

Chapter 2

Adaptive Channel Recommendation Mechanism

2.1 Introduction

Designing an efficient spectrum access mechanism for cognitive radio networks is challenging for several reasons: (1) *time-variation*: spectrum opportunities available for secondary users are often time-varying due to primary users' stochastic activities [1]; and (2) *limited observations*: each secondary user often has a limited view of the spectrum opportunities due to the limited spectrum sensing capability [2]. Several characteristics of the wireless channels, on the other hand, turn out to be useful for designing efficient spectrum access mechanisms: (1) *temporal correlations*: spectrum availabilities are correlated in time, and thus observations in the past can be useful in the near future [3]; and (2) *spatial correlation*: secondary users close to one another may experience similar spectrum availabilities [4]. In this chapter, we shall explore the time and space correlations and propose a recommendation-based cooperative spectrum access algorithm, which achieves good communication performances for the secondary users.

Our algorithm design is directly inspired by the recommendation system in the electronic commerce industry. For example, existing owners of various products can provide recommendations (reviews) on Amazon.com, so that other potential customers can pick the products that best suit their needs. Motivated by this, Li [5] proposed a static channel recommendation scheme that encourages secondary users to recommend the channels they have successfully accessed to nearby secondary users. Since each secondary user originally only has a limited view of spectrum availability, such information exchange enables secondary users to take advantages of the correlations in time and space, make more informed decisions, and achieve a high total transmission rate. Similarly as the Geo-location database approach required by FCC for white-space spectrum access [6], we can view the channel recommendation approach as a real-time distributed database generated by the secondary users. This is desirable, for example, when the PU activities change fast (e.g., cellular systems) and a centralized database is difficult to capture the real-time status of all primary users.

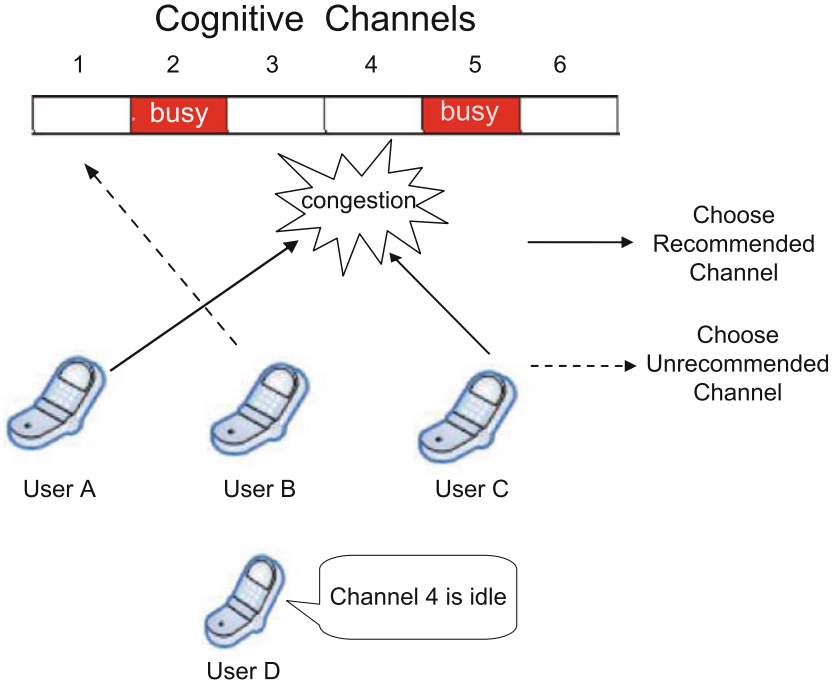


Fig. 2.1 Illustration of the channel recommendation scheme. User D recommends channel 4 to other users. As a result, both user A and user C access the same channel 4, and thus lead to congestion and a reduced rate for both users

The static recommendation scheme in [5], however, ignores two important characteristics of cognitive radios. The first one is the *time variability* we mentioned before. The second one is the *congestion effect*. As depicted in Fig. 2.1, too many users accessing the same channel leads to congestion and a reduced rate for everyone.

To address the shortcomings of the static recommendation scheme, in this chapter we propose an adaptive channel recommendation scheme, which adaptively changes the spectrum access probabilities based on users' latest channel recommendations. We formulate and analyze the system as a Markov decision process (MDP), and propose a numerical algorithm that always converges to the optimal spectrum access policy.

The main results and contributions of this chapter include:

- *Markov decision process formulation*: we formulate and analyze the optimal recommendation-based spectrum access as an average reward MDP.
- *Existence and structure of the optimal policy*: we show that there always exists a stationary optimal spectrum access policy, which requires only the channel recommendation information of the most recent time slot. We also explicitly characterize the structure of the optimal stationary policy with channel homogeneity in

two asymptotic cases (either the number of channels or the number of users goes to infinity).

- *Novel algorithm for finding the optimal policy*: we propose an algorithm based on the recently developed Model Reference Adaptive Search method [7] to find the optimal stationary spectrum access policy. The algorithm has a low complexity even when dealing with a continuous action space of the MDP. We also show that it always converges to the optimal stationary policy. We further propose an efficient heuristic scheme for the heterogeneous channel recommendation, which can significantly reduce the computational time while has small performance loss.
- *Superior performance*: we show that the proposed algorithm achieves up to 18 and 100 % performance improvement than the static channel recommendation scheme in homogeneous and heterogeneous channel environments, respectively, and is also robust to channel dynamics.

The rest of the chapter is organized as follows. We introduce the system model in Sect. 2.2. We then review the static channel recommendation scheme and discuss the motivation for designing an adaptive channel recommendation scheme in Sect. 2.3. The Markov decision process formulation and the structure results of the optimal policy are presented in Sect. 2.4, followed by the Model Reference Adaptive Search based algorithm in Sect. 2.5. We then develop a heuristic scheme for heterogeneous channel recommendation in Sect. 2.6. We illustrate the performance of the algorithms through numerical results in Sect. 2.8 and conclude in Sect. 2.9. *Due to space limitations, the details for several proofs are provided in [8].*

2.2 System Model

We consider a cognitive radio network with M parallel and stochastically heterogeneous primary channels. N homogeneous secondary users try to access these channels using a slotted transmission structure (see Fig. 2.2). The secondary users can exchange information by broadcasting messages over a common control channel.¹ We assume that the secondary users are located close-by, thus they experience similar spectrum availabilities and can hear one another's broadcasting messages. To protect the primary transmissions, secondary users need to sense the channel states before their data transmission.

The system model is described as follows:

- *Channel state*: For each primary channel m , the channel state at time slot t is

$$S_m(t) = \begin{cases} 0, & \text{if channel } m \text{ is occupied by primary transmissions,} \\ 1, & \text{if channel } m \text{ is idle.} \end{cases}$$

¹ Please refer to [9] for the details on how to set up and maintain a reliable common control channel in cognitive radio networks.

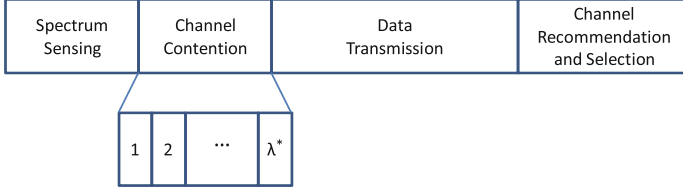
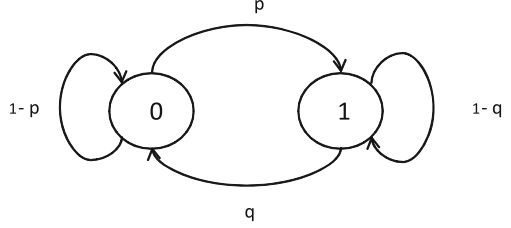


Fig. 2.2 Structure of each spectrum access time slot

Fig. 2.3 Two states Markovian channel model



- *Channel state transition:* The states of different channels change according to independent Markovian processes (see Fig. 2.3). We denote the channel state probability vector of channel m at time t as $\mathbf{p}_m(t) \triangleq (Pr\{S_m(t) = 0\}, Pr\{S_m(t) = 1\})$, which follows a two-state Markov chain as $\mathbf{p}_m(t) = \mathbf{p}_m(t-1)\Gamma_m, \forall t \geq 1$, with the transition matrix

$$\Gamma_m = \begin{bmatrix} 1 - p_m & p_m \\ q_m & 1 - q_m \end{bmatrix}.$$

Note that when $p_m = 0$ or $q_m = 0$, the channel state stays unchanged. In the rest of the chapter, we will look at the more interesting and challenging cases where $0 < p_m \leq 1$ and $0 < q_m \leq 1$. The stationary distribution of the Markov chain is given as

$$\lim_{t \rightarrow \infty} Pr\{S_m(t) = 0\} = \frac{q_m}{p_m + q_m}, \quad (2.1)$$

$$\lim_{t \rightarrow \infty} Pr\{S_m(t) = 1\} = \frac{p_m}{p_m + q_m}. \quad (2.2)$$

- *Heterogeneous channel throughput:* When a secondary user transmits successfully on an idle channel m , it achieves a data rate of B_m . Different channels can support different data rates.
- *Channel contention:* To resolve the transmission collision when multiple secondary users access the same channel, a backoff mechanism is used (see Fig. 2.2 for illustration). The contention stage of a time slot is divided into λ^* mini-slots, and each user n executes the following two steps:

1. Count down according to a randomly and uniformly chosen integral backoff time (number of mini-slots) λ_n between 1 and λ^* .
2. Once the timer expires, monitor the channel and transmit RTS/CTS messages to grab the channel if the channel is clear (i.e., no ongoing transmission). Note that if multiple users choose the same backoff mini-slot, a collision will occur with RTS/CTS transmissions and no users can grab the channel. Once successfully grabbing the channel, the user starts to transmit its data packet.

