e., average channel utility). The SU does not know which specific channel it can get from the matched PU. However, if a PU accepts an SU, a vacant channel is granted to the SU. PUs accept or refuse SUs with the aim of maximizing the total utility. When a PU is oversubscribed, it refuses the SU with the lowest utility. A PU randomly assigns channels to its matched SUs. In DMA, an SU can send at most K_k CARs to PU_k, where K_k is the number of PU_k's vacant channels. However, in FDMA, it is clear that an SU can only send at most one CAR to a PU. In addition, an SU will only be refused by the matched PU if it is currently the 'worst' SU that is matched to the oversubscribed PU. Fewer CARs and ACK-R messages means FDMA needs fewer iterations to terminate compared with DMA. However, since less CSI is utilized, we expect suboptimal results from FDMA compared with DMA.

V. Simulation Results

In our simulations, we assume that the channels follow the independent identically distributed Rayleigh fading channel model and do not change over the channel allocation phase. Each channel has 50 kHz bandwidth. The transmission power is 200 mW for all SUs, the receive noise variance is 50 μW and the path loss exponent α is 2. For simplicity, we assume that the distance between all SUs and their node receivers is 20 m and the effect of collisions is neglected. We also assume there are three types of SUs with QoS constraints as follows:

TABLE I
QoS REQUIREMENTS OF SUs IN DIFFERENT TYPE

Type	R_{SU} [kbps]	D_{SU} [ms]	L_{SU} [bit]	E_{SU}
Type-I	144	10	1024	0.05
Type-II	200	1	128	0.1
Type-III	150	5	512	0.02

From (4), we know that the QoS requirement $Q_{i,j}$ is also affected by P_{CH_j}. In our simulations, we randomly generate P_{CH_j} for CH_j according to a uniform distribution on [0, 1].

We first consider a small size low competition CRN where the number of vacant channels K is larger than the number of SUs N. Assuming that each PU has at most 3 channels for SUs, we set $M = 4$, $K = 12$ and vary N from 1 to 8. Suppose there is no QoS constraints and we only care about maximizing the total utility (network throughput). We set the utility function in (6) with $\lambda = 0$ to represent the network throughput with no QoS constraints. In Fig. 3, we show that DMA has a better performance than FDMA. In a CRN where $N < K$, DMA gives a near optimal result comparable with that of the Hungarian algorithm (HA).

We now show that our algorithms can handle different levels of QoS constraints in the CRN and use DMA as an example. We assume that there are 2 Type-I, 3 Type-II and 3 Type-III SUs in the network. In (6), we use λ to adjust
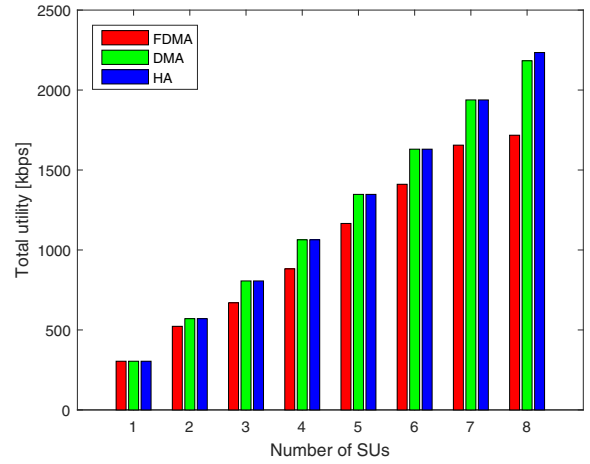

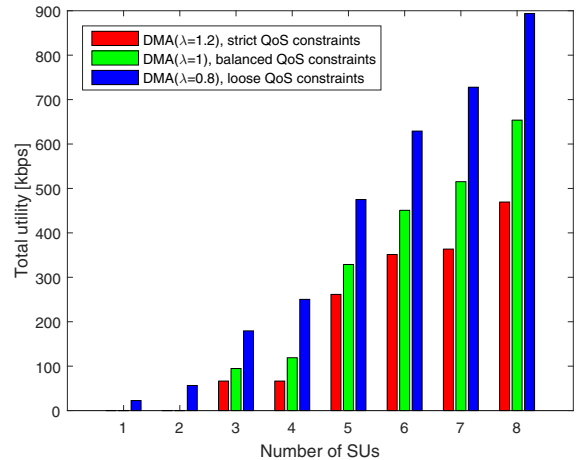Fig. 3. The total utility with different number of SUs

