Maximizing System Throughput by Cooperative Sensing in Cognitive Radio Networks

Shuang Li, *Student Member, IEEE,* Zizhan Zheng, *Member, IEEE,* Eylem Ekici, *Senior Member, IEEE,* and Ness Shroff, *Fellow, IEEE*

***Abstract*—Cognitive Radio Networks allow unlicensed users to opportunistically access the licensed spectrum without causing disruptive interference to the primary users (PUs). One of the main challenges in CRNs is the ability to detect PU transmissions. Recent works have suggested the use of secondary user (SU) cooperation over individual sensing to improve sensing accuracy. In this paper, we consider a CRN consisting of multiple PUs and SUs to study the problem of maximizing the total expected system throughput. First, we study the sensing decision problem for maximizing the system throughput subject to a constraint on the PU throughput and we design a Bayesian decision rule based algorithm. The problem is shown to be strongly NP-hard and solved via a greedy algorithm with time complexity $O(\frac{N^5}{\log^2 \frac{1}{1-\epsilon}})$ where N is the total number of SUs. The algorithm achieves a throughput strictly greater than $\frac{1}{2}(1-\epsilon)$ of the optimal solution and results in a small constraint violation that goes to zero with ϵ. We then investigate the more general problem with constraints on both PU throughput and the sensing time overhead, which limits the number of SUs that can participate in cooperative sensing. We illustrate the efficacy of the performance of our algorithms and provide sensitivity analysis via a numerical investigation.**

I. Introduction

Cognitive radio networks (CRNs) have been proposed to address the spectrum scarcity problem by allowing unlicensed users (secondary users, SUs) to access licensed spectrum on the condition of not disrupting the communication of licensed users (primary users, PUs). To this end, SUs sense licensed channels to detect the primary user (PU) activities and find underutilized "white spaces". FCC has opened the TV bands for unlicensed access [6], and IEEE has formed a working group (IEEE 802.22 [8]) to regulate the unlicensed access without interference. Many other organizations are also making efforts on the spectrum access policy in the CRN environment, e.g., DARPA's 'Next Generation' (XG) program [21] mandates cognitive radios to sense signals and prevent interference to existing military and civilian radio systems. A practical example is the opportunistic access of the police radio spectrum. Since the PU activity is not known by the SUs in real time, SUs have to sense the spectrum and make sure their transmissions do not collide with the primary traffic. To avoid the interference to PUs, sensing becomes an indispensable part of CRN design.

S. Li is with the Department of CSE at The Ohio State University. Z. Zheng and E. Ekici are with the Department of ECE at The Ohio State University. N. B. Shroff is with the Department of ECE and CSE at The Ohio State University. Emails: li.908@osu.edu, {zhengz, ekici, shroff}@ece.osu.edu.
This grant has been supported in part via NSF grants CNS-1065136, CNS-1012700, CNS-0905408, and a MURI grant from the Army Research Office W911NF-08-1-0238.

Sensing can be performed via several methods, including energy detection, cyclostationary feature detection, and compressed sensing [1]. Energy detection is a simple method and requires no a priori knowledge of PU signals [26]. Its main disadvantage is its decreased accuracy in the presence of fading, shadowing, and unknown noise power profiles. For instance, if an SU suffers from shadowing or heavy fading, the sensed signal tends to be weak while the PU is transmitting, leading to incorrect decisions. To address these problems while maintaining sensing simplicity, cooperative sensing schemes that fuse the sensing results of multiple SUs have been proposed in the literature [13][16][18].

Cooperative sensing overcomes the shortcomings of individual sensing results by jointly processing observations. SUs report their individual sensing results, which are then used in a predefined decision rule to optimize an objective function. Examples of such functions include maximizing sensing accuracy (generally, a function of false alarm probability and misdetection probability) or maximizing the system throughput. Aside from sensing accuracy related metrics, cooperative sensing schemes are also designed to estimate the maximum transmit power for SUs so that they do not cause disruptive interference to PUs [14]. On the other hand, cooperative sensing incurs additional sensing delay over individual sensing.

Three main categories of decision rules have been identified in [1]: *Soft Combining, Quantized Soft Combining,* and *Hard Combining*. In each of them, a control channel is required to collect information. It is a common dedicated channel orthogonal to the PU channel as in [1]. However, different rules have different requirement on control channel bandwidth. In soft combining, raw sensing results, i.e., data sequences, are sent to the fusion center. In quantized soft combining, quantized sensing results are sent to the fusion center for soft combining to reduce the control channel communication overhead. While in hard combining, binary local decisions on the sensing results are reported. Compared to the other two categories, only a single bit is sent to the fusion center in hard combining.

Among hard combining rules, linear fusion rules [1] are widely applied to achieve a cooperative decision, such as AND, OR and majority rules [18]. AND and OR both take extreme approaches: In AND, only when all stations decide the channel to be "busy", the decision after fusion is "busy", which promotes the SU activity; In OR, only when all stations decide the channel to be "idle", the decision after fusion is "idle", which tends to protect the PU activity. The majority rule uses the majority of the local decisions as the final decision, which places it between AND and OR in terms of SU transmission eagerness. In addition, a linear-quadratic fusion rule that utilizes statistical knowledge [25] has been devised to capture the correlation between SUs in cooperative sensing. None