Suppose that k_m users choose channel m to access. Then the probability that user n (out of the k_m users) successfully grabs the channel m is

$$\begin{aligned}
 Pr_n &= Pr\{\lambda_n < \min_{i \neq n} \{\lambda_i\}\} \\
 &= \sum_{\lambda=1}^{\lambda^*} Pr\{\lambda_n = \lambda\} Pr\{\lambda < \min_{i \neq n} \{\lambda_i\} | \lambda_n = \lambda\} \\
 &= \sum_{\lambda=1}^{\lambda^*} \frac{1}{\lambda^*} \left(\frac{\lambda^* - \lambda}{\lambda^*} \right)^{k_m-1}.
 \end{aligned} \tag{2.3}$$

For the ease of exposition, we will focus on the asymptotic case where λ^* goes to ∞ . This is a good approximation when the number of mini-slots λ^* for backoff is much larger than the number of users N and collisions rarely occur. It simplifies the analysis as

$$\begin{aligned}
 \lim_{\lambda^* \rightarrow \infty} \sum_{\lambda=1}^{\lambda^*} \frac{1}{\lambda^*} \left(\frac{\lambda^* - \lambda}{\lambda^*} \right)^{k_m-1} &= \lim_{\frac{1}{\lambda^*} \rightarrow 0} \sum_{\lambda=0}^{\lambda^*-1} \left(\frac{\lambda}{\lambda^*} \right)^{k_m-1} \frac{1}{\lambda^*} \\
 &= \int_0^1 z^{k_m-1} dz = \frac{1}{k_m},
 \end{aligned} \tag{2.4}$$

and thus the expected throughput of user n is

$$u_n(t) = \frac{B_m S_m(t)}{k_m}. \tag{2.5}$$

In Sect. 2.7, we also generalize the results to the case that $\lambda^* < \infty$.

2.3 Introduction to Channel Recommendation

In this section, we first give a review of the static channel recommendation scheme in [5] and then discuss the motivation for adaptive channel recommendation.

2.3.1 Review of Static Channel Recommendation

The key idea of the static channel recommendation scheme is that secondary users inform each other about the available channels they have just accessed. More specifically, each secondary user executes the following four stages synchronously during each time slot (See Fig. 2.2):

- *Spectrum sensing*: sense one of the channels based on channel selection result made at the end of the previous time slot.
- *Channel contention*: if the channel sensing result is idle, compete for the channel with the backoff mechanism described in Sect. 2.2.
- *Data transmission*: transmit data packets if the user successfully grabs the channel.
- *Channel recommendation and selection*:
 - *Announce recommendation*: if the user has successfully accessed an idle channel, broadcast this channel ID to all other secondary users.
 - *Collect recommendation*: collect recommendations from other secondary users and store them in a buffer. Typically, the correlation of channel availabilities between two slots diminishes as the time difference increases. Therefore, each secondary user will only keep the recommendations received from the most recent W slots and discard the out-of-date information. The user's own successful transmission history within W recent time slots is also stored in the buffer. W is a system design parameter and will be further discussed later.
 - *Select channel*: choose a channel to sense at the next time slot by putting more weights on the recommended channels according to a *static branching probability* P_{rec} . Suppose that the user has $0 < R < M$ different channel recommendations in the buffer, then the probability of accessing a channel m is

$$P_m = \begin{cases} \frac{P_{rec}}{R}, & \text{if channel } m \text{ is recommended,} \\ \frac{1-P_{rec}}{M-R}, & \text{otherwise.} \end{cases} \quad (2.6)$$

A larger value of P_{rec} means that putting more weight on the recommended channels. When $R = 0$ (no channel is recommended) or M (all channels are recommended), the random access is used and the probability of selecting channel m is $P_m = \frac{1}{M}$.

To illustrate the channel selection process, let us take the network in Fig. 2.1 as an example. Suppose that the branching probability $P_{rec} = 0.4$. Since only $R = 1$ recommendation is available (i.e., channel 4), the probabilities of choosing the recommended channel 4 and any unrecommended channel are $\frac{0.4}{1} = 0.4$ and $\frac{1-0.4}{6-1} = 0.12$, respectively.

Numerical studies in [5] showed that the static channel recommendation scheme achieves a higher performance over the traditional random channel access scheme without information exchange. However, the fixed value of P_{rec} limits the performance of the static scheme, as explained next.

2.3.2 Motivations for Adaptive Channel Recommendation

The static channel recommendation mechanism is simple to implement due to a fixed value of P_{rec} . However, it may lead to significant congestions when the number of recommended channels is small. In the extreme case when only $R = 1$ channel is recommended, calculation (2.6) suggests that every user will access that channel with a probability P_{rec} . When the number of users N is large, the expected number of users accessing this channel $N P_{rec}$ will be high. Thus heavy congestion happens and each secondary user will get a low expected throughput.

A better way is to adaptively change the value of P_{rec} based on the number of recommended channels. This is the key idea of our proposed algorithm. To illustrate the advantage of adaptive algorithms, let us first consider a simple heuristic adaptive algorithm in a homogeneous channel environment, i.e., for each channel m , its data rate $B_m = B$ and channel state changing probabilities $p_m = p, q_m = q$. In this algorithm, we choose the branching probability such that the expected number of secondary users choosing a single recommended channel is one. To achieve this, we need to set P_{rec} as in Lemma 2.1.

Lemma 2.1 *If we choose the branching probability $P_{rec} = \frac{R}{N}$, then the expected number of secondary users choosing any one of the R recommended channels is one.*

Without going through detailed analysis, it is straightforward to show the benefit for such adaptive approach through simple numerical examples. Let us consider a network with $M = 10$ channels and $N = 5$ secondary users. For each channel m , the initial channel state probability vector is $\mathbf{p}_m(0) = (0, 1)$ and the transition matrix is

$$\Gamma_m = \begin{bmatrix} 1 - 0.01\varepsilon & 0.01\varepsilon \\ 0.01\varepsilon & 1 - 0.01\varepsilon \end{bmatrix},$$

where ε is called the dynamic factor. A larger value of ε implies that the channels are more dynamic over time. We are interested in time average system throughput $U = \frac{\sum_{t=1}^T \sum_{n=1}^N u_n(t)}{T}$, where $u_n(t)$ is the throughput of user n at time slot t . In the simulation, we set the total number of time slots $T = 2,000$.

We implement the following three channel access schemes:

- Random access scheme: each secondary user selects a channel randomly.
- Static channel recommendation scheme as in [5] with the *optimal* constant branching probability $P_{rec} = 0.7$.
- Heuristic adaptive channel recommendation scheme with the variable branching probability $P_{rec} = \frac{R}{N}$.

Figure 2.4 shows that the heuristic adaptive channel recommendation scheme outperforms the static channel recommendation scheme, which in turn outperforms the random access scheme. Moreover, the heuristic adaptive scheme is more robust to the dynamic channel environment, as it decreases slower than the static scheme when ε increases.

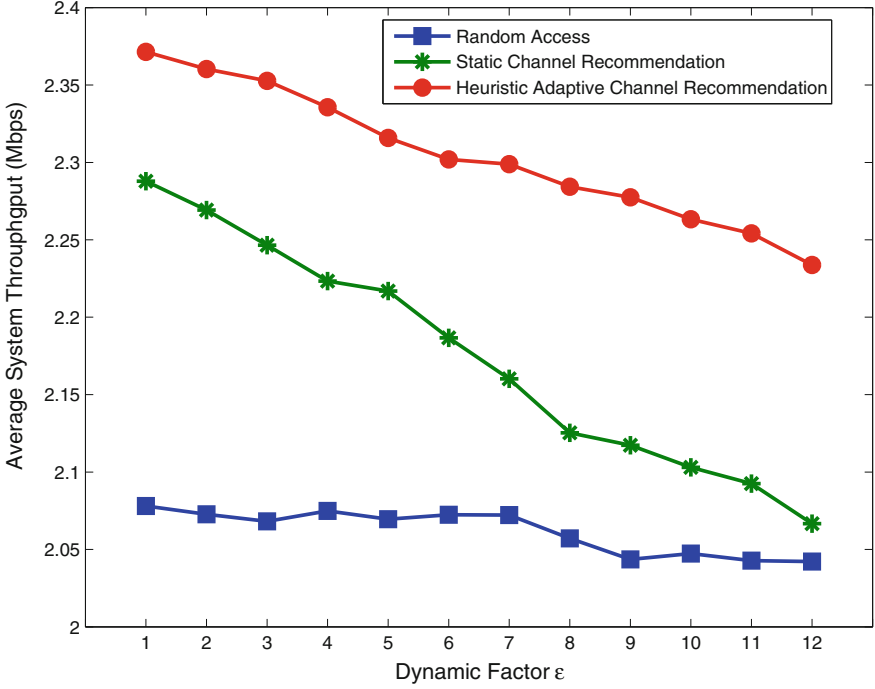


Fig. 2.4 Comparison of three channel access schemes

We can imagine that an optimal adaptive scheme (by setting the right $P_{rec}(t)$ over time) can further increase the network performance. However, computing the optimal branching probability in closed-form is very difficult. In the rest of the chapter, we will focus on characterizing the structures of the optimal spectrum access strategy and designing an efficient algorithm to achieve the optimum.

2.4 Adaptive Channel Recommendation with Channel Homogeneity

We first study the optimal channel recommendation in the homogeneous channel environment, i.e., each channel m has the same data rate $B_m = B$ and identical channel state changing probabilities $p_m = p, q_m = q$. The generalization to the heterogeneous channel setting will be discussed in Sect. 2.6. To find the optimal adaptive spectrum access strategy, we formulate the system as a Markov Decision Process (MDP). For the sake of simplicity, we assume that the recommendation buffer size $W = 1$, i.e., users only consider the recommendations received in the last time slot. Our method also applies to the case when $W > 1$ by using a high-order MDP formulation, although the analysis is more involved.