Fig. 4. DMA performance with different utility functions

the weight of QoS requirements in the utility function where the strictness of QoS constraints is in direct proportion to λ. In a QoS-constrained CRN, channels that fail to fulfill the QoS requirements are unacceptable. We consider several different scenarios and use λ to express the strictness of the QoS requirements. In a fast fading wireless environment, we generally require strict QoS constraints. While for applications having a high fault tolerance, loose QoS constraints may be applied. We show the total utility of DMA with $\lambda = 1.2$, $\lambda = 1$ and $\lambda = 0.8$ respectively in Fig. 4. It is clear that DMA with a larger value of λ generally achieves lower total utility since high QoS requirements make fewer channels acceptable to SUs. Note that with strict or balanced QoS constraints, the total utility is zero when $N \leq 2$ which means all the vacant channels fail to fulfill the QoS requirements of those SUs.

In Fig. 5, we evaluate our algorithms in a highly competitive CRN. We set $\lambda = 1$ in (6) and assume there are 10 PUs ($M = 10$). Each PU has 3 channels for SUs and there are 30 vacant channels ($K = 30$) in our simulation. We vary
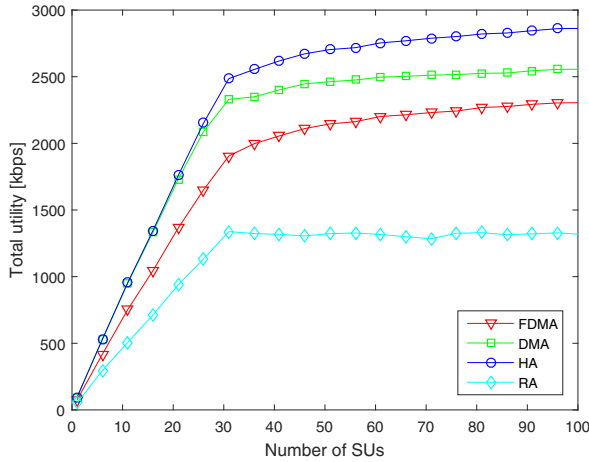
CRN, the total number of message exchanges of DMA is upper bounded by $O(N^2)$. Thus, the average number of message exchanges per SU is upper bounded by $O(N)$. For FDMA, since it utilizes the average CSI and ignores the channel differences for the same PU, it generally takes fewer iterations to terminate than does DMA. The average number of message exchanges for both algorithms is much higher once N is larger than K ($K = 30$ in the simulation) since an SU may fail several times before a channel is assigned to it owing to competitive spectrum access. From the simulation results in Fig. 5 and Fig. 6, we show that DMA generally has better performance than FDMA. However, FDMA may be a better choice for highly competitive large-scale CRN due to algorithm complexity. The performance comparison in Fig. 5 and Fig. 6 shows a tradeoff between maximizing the total utility and minimizing the total number of message exchanges.

VI. CONCLUSION

In this paper, we considered competitive distributed spectrum access in a QoS-constrained CRN. We first analyzed the QoS constraints and designed a utility function for the CRN. Then we proposed a distributed matching algorithm (DMA) to handle the dynamic spectrum access in the QoS-constrained CRN. We proved that DMA always generates a stable matching that maximizes the total utility. We further proposed a fast distributed matching algorithm (FDMA) for competitive spectrum access in a large-scale CRN. Simulation results showed that both algorithms have near optimal performance compared with the centralized algorithm. We also showed that there is a tradeoff between maximizing the total utility and minimizing the total number of message exchanges for competitive spectrum access in a large-scale CRN.

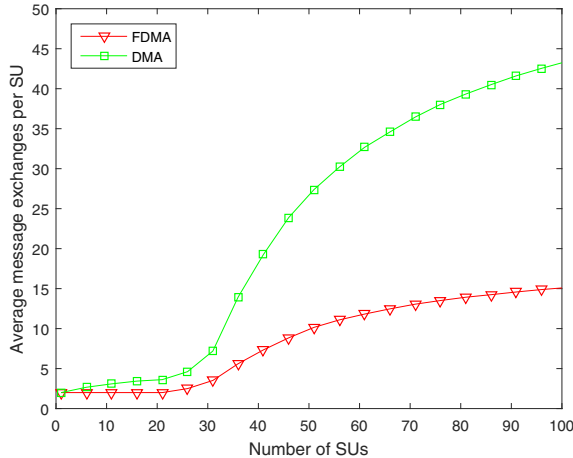Fig. 5. The total utility in large-scale CRN with different number of SUs

Fig. 6. The average message exchanges per SU with different number of SUs

the number of SUs from 1 to 100 with 30% Type-I, 30% Type-II and 40% Type-III SUs that are randomly selected. Note that some SUs may fail to gain access to any available channels, since the number of available channels is fewer than the maximum number of SUs in the network. Fig. 5 shows that DMA can achieve in excess of 95% of the total utility achieved by HA for $N \leq 25$ and about 90% of that for $N > 35$. In addition, FDMA can achieve about 80% total utility compared with HA in the highly competitive CRN and 50% more total utility compared with the random access algorithm (RA). In RA, SUs randomly select vacant channels to apply for and channels are assigned to SUs on a first-come first-served basis.

In Fig. 6, we give the average number of message exchanges per SU of our algorithms with different number of SUs. From [8], we know that HA takes $O(n^3)$ iterations to terminate, where $n = \max(N, K)$. The total number of iterations of DMA is $O(NK)$ as illustrated in Theorem 1. Since one iteration contains a constant number of message exchanges and we have $N > K$ in the highly competitive

REFERENCES

[1] M. McHenry, E. Livsics, T. Nguyen, and N. Majumdar, "Xg dynamic spectrum access field test results [topics in radio communications]," *IEEE Communications Magazine*, vol. 45, no. 6, pp. 51–57, June 2007.

[2] F. Wang, M. Krunz, and S. Cui, "Price-based spectrum management in cognitive radio networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 1, pp. 74–87, Feb 2008.

[3] L. Gao, X. Wang, Y. Xu, and Q. Zhang, "Spectrum trading in cognitive radio networks: A contract-theoretic modeling approach," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 4, pp. 843–855, 2011.

[4] Y. Gu, W. Saad, M. Bennis, M. Debbah, and Z. Han, "Matching theory for future wireless networks: fundamentals and applications," *IEEE Communications Magazine*, vol. 53, no. 5, pp. 52–59, May 2015.

[5] I. Brito and P. Meseguer, "Distributed stable matching problems with ties and incomplete lists," in *Proceedings of the 12th International Conference on Principles and Practice of Constraint Programming*, ser. CP'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 675–679.

[6] A. Leshem, E. Zehavi, and Y. Yaffe, "Multichannel opportunistic carrier sensing for stable channel access control in cognitive radio systems," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 1, pp. 82–95, January 2012.

[7] S. Vassaki, M. I. Poulakis, and A. D. Panagopoulos, "Spectrum leasing in cognitive radio networks: A matching theory approach," in *Vehicular Technology Conference (VTC Spring), IEEE 81st*, May 2015, pp. 1–5.

[8] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.

[9] G. Bianchi, L. Fratta, and M. Oliveri, "Performance evaluation and enhancement of the csma/ca mac protocol for 802.11 wireless lans," in *Personal, Indoor and Mobile Radio Communications, 1996. PIMRC'96., Seventh IEEE International Symposium on*, vol. 2, 1996, pp. 392–396.