2.4.1 MDP Formulation for Adaptive Channel Recommendation

We model the system as a MDP as follows:

- **System state:** $R \in \mathcal{R} \triangleq \{0, 1, \dots, \min\{M, N\}\}$ denotes the number of recommended channels at the end of time slot t . Since all channels are statistically homogenous, then there is no need to keep track of the recommended channel IDs.
- **Action:** $P_{rec} \in \mathcal{P} \triangleq (0, 1)$ denotes the branching probability of choosing the set of recommended channels.
- **Transition probability:** The probability that action P_{rec} in system state R in time slot t will lead to system state R' in the next time slot is $P_{R,R'}^{P_{rec}} = \Pr\{R(t+1) = R' | R(t) = R, P_{rec}(t) = P_{rec}\}$. We can compute this probability as in (2.7), with detailed derivations given in [8].

$$\begin{aligned}
 P_{R,R'}^{P_{rec}} = & \sum_{m_r+m_u=R'} \sum_{R \geq \bar{m}_r \geq m_r, M-R \geq \bar{m}_u \geq m_u} \sum_{n_r+n_u=N, n_r \geq \bar{m}_r, n_u \geq \bar{m}_u} \binom{N}{n_r} \\
 & \times P_{rec}^{n_r} (1 - P_{rec})^{n_u} \binom{\bar{m}_r}{m_r} (1-q)^{m_r} q^{\bar{m}_r-m_r} \frac{R!}{(R-\bar{m}_r)!} \binom{n_r-1}{\bar{m}_r-1} R^{-n_r} \\
 & \times \binom{\bar{m}_u}{m_u} \left(\frac{p}{p+q}\right)^{m_u} \left(\frac{q}{p+q}\right)^{\bar{m}_u-m_u} \frac{(M-R)!}{(M-R-\bar{m}_u)!} \binom{n_u-1}{\bar{m}_u-1} (M-R)^{-n_u}.
 \end{aligned} \tag{2.7}$$

- **Reward:** $U(R, P_{rec})$ is the expected system throughput in next time slot when the action P_{rec} is taken in current system state R , i.e., $U(R, P_{rec}) = \sum_{R' \in \mathcal{R}} P_{R,R'}^{P_{rec}} U_{R'}$, where $U_{R'}$ is the system throughput in state R' . If R' idle channels are utilized by the secondary users in a time slot, then these R' channels will be recommended at the end of the time slot. Thus, we have $U_{R'} = R'B$. Recall that B is the data rate that a single user can obtain on an idle channel.
- **Stationary policy:** $\pi \in \Omega \triangleq \mathcal{P}^{|\mathcal{R}|}$ maps from each state R to an action P_{rec} , i.e., $\pi(R)$ is the action P_{rec} taken when the system is in state R . The mapping is stationary and does not depend on time t .

Given a stationary policy π and the initial state $R_0 \in \mathcal{R}$, we define the network's value function as the time average system throughput, i.e.,

$$\Phi_\pi(R_0) = \lim_{T \rightarrow \infty} \frac{1}{T} E_\pi \left[\sum_{t=0}^{T-1} U(R(t), \pi(R(t))) \right].$$

We want to find an optimal stationary policy π^* that maximizes the value function $\Phi_\pi(R_0)$ for any initial state R_0 , i.e., $\pi^* = \arg \max_\pi \Phi_\pi(R_0)$, $\forall R_0 \in \mathcal{R}$. Notice that this is a system wide optimization, although the optimal solution can be implemented in a distributed fashion. For example, each user can calculate the optimal spectrum access policy off-line, and determine the real-time optimal channel access probability

P_{rec} locally by observing the number of recommended channels R after entering the network.

2.4.2 Existence of Optimal Stationary Policy

MDP formulation above is an average reward based MDP. We show in Theorem 2.1 that an optimal stationary policy that is independent of initial system state always exists in our MDP formulation.

Theorem 2.1 *There exists an optimal stationary policy for the adaptive channel recommendation MDP.*

Furthermore, the optimal stationary policy π^* is independent of the initial state R_0 due to the irreducibility of the adaptive channel recommendation MDP, i.e., $\Phi_{\pi^*}(R_0) = \Phi_{\pi^*}, \forall R_0 \in \mathcal{R}$, where Φ_{π^*} is the maximum time average system throughput. In the rest of the chapter, we will just use “optimal policy” to refer “optimal stationary policy that is independent of the initial system state”.

2.4.3 Structure of Optimal Stationary Policy

Next we characterize the structure of the optimal policy without using the closed-form expressions of the policy (which is generally hard to achieve). The key idea is to treat the average reward based MDPs as the limit of a sequence of discounted reward MDPs with discounted factors going to one. Under the irreducibility condition, the average reward based MDP thus inherits the structure property from the corresponding discounted reward MDP [10]. We can write down the Bellman equations of the discounted version of our MDP problem as:

$$V_t(R) = \max_{P_{rec} \in \mathcal{P}} \sum_{R' \in \mathcal{R}} P_{R,R'}^{P_{rec}} [U_{R'} + \beta V_{t+1}(R')], \quad \forall R \in \mathcal{R}, \quad (2.8)$$

where $V_t(R)$ is the discounted maximum expected system throughput starting from time slot t when the system in state R , and $0 < \beta < 1$ is the discounted factor.

Due to the combinatorial complexity of the transition probability $P_{R,R'}^{P_{rec}}$ in (2.7), it is difficult to obtain the structure results for the general case. We further limit our attention to the following two asymptotic cases.

2.4.3.1 Case One: The Number of Channels M Goes to Infinity While the Number of Users N Stays Finite

In this case, the number of channels is much larger than the number of secondary users, and thus heavy congestion rarely happens on any channel. Thus it is safe to emphasizing on accessing the recommended channels. Before proving the main

result of Case One in Theorem 2.2, let us first characterize the property of discounted maximum expected system payoff $V_t(R)$.

Proposition 2.1 *When $M = \infty$ and $N < \infty$, the value function $V_t(R)$ for the discounted adaptive channel recommendation MDP is nondecreasing in R .*

Based on the monotone property of the value function $V_t(R)$, we prove the following main result.

Theorem 2.2 *When $M = \infty$ and $N < \infty$, for the adaptive channel recommendation MDP, the optimal stationary policy π^* is monotone, that is, $\pi^*(R)$ is nondecreasing on $R \in \mathcal{R}$.*

2.4.3.2 Case Two: The Number of Users N Goes to Infinity While the Number of Channels M Stays Finite

In this case, the number of secondary users is much larger than the number of channels, and thus congestion becomes a major concern. However, since there are infinitely many secondary users, all the idle channels at each time slot can be utilized as long as users have positive probabilities to access all channels. From the system's point of view, the cognitive radio network operates in the saturation state. Formally, we show that

Theorem 2.3 *When $N = \infty$ and $M < \infty$, for the adaptive channel channel recommendation MDP, any stationary policy π satisfying $0 < \pi(R) < 1, \forall R \in \mathcal{R}$ is optimal.*

2.5 Model Reference Adaptive Search for Optimal Spectrum Access Policy

Next we will design an algorithm that can converge to the optimal policy under general system parameters (not limiting to the two asymptotic cases). Since the action space of the adaptive channel recommendation MDP is continuous (i.e., choosing a probability P_{rec} in $(0, 1)$), the traditional method of discretizing the action space followed by the policy, value iteration, or Q-learning cannot guarantee to converge to the optimal policy. To overcome this difficulty, we propose a new algorithm developed from the Model Reference Adaptive Search method, which was recently developed in the Operations Research community [7]. We will show that the proposed algorithm is easy to implement and is provably convergent to the optimal policy.

2.5.1 Model Reference Adaptive Search Method

We first introduce the basic idea of the Model Reference Adaptive Search (MRAS) method. Later on, we will show how the method can be used to obtain optimal spectrum access policy for our problem.

The MRAS method is a new randomized method for global optimization [7]. The key idea is to randomize the original optimization problem over the feasible region according to a specified probabilistic model. The method then generates candidate solutions and updates the probabilistic model on the basis of elite solutions and a reference model, so that to guide the future search toward better solutions.

Formally, let $J(x)$ be the objective function to maximize. The MRAS method is an iterative algorithm, and it includes three phases in each iteration k :

- *Random solution generation*: generate a set of random solutions $\{x\}$ in the feasible set χ according to a parameterized probabilistic model $f(x, v_k)$, which is a probability density function (pdf) with parameter v_k . The number of solutions to generate is a fixed system parameter.
- *Reference distribution construction*: select elite solutions among the randomly generated set, such that the chosen ones satisfy $J(x) \geq \gamma$. Construct a reference probability distribution as

$$g_k(x) = \begin{cases} \frac{I_{\{J(x) \geq \gamma\}}}{E_{f(x, v_0)} \left[\frac{I_{\{J(x) \geq \gamma\}}}{f(x, v_0)} \right]} & k = 1, \\ \frac{e^{J(x)} I_{\{J(x) \geq \gamma\}} g_{k-1}(x)}{E_{g_{k-1}} [e^{J(x)} I_{\{J(x) \geq \gamma\}}]} & k \geq 2, \end{cases} \quad (2.9)$$

where $I_{\{\varpi\}}$ is an indicator function, which equals 1 if the event ϖ is true and zero otherwise. Parameter v_0 is the initial parameter for the probabilistic model (used during the first iteration, i.e., $k = 1$), and $g_{k-1}(x)$ is the reference distribution in the previous iteration (used when $k \geq 2$).

- *Probabilistic model update*: update the parameter v of the probabilistic model $f(x, v)$ by minimizing the Kullback-Leibler divergence between $g_k(x)$ and $f(x, v)$, i.e.,

$$v_{k+1} = \arg \min_v E_{g_k} \left[\ln \frac{g_k(x)}{f(x, v)} \right]. \quad (2.10)$$

By constructing the reference distribution according to (2.9), the expected performance of random elite solutions can be improved under the new reference distribution, i.e.,

$$\begin{aligned} E_{g_k} [e^{J(x)} I_{\{J(x) \geq \gamma\}}] &= \frac{\int_{x \in \chi} e^{2J(x)} I_{\{J(x) \geq \gamma\}} g_{k-1}(x) dx}{E_{g_{k-1}} [e^{J(x)} I_{\{J(x) \geq \gamma\}}]} \\ &= \frac{E_{g_{k-1}} [e^{2J(x)} I_{\{J(x) \geq \gamma\}}]}{E_{g_{k-1}} [e^{J(x)} I_{\{J(x) \geq \gamma\}}]} \geq E_{g_{k-1}} [e^{J(x)} I_{\{J(x) \geq \gamma\}}]. \end{aligned} \quad (2.11)$$

To find a better solution to the optimization problem, it is natural to update the probabilistic model (from which random solution are generated in the first stage) to as close to the new reference probability as possible, as done in the third stage.

2.5.2 Model Reference Adaptive Search for Optimal Spectrum Access Policy

In this section, we design an algorithm based on the MRAS method to find the optimal spectrum access policy. Here we treat the adaptive channel recommendation MDP as a global optimization problem over the policy space. The key challenge is the choice of proper probabilistic model $f(\cdot)$, which is crucial for the convergence of the MRAS algorithm.

2.5.2.1 Random Policy Generation

To apply the MRAS method, we first need to set up a random policy generation mechanism. Since the action space of the channel recommendation MDP is continuous, we use the Gaussian distributions. Specifically, we generate sample actions $\pi(R)$ from a Gaussian distribution for each system state $R \in \mathcal{R}$ independently, i.e. $\pi(R) \sim \mathcal{N}(\mu_R, \sigma_R^2)$.² In this case, a candidate policy π can be generated from the joint distribution of $|\mathcal{R}|$ independent Gaussian distributions, i.e.,

$$(\pi(0), \dots, \pi(\min\{M, N\})) \sim \mathcal{N}(\mu_0, \sigma_0^2) \times \dots \times \mathcal{N}(\mu_{\min\{M, N\}}, \sigma_{\min\{M, N\}}^2).$$

As shown later, Gaussian distribution has nice analytical and convergent properties for the MRAS method.

For the sake of brevity, we denote $f(\pi(R), \mu_R, \sigma_R)$ as the pdf of the Gaussian distribution $\mathcal{N}(\mu_R, \sigma_R^2)$, and $f(\pi, \boldsymbol{\mu}, \boldsymbol{\sigma})$ as random policy generation mechanism with parameters $\boldsymbol{\mu} \triangleq (\mu_0, \dots, \mu_{\min\{M, N\}})$ and $\boldsymbol{\sigma} \triangleq (\sigma_0, \dots, \sigma_{\min\{M, N\}})$, i.e.,

$$\begin{aligned} f(\pi, \boldsymbol{\mu}, \boldsymbol{\sigma}) &= \prod_{R=0}^{\min\{M, N\}} f(\pi(R), \mu_R, \sigma_R) \\ &= \prod_{R=0}^{\min\{M, N\}} \frac{1}{\sqrt{2\varphi\sigma_R^2}} e^{-\frac{(\pi(R)-\mu_R)^2}{2\sigma_R^2}}, \end{aligned}$$

where φ is the circumference-to-diameter ratio.

² Note that the Gaussian distribution has a support over $(-\infty, +\infty)$, which is larger than the feasible region of $\pi(R)$. This issue will be handled in Sect. 2.5.2.2.

2.5.2.2 System Throughput Evaluation

Given a candidate policy π randomly generated based on $f(\pi, \mu, \sigma)$, we need to evaluate the expected system throughput Φ_π . From (2.7), we obtain the transition probabilities $P_{R,R'}^{\pi(R)}$ for any system state $R, R' \in \mathcal{R}$. Since a policy π leads to a finitely irreducible Markov chain, we can obtain its stationary distribution. Let us denote the transition matrix of the Markov chain as $Q \triangleq [P_{R,R'}^{\pi(R)}]_{|\mathcal{R}| \times |\mathcal{R}|}$ and the stationary distribution as $\mathbf{p} = (Pr(0), \dots, Pr(\min\{M, N\}))$. Obviously, the stationary distribution can be obtained by solving the equation $\mathbf{p}Q = \mathbf{p}$. We then calculate the expected system throughput Φ_π by $\Phi_\pi = \sum_{R \in \mathcal{R}} Pr(R)U_R$.

Note that in the discussion above, we assume that $\pi \in \Omega$ implicitly, where Ω is the feasible policy space. Since Gaussian distribution has a support over $(-\infty, +\infty)$, we thus extend the definition of expected system throughput Φ_π over $(-\infty, +\infty)^{|\mathcal{R}|}$ as

$$\Phi_\pi = \begin{cases} \sum_{R \in \mathcal{R}} Pr(R)U_R & \pi \in \Omega, \\ -\infty & \text{Otherwise.} \end{cases}$$

In this case, whenever any generated policy π is not feasible, we have $\Phi_\pi = -\infty$. As a result, such policy π will not be selected as an elite sample (discussed next) and will not be used for probability updating. Hence the search of MRAS algorithm will not bias towards any unfeasible policy space.

2.5.2.3 Reference Distribution Construction

To construct the reference distribution, we first need to select the elite policies. Suppose L candidate policies, $\pi_1, \pi_2, \dots, \pi_L$, are generated at each iteration. We order them based on an increasing order of the expected system throughputs Φ_π , i.e., $\Phi_{\hat{\pi}_1} \leq \Phi_{\hat{\pi}_2} \leq \dots \leq \Phi_{\hat{\pi}_L}$, and set the elite threshold as $\gamma = \Phi_{\hat{\pi}_{[(1-\rho)L]}}$, where $0 < \rho < 1$ is the elite ratio. For example, when $L = 100$ and $\rho = 0.4$, then $\gamma = \Phi_{\hat{\pi}_{60}}$ and the last 40 samples in the sequence will be selected as elite samples. Note that as long as L is sufficiently large, we shall have $\gamma < \infty$ and hence only feasible policies π are selected. According to (2.9), we then construct the reference distribution as

$$g_k(\pi) = \begin{cases} \frac{I_{\{\Phi_\pi \geq \gamma\}}}{E_{f(\pi, \mu_0, \sigma_0)}[I_{\{\Phi_\pi \geq \gamma\}}]} & k = 1, \\ \frac{e^{\Phi_\pi} I_{\{\Phi_\pi \geq \gamma\}} g_{k-1}(\pi)}{E_{g_{k-1}}[e^{\Phi_\pi} I_{\{\Phi_\pi \geq \gamma\}}]} & k \geq 2. \end{cases} \quad (2.12)$$

2.5.2.4 Policy Generation Update

For the MRAS algorithm, the critical issue is the updating of random policy generation mechanism $f(\pi, \mu, \sigma)$, or solving the problem in (2.10). The optimal update rule is described as follow.

Theorem 2.4 *The optimal parameter (μ, σ) that minimizes the Kullback-Leibler divergence between the reference distribution $g_k(\pi)$ in (2.12) and the new policy generation mechanism $f(\pi, \mu, \sigma)$ is*

$$\mu_R = \frac{\int_{\pi \in \Omega} e^{(k-1)\Phi_\pi} I_{\{\Phi_\pi \geq \gamma\}} \pi(R) d\pi}{\int_{\pi \in \Omega} e^{(k-1)\Phi_\pi} I_{\{\Phi_\pi \geq \gamma\}} d\pi}, \quad \forall R \in \mathcal{R}, \quad (2.13)$$

$$\sigma_R^2 = \frac{\int_{\pi \in \Omega} e^{(k-1)\Phi_\pi} I_{\{\Phi_\pi \geq \gamma\}} [\pi(R) - \mu_R]^2 d\pi}{\int_{\pi \in \Omega} e^{(k-1)\Phi_\pi} I_{\{\Phi_\pi \geq \gamma\}} d\pi}, \quad \forall R \in \mathcal{R}. \quad (2.14)$$

2.5.2.5 MARS Algorithm for Optimal Spectrum Access Policy

Based on the MARS algorithm, we generate L candidate polices at each iteration. Then the updates in (2.13) and (2.14) are replaced by the sample average version in (2.15) and (2.16) in Algorithm 1, respectively. As a summary, we describe the MARS-based algorithm for finding the optimal spectrum access policy of adaptive channel recommendation MDP in Algorithm 1.

We then analyze the computational complexity of the MRAS algorithm. For each iteration, the sample generation in Line 4 in Algorithm 1 involves L samples with each generated from $|\mathcal{R}|$ Gaussian distributions. This step has the complexity of $\mathcal{O}(L|\mathcal{R}|)$. The elite sample selection in Line 5 involves the sorting operation, which typically has the complexity of $\mathcal{O}(L \ln L)$. The update in Line 6 involving the summation operation also has the complexity of $\mathcal{O}(L|\mathcal{R}|)$. Suppose that it takes Z iterations for the algorithm to converge. Then the total computational complexity of the MRAS algorithm is $\mathcal{O}(ZL|\mathcal{R}| + ZL \ln L)$.

Algorithm 1 MRAS-based Algorithm For Adaptive Recommendation Based Optimal Spectrum Access

- 1: **initialize** parameters for Gaussian distributions (μ_0, σ_0) , the elite ratio ρ , and the stopping criterion ξ . Set initial elite threshold $\gamma_0 = 0$ and iteration index $k = 0$.
- 2: **repeat**:
- 3: **increase** iteration index k by 1.
- 4: **generate** L candidate policies $\pi_1, \dots, \pi_L$ from the random policy generation mechanism $f(\pi, \mu_{k-1}, \sigma_{k-1})$.
- 5: **select** elite policies by setting the elite threshold $\gamma_k = \max\{\Phi_{\hat{\pi}_{(1-\rho)L}}, \gamma_{k-1}\}$.
- 6: **update** the random policy generation mechanism by (for any $\forall R \in \mathcal{R}$)

$$\mu_{R,k} = \frac{\sum_{i=1}^L e^{(k-1)\Phi_{\pi_i}} I_{\{\Phi_{\pi_i} \geq \gamma_k\}} \pi_i(R)}{\sum_{i=1}^L e^{(k-1)\Phi_{\pi_i}} I_{\{\Phi_{\pi_i} \geq \gamma_k\}}}, \quad (2.15)$$

$$\sigma_{R,k}^2 = \frac{\sum_{i=1}^L e^{(k-1)\Phi_{\pi_i}} I_{\{\Phi_{\pi_i} \geq \gamma_k\}} [\pi_i(R) - \mu_{R,k}]^2}{\sum_{i=1}^L e^{(k-1)\Phi_{\pi_i}} I_{\{\Phi_{\pi_i} \geq \gamma_k\}}}. \quad (2.16)$$

- 7: **until** $\max_{R \in \mathcal{R}} \sigma_{R,k} < \xi$.
-

2.5.3 Convergence of Model Reference Adaptive Search

In this part, we discuss the convergence property of the MRAS-based optimal spectrum access policy. For ease of exposition, we assume that the adaptive channel recommendation MDP has a unique global optimal policy. Numerical studies in [7] show that the MRAS method also converges for the multiple global optimum case. We shall show that the random policy generation mechanism $f(\pi, \mu_k, \sigma_k)$ will eventually generate the optimal policy.

Theorem 2.5 *For the MRAS algorithm, the limiting point of the policy sequence $\{\pi_k\}$ generated by the sequence of random policy generation mechanism $\{f(\pi, \mu_k, \sigma_k)\}$ converges point-wisely to the optimal spectrum access policy π^* for the adaptive channel recommendation MDP, i.e.,*

$$\lim_{k \rightarrow \infty} E_{f(\pi, \mu_k, \sigma_k)}[\pi(R)] = \pi^*(R), \quad \forall R \in \mathcal{R}, \quad (2.17)$$

$$\lim_{k \rightarrow \infty} \text{Var}_{f(\pi, \mu_k, \sigma_k)}[\pi(R)] = 0, \quad \forall R \in \mathcal{R}. \quad (2.18)$$

From Theorem 2.5, we see that parameter $(\mu_{R,k}, \sigma_{R,k})$ for updating in (2.15) and (2.16) also converges, i.e.,

$$\begin{aligned} \lim_{k \rightarrow \infty} \mu_{R,k} &= \pi^*(R), \quad \forall R \in \mathcal{R}, \\ \lim_{k \rightarrow \infty} \sigma_{R,k} &= 0, \quad \forall R \in \mathcal{R}. \end{aligned}$$

Thus, we can use $\max_{R \in \mathcal{R}} \sigma_{R,k} < \xi$ as the stopping criterion in Algorithm 1.

2.6 Adaptive Channel Recommendation with Channel Heterogeneity

We now generalize the adaptive channel recommendation to the heterogeneous channel setting. Recall that the system state R in the homogeneous channel case only keeps track of how many channels are recommended. In a heterogeneous channel environment, each channel has different data rate B_m and channel state changing probabilities p_m and q_m . Keeping track of the number of recommend channels is not enough for optimal decision. Intuitively, if a channel with higher data rate B_m is recommended, users should choose this channel with a higher weight. The new system state for the heterogeneous channel case should be defined as a vector $\mathbf{R} \triangleq (I_1, \dots, I_M)$, where $I_m = 1$ if channel m is recommended and $I_m = 0$ otherwise. The objective of the heterogeneous channel recommendation MDP is then to find the optimal channel access probabilities $\{P_m(\mathbf{R})\}_{m=1}^M$ for each system state $\mathbf{R}$ where $P_m(\mathbf{R})$ is the probability of selecting channel m .

Similarly with the homogeneous channel case, we can apply the MRAS method (by replacing system state R and decision variables P_{rec} in Algorithm 1 with $\mathbf{R}$ and $\{P_m(\mathbf{R})\}_{m=1}^M$, respectively) to obtain the optimal solutions with the new formulation.

However, the number of decision variables $\{P_m(\mathbf{R})\}_{m=1}^M$ in the heterogeneous channel model equals to $M2^M$, which causes exponential blow up in the computational complexity (i.e., $\mathcal{O}(ZLM2^M + ZL \ln L)$) with the similar analysis as in Sect. 2.5.2.5). We next focus on developing a low complexity efficient heuristic algorithm to solve the MDP.

Recall that in the heuristic algorithm in Lemma 2.1 for the homogeneous channel recommendation, the weight of selecting each recommended channel is $\frac{1}{N}$ and total weights of choosing recommended channels are $R \frac{1}{N}$. Similarly, we can design a low complexity heuristic algorithm for the heterogeneous channel recommendation. More specifically, we set the weight of selecting channel m is P_1^m (P_0^m , respectively) when the channel is recommended (the channel is not recommended, respectively). Given the system is in state $\mathbf{R}$, the probability of choosing channel m is proportional to its weight of its state I_m , i.e.,

$$P_m(\mathbf{R}) = \frac{P_{I_m}^m}{\sum_{m'=1}^M P_{I_{m'}}^m}. \quad (2.22)$$

In this case, the total number of decision variables $P_{I_m}^m$ is reduced to $2M$, which grows linearly in the number of channels M . Let $\boldsymbol{\pi} = \{(P_1^m, P_0^m)\}_{m=1}^M \in (0, 1)^{2M}$ denote the set of corresponding decision variables. Our objective is to find the optimal $\boldsymbol{\pi}$ that maximizes the time average throughput $\Phi_{\boldsymbol{\pi}}$. We can again apply the MRAS method to find the optimal solution, which is given in Algorithm 2. The procedures of derivation is very similar with the MRAS method for the homogeneous channel recommendation; we omit the details due to space limit. With the similar analysis as in Sect. 2.5.2.5, we see that the heuristic algorithm has the computational complexity of $\mathcal{O}(ZLM + ZL \ln L)$.

Algorithm 2 MRAS-based Algorithm For Optimizing Heuristic Heterogeneous Channel Recommendation

- 1: **initialize** parameters for the elite ratio ρ , Gaussian distributions $\boldsymbol{\mu}(0) = \{(\mu_1^m(0), \mu_0^m(0))\}_{m=1}^M$, $\boldsymbol{\sigma}(0) = \{(\sigma_1^m(0), \sigma_0^m(0))\}_{m=1}^M$, and the stopping criterion ξ . Set initial elite threshold $\gamma_0 = 0$ and iteration index $k = 0$.
- 2: **repeat**:
- 3: **increase** iteration index k by 1.
- 4: **generate** L candidate policies $\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_L$ from the random policy generation mechanism $f(\boldsymbol{\pi}, \boldsymbol{\mu}(k-1), \boldsymbol{\sigma}(k-1))$.
- 5: **select** elite policies by setting the elite threshold $\gamma_k = \max\{\Phi_{\hat{\boldsymbol{\pi}}_{\lceil(1-\rho)L\rceil}}, \gamma_{k-1}\}$.
- 6: **update** the random policy generation mechanism by (for any $I_m \in \{0, 1\}, m \in \mathcal{M}$)

$$\mu_{I_m}^m(k) = \frac{\sum_{i=1}^L e^{(k-1)\Phi_{\boldsymbol{\pi}}} I_{\{\Phi_{\boldsymbol{\pi}_i} \geq \gamma_k\}} P_{I_m}^m}{\sum_{i=1}^L e^{(k-1)\Phi_{\boldsymbol{\pi}}} I_{\{\Phi_{\boldsymbol{\pi}_i} \geq \gamma_k\}}}, \quad (2.20)$$

$$\sigma_{I_m}^m(k) = \left(\frac{\sum_{i=1}^L e^{(k-1)\Phi_{\boldsymbol{\pi}}} I_{\{\Phi_{\boldsymbol{\pi}_i} \geq \gamma_k\}} (P_{I_m}^m - \mu_{I_m}^m(k))^2}{\sum_{i=1}^L e^{(k-1)\Phi_{\boldsymbol{\pi}}} I_{\{\Phi_{\boldsymbol{\pi}_i} \geq \gamma_k\}}} \right)^{\frac{1}{2}}. \quad (2.21)$$

- 7: **until** $\max_{I_m \in \{0, 1\}, m \in \mathcal{M}} \sigma_{I_m}^m(k) < \xi$.
-

Note that the optimal policy π^* for the heuristic heterogeneous channel recommendation is also a feasible policy for the heterogeneous channel recommendation MDP. The performance of the optimal policy for the heterogeneous channel recommendation MDP thus dominates the heuristic heterogeneous channel recommendation. However, numerical results show that the heuristic heterogeneous channel recommendation has a small performance loss comparing to the optimal policy while gaining a significant computation complexity reduction.

2.7 Adaptive Channel Recommendation in General Channel Environment

For the ease of exposition, we consider the Markovian channel model in the analysis above. Such a channel model can be a good approximation of reality if the primary traffic is highly bursty [11]. We now extend the MRAS-based channel recommendation algorithm to a general channel environment including the non-Markovian setting, where it is difficult to obtain the statistical properties apriori.

The key idea is to cast the system throughput optimization problem in the general channel environment as a stochastic optimization problem. Let $\mathbf{S} = (S_1, \dots, S_M)$ be the states of all channels, which is a random vector generated from a general probability distribution ψ . Then the stochastic system throughput optimization problem is given as

$$\max_{\pi} E_{\mathbf{S} \sim \psi} [\Phi_{\pi}(\mathbf{S})], \quad (2.23)$$

where $\Phi_{\pi}(\mathbf{S})$ denotes the system throughput under the channel states $\mathbf{S}$, and $E_{\mathbf{S} \sim \psi}[\cdot]$ denotes the expected system throughput under the channel state distribution ψ . Recent result in [12] shows that the MRAS algorithm can be used to solve such stochastic optimization problem by drawing a large samples of channel-states $\{\mathbf{S}(1), \dots, \mathbf{S}(L)\}$ from the probability distribution ψ and evaluating the expected performance by the sample average (i.e., $E_{\mathbf{S} \sim \psi} [\Phi_{\pi}(\mathbf{S})] = \frac{1}{L} \sum_{l=1}^L \Phi_{\pi}(\mathbf{S}(l))$). When the size of channel-states samples is large enough, the MRAS algorithm can converge to the optimal solution π^* approximately [12]. Based on the idea above, secondary users can first probe the channel environment by sensing and recording the channel states $\{\mathbf{S}(t)\}_{t=1}^T$ over a long time period consisting of T time slots. Note that the channel probing can be achieved in a collaborative way that each user selects one channel to sense, and shares the sensing results with other users at end of the probe period. Then each user can apply the MRAS algorithm to compute the near-optimal channel recommendation policy π^* by constituting Φ_{π} as $\frac{1}{T} \sum_{t=1}^T \Phi_{\pi}(\mathbf{S}(t))$ in Algorithm 2.

Note that the optimization problem in (2.23) can also be generalized to take other dynamic factors into account. For example, let $\mathbf{q} = (q_1, \dots, q_M)$ denote the loss

rates of all the channels, which follow a probability distribution ϕ . Then the stochastic system throughput optimization problem can be written as

$$\max_{\pi} E_{\mathbf{S} \sim \psi, \boldsymbol{\varrho} \sim \phi} [\Phi_{\pi}(\mathbf{S}, \boldsymbol{\varrho})], \quad (2.24)$$

where $\Phi_{\pi}(\mathbf{S}, \boldsymbol{\varrho})$ denotes the expected system throughput under the channel states $\mathbf{S}$ and channel loss rates $\boldsymbol{\varrho}$. We can solve the problem (2.24) with a similar procedure as described above.

As another example, we can apply the optimization formulation in (2.23) to address the issue of heterogeneous user capacities. Let $\mathbf{a}(t) = (a_1(t), \dots, a_N(t))$ be the channel selections of all users at time slot t , and let B_n^m denote the mean data rate that user n achieves on channel m . Then the stochastic system throughput optimization problem in (2.23) can be written as

$$\begin{aligned} \max_{\pi} E_{\mathbf{S} \sim \psi} [\Phi_{\pi}(\mathbf{S})] &= \max_{\pi} \frac{1}{T} \sum_{t=1}^T \Phi_{\pi}(\mathbf{S}(t)) \\ &= \max_{\pi} E_{\{\mathbf{a}(t)\}_{t=1}^T \sim \pi} \left[\frac{1}{T} \sum_{t=1}^T U(\mathbf{S}(t), \mathbf{a}(t)) \right]. \end{aligned}$$

where $U(\mathbf{S}(t), \mathbf{a}(t))$ denotes the system throughput under channel states $\mathbf{S}$ and channel selections $\mathbf{a}$, which can be computed as $U(\mathbf{S}(t), \mathbf{a}(t)) = \sum_{n=1}^N S_{a_n(t)}(t) B_n^{a_n(t)} \times g_n^{a_n(t)}(\mathbf{a}(t))$. Here $g_n^{a_n}(\mathbf{a})$ denotes the probability that user n successfully grabs the channel a_n , which can be derived from the adopted channel contention mechanism. For the random backoff mechanism in this chapter, we have $g_n^{a_n(t)}(\mathbf{a}(t)) = \frac{1}{\sum_{i=1}^N I_{\{a_i(t)=a_n(t)\}}}$. Similarly, by the sample average approach (i.e., drawing L samples of actions over T time slots $\{\mathbf{a}(t)\}_{t=1}^T$ from the policy π), we can obtain the expected system throughput as

$$E_{\mathbf{S} \sim \psi} [\Phi_{\pi}(\mathbf{S})] = \frac{\sum_{t=1}^T \sum_{l=1}^L U(\mathbf{S}(t), \mathbf{a}_l(t))}{TL},$$

and then apply the MRAS algorithm to find the solution.

2.8 Simulation Results

In this section, we investigate the proposed adaptive channel recommendation scheme by simulations. The results show that the adaptive channel recommendation scheme not only achieves a higher performance over the static scheme and random access scheme, but also is more robust to the dynamic change of the channel environments.

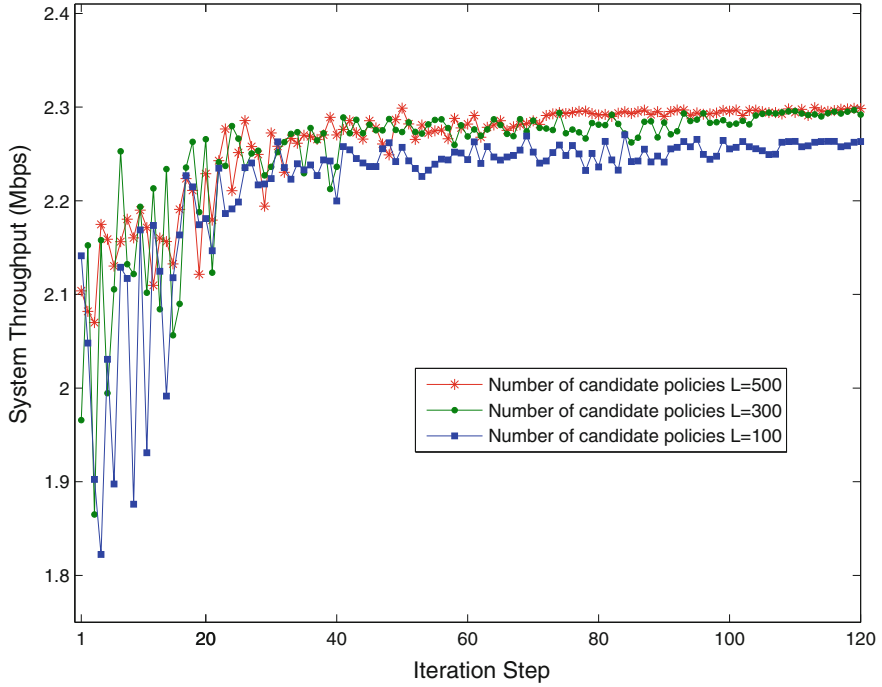


Fig. 2.5 The convergence of MRAS-based algorithm with different number of candidate policies per iteration

2.8.1 Simulation Setup

We initialize the parameters of MRAS algorithm as follows. We set $\mu_R = 0.5$ and $\sigma_R = 0.5$ for the Gaussian distribution, which has 68.2 % support over the feasible region $(0, 1)$. We found that the performance of the MRAS algorithm is insensitive to the elite ratio ρ when $\rho \leq 0.3$. We thus choose $\rho = 0.1$.

When using the MRAS-based algorithm, we need to determine how many (feasible) candidate policies to generate in each iteration. Figure 2.5 shows the convergence of MRAS algorithm with 100, 300, and 500 candidate policies per iteration, respectively. We have two observations. First, the number of iterations to achieve convergence reduces as the number of candidate policies increases. Second, the convergence speed is insignificant when the number changes from 300 to 500. We thus choose $L = 500$ for the experiments in the sequel.

Homogeneous Channel Recommendation

We first consider a cognitive radio network consisting of $M = 10$ stochastically homogeneous primary channels, and $N = 5$ secondary users. The data rate of each

channel is normalized to be 1 Mbps. In order to take the impact of primary user's long run behavior into account, we consider the following two types of homogeneous channel environments (i.e., channel state transition matrices):

$$\text{Type 1: } \Gamma^1 = \begin{bmatrix} 1 - 0.005\varepsilon & 0.005\varepsilon \\ 0.025\varepsilon & 1 - 0.025\varepsilon \end{bmatrix}, \quad (2.25)$$

$$\text{Type 2: } \Gamma^2 = \begin{bmatrix} 1 - 0.01\varepsilon & 0.01\varepsilon \\ 0.01\varepsilon & 1 - 0.01\varepsilon \end{bmatrix}, \quad (2.26)$$

where ε is the dynamic factor. Recall that a larger ε means that the channels are more dynamic over time. Using (2.2), we know that channel environments Γ^1 and Γ^2 have the stationary channel idle probabilities of $1/6$ and $1/2$, respectively. In other words, the primary activity level is much higher with the Type 1 channel environment than with the Type 2 channel environment. We implement the adaptive channel recommendation scheme, and benchmark it with the static channel recommendation scheme in [5] and the random access scheme. We choose the dynamic factor ε within a wide range to investigate the robustness of the schemes to the channel dynamics. The results are shown in Figs. 2.6 and 2.7. From these figures, we see that the adaptive channel recommendation scheme offers 5–18% performance gain over the static scheme. Moreover, the adaptive channel recommendation is much more robust to

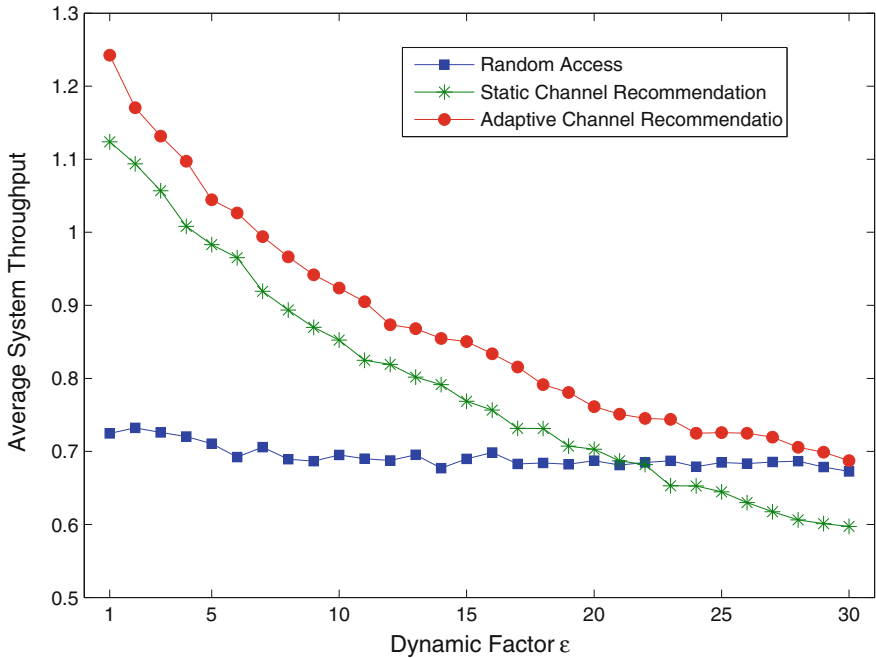


Fig. 2.6 System throughput with $M = 10$ channels and $N = 5$ users under the Type 1 channel state transition matrix

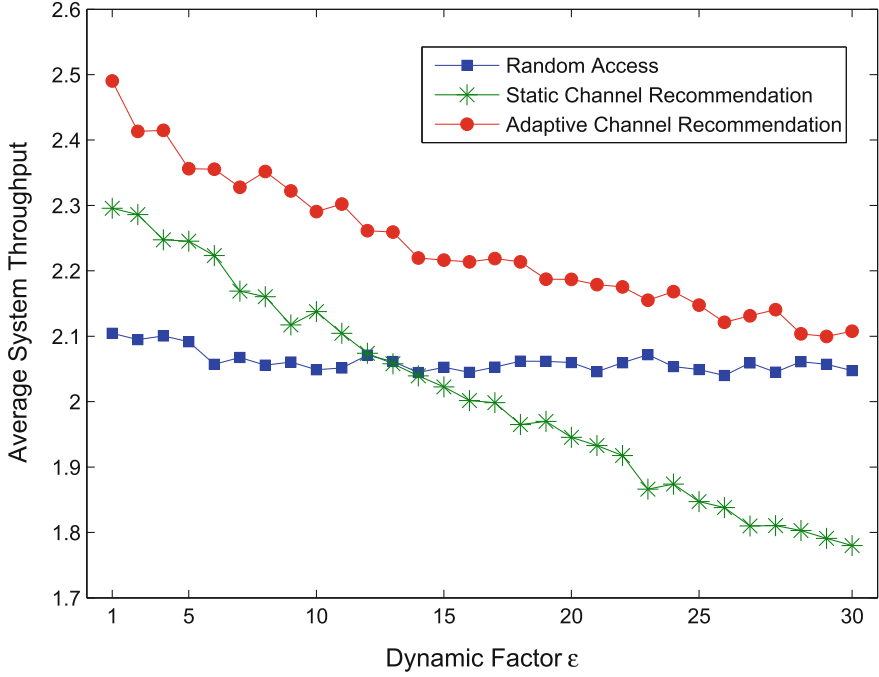


Fig. 2.7 System throughput with $M = 10$ channels and $N = 5$ users under the Type 2 channel state transition matrix

the dynamic channel environment changing. The reason is that the optimal adaptive policy takes the channel dynamics into account while the static one does not.

2.8.2 Heuristic Heterogenous Channel Recommendation

We now evaluate the proposed heuristic heterogeneous channel recommendation mechanism in Sect. 2.6. We implement the heuristic heterogeneous channel recommendation mechanism in heterogeneous channel environments. The data rates of $M = 10$ channels are $\{B_1 = 0.2, B_2 = 0.6, B_3 = 0.8, B_4 = 1, B_5 = 2, B_6 = 4, B_7 = 6, B_8 = 8, B_9 = 10, B_{10} = 20\}$ Mbps. The stochastic channel state changing environment is given as:

$$\{\Gamma_1 = \Gamma^1, \Gamma_2 = \Gamma^1, \Gamma_3 = \Gamma^1, \Gamma_4 = \Gamma^1, \Gamma_5 = \Gamma^1, \Gamma_6 = \Gamma^2, \Gamma_7 = \Gamma^2, \Gamma_8 = \Gamma^2, \Gamma_9 = \Gamma^2, \Gamma_{10} = \Gamma^2\}. \quad (2.27)$$

Here subscript denotes channel index, and superscript denote channel type index. We also implement static channel recommendation, the optimal homogeneous channel

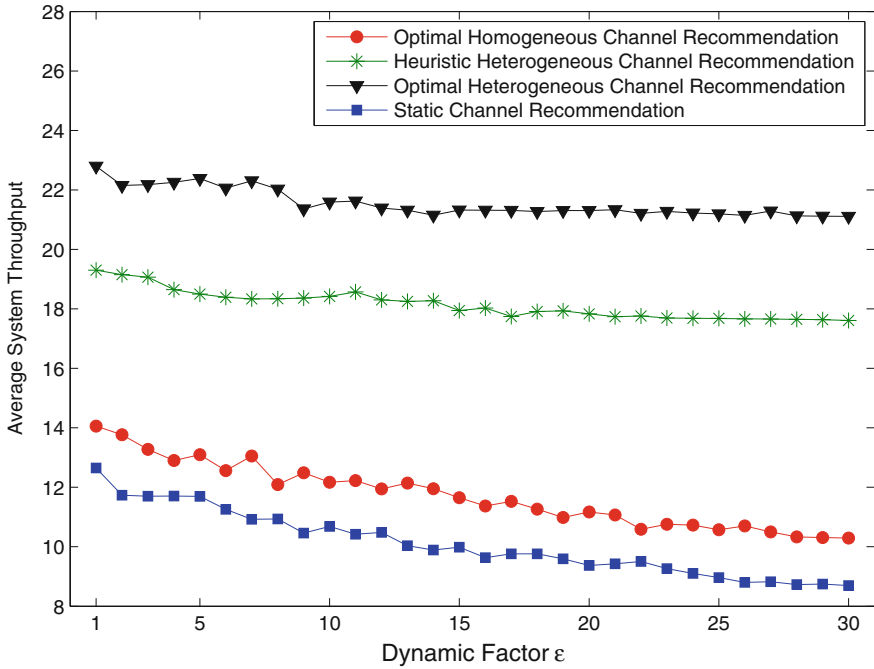


Fig. 2.8 Comparison of heuristic heterogeneous channel recommendation, optimal homogeneous channel recommendation and optimal heterogeneous channel recommendation

recommendation (Algorithm 1) and optimal heterogeneous channel recommendation (similar with Algorithm 1 by replacing system state R and decision variables P_{rec} with $\mathbf{R}$ and $\{P_m(\mathbf{R})\}_{m=1}^M$, respectively) as benchmarks. The results are depicted in Fig. 2.8. From the figure, we see that the heuristic heterogeneous channel recommendation achieves up-to 70 and 100 % performance improvement over the optimal homogeneous channel recommendation and static channel recommendation, respectively. The performance loss is at most 20 % comparing with the the optimal heterogeneous channel recommendation. Note that the number of decision variables in the optimal heterogeneous channel recommendation is $M2^M = 10,240$, while the number of decision variables in the heuristic heterogeneous channel recommendation is only $2M = 20$. The convergence of the heuristic heterogeneous channel recommendation hence is much faster than the optimal heterogeneous channel recommendation.

2.8.3 Simulation with Real Channel Data

We now evaluate the adaptive channel recommendation scheme using real channel data. The data we used (from Xu et al. [13]) is the spectral measurements taken in 850–870MHz public safety band in Maryland. The measured band is divided

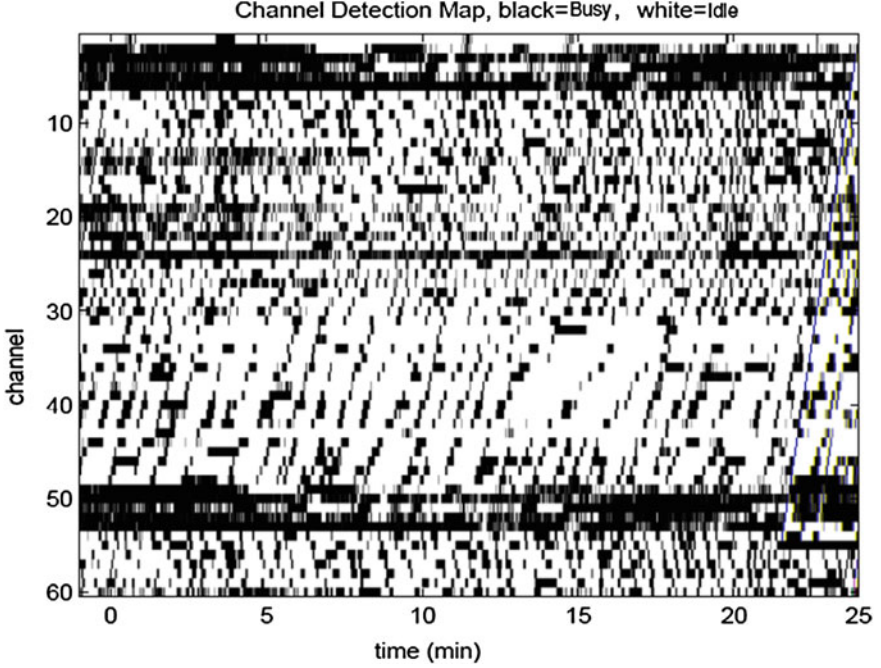


Fig. 2.9 Channel activity map from trace data of 850–870 MHz band in Maryland [13]

into 60 channels, and each channel has a bandwidth of 25 KHz. The measurements were taken over a duration of 25 min, with each time slot being 0.01 s. PU’s activity is determined by the energy detection with a threshold of 10 dB above the noise floor [14]. Figure 2.9 visualizes the real trace data. We observe that these channels exhibit a large number of busy/idle cycles (i.e., temporal correlations) and statistically heterogeneous channel availabilities.

We implement the heuristic heterogeneous channel recommendation scheme in a network consisting of 6 channels from the real data. We set the mean data rates of all channels as $\{B_1 = 5, B_2 = 8, B_3 = 12, B_4 = 15, B_5 = 18, B_6 = 20\}$ Mbps. For the channel contention, we set the number of backoff mini-slots $\lambda^* = 20$. Besides the system-wide throughput, we also consider the average access delay, i.e., the average number of time slots that a secondary user needs to wait until its data packet can successfully go through for transmission without blocking. A data packet can be blocked due to the factors such as the channel availability and channel contentions. As a benchmark, we also implement a belief-based channel access scheme proposed in previous work [15, 16] as follows:

- Each user n maintains the following two vectors: $\mathbf{X}_n = (X_1^n, \dots, X_M^n)$ and $\mathbf{Y}_n = (Y_1^n, \dots, Y_M^n)$, where X_m^n and Y_m^n record the number of time slots in which the

channel m has been sensed to be free, and the number of time slots in which the channel m has been sensed. Set $X_m^n = Y_m^n = 1$ initially.

- At the beginning of each time slot, each user n computes its belief as $\omega_m^n = \frac{X_m^n}{Y_m^n}$ and chooses each channel m with probability $\frac{\omega_m^n}{\sum_{m'=1}^M \omega_{m'}^n}$.
- At the end of each time slot, each user n broadcasts the sensing result to other users, and then updates the parameters X_n and Y_n based on the overall sensing results of all users.

The key idea of the belief-based channel access is to select channels based on the belief ω_m^n generated from the history of users' observations X_n and Y_n . We implement the adaptive channel recommendation and belief-based channel access schemes with the number of users N ranging from 2 to 8. The results are shown in Figs. 2.10 and 2.11. Compared with the belief-based channel access scheme, we see that the channel recommendation scheme can achieve up-to 30 % system throughput improvement, and reduce up-to 15 % access delay.

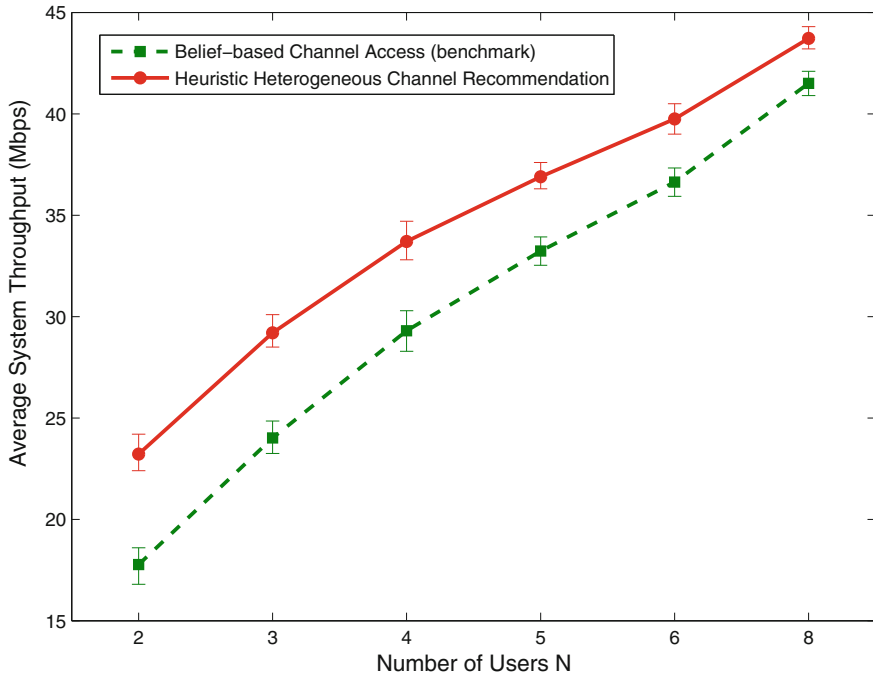


Fig. 2.10 System average throughput of channel recommendation and belief-based channel access (benchmark) schemes with heterogeneous channel data rates

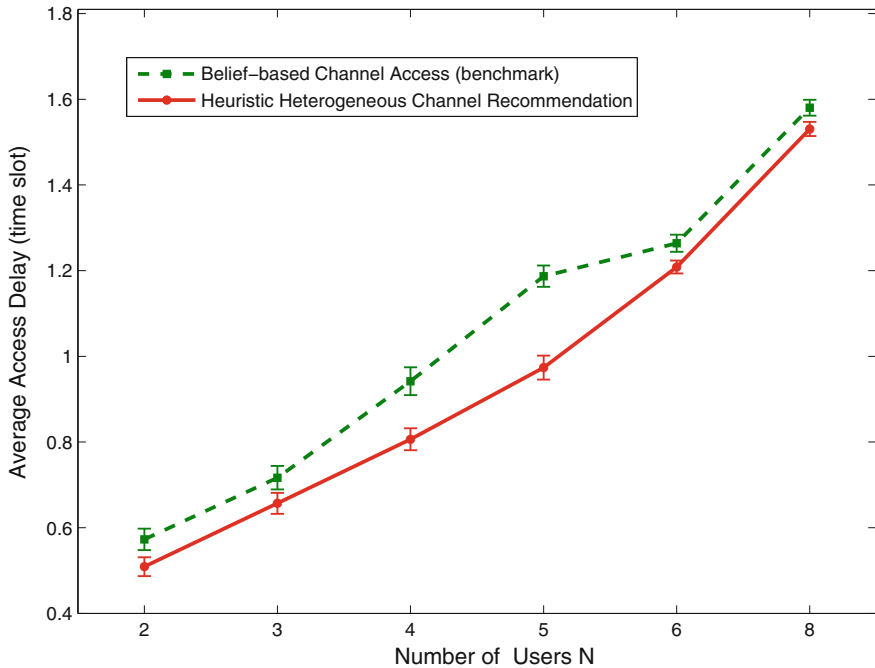


Fig. 2.11 Average access delay of channel recommendation and belief-based channel access (benchmark) schemes with heterogeneous channel data rates

2.9 Summary

In this chapter, we propose an adaptive channel recommendation scheme for efficient spectrum sharing. We formulate the problem as an average reward based Markov decision process. We first prove the existence of the optimal stationary spectrum access policy, and then characterize the structure of the optimal policy in two asymptotic cases. Furthermore, we propose a novel MRAS-based algorithm that is provably convergent to the optimal policy. Numerical results show that our proposed algorithm outperforms the static approach in the literature by up to 100 % in terms of system throughput. Our algorithm is also more robust to the channel dynamics compared to the static counterpart.

References

1. J. Mitola, Cognitive radio: an integrated agent architecture for software defined radio, Ph.D. dissertation, Royal Institute of Technology (KTH) Stockholm, Sweden, 2000
2. Q. Zhao, L. Tong, A. Swami, Y. Chen, Decentralized cognitive MAC for opportunistic spectrum access in ad hoc networks: a pomdp framework. *IEEE J. Sel. Areas Commun.* **25**, 589–600 (2007)

3. M. Wellens, J. Riihijarvi, P. Mahonen, Empirical time and frequency domain models of spectrum use. *Elsevier Phys. Commun.* **2**, 10–32 (2009)
4. M. Wellens, J. Riihijarvi, M. Gordziel, P. Mahonen, Spatial statistics of spectrum usage: from measurements to spectrum models, in *IEEE International Conference on Communications* (2009)
5. H. Li, Customer reviews in spectrum: recommendation system in cognitive radio networks, in *IEEE Symposia on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)* (2010)
6. R. Murty, R. Chandra, T. Moscibroda, P. Bahl, Senseless: a database-driven white spaces network, in *IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)* (2011)
7. J. Hu, M. Fu, S. Marcus, A model reference adaptive search algorithm for global optimization. *Oper. Res.* **55**, 549–568 (2007)
8. X. Chen, J. Huang, H. Li, Adaptive channel recommendation for opportunistic spectrum access. *IEEE Trans. Mob. Comput.* **12**(9), 1788–1800 (2013). <http://arxiv.org/pdf/1102.4728.pdf>
9. C. Cormio, K.R. Chowdhury, Common control channel design for cognitive radio wireless ad hoc networks using adaptive frequency hopping. *Elsevier J. Ad Hoc Netw.* **8**, 430–438 (2010)
10. S.M. Ross, *Introduction to Stochastic Dynamic Programming* (Academic Press, New York, 1993)
11. E. Gilbert et al., Capacity of a burst-noise channel. *Bell Syst. Tech. J.* **39**(9), 1253–1265 (1960)
12. J. Hu, M. Fu, S. Marcus, A model reference adaptive search method for stochastic global optimization. *Commun. Inf. Syst.* **8**(3), 245–276 (2008)
13. D. Xu, E. Jung, X. Liu, Optimal bandwidth selection in multi-channel cognitive radio networks: how much is too much? in *3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)* (2008)
14. S. Jones, E. Jung, X. Liu, N. Merheb, I. Wang, Characterization of spectrum activities in the US public safety band for opportunistic spectrum access, in *IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)* (2007)
15. H. Liu, B. Krishnamachari, Q. Zhao, Cooperation and learning in multiuser opportunistic spectrum access, in *IEEE International Conference on Communications (ICC)* (2008)
16. L. Lai, H. El Gamal, H. Jiang, H. Poor, Cognitive medium access: exploration, exploitation, and competition. *IEEE Trans. Mob. Comput.* **10**(2), 239–253 (2011)



<http://www.springer.com/978-3-319-15214-1>

Social Cognitive Radio Networks

Chen, X.; Huang, J.

2015, XI, 83 p. 32 illus., Softcover

ISBN: 978-3-319-15214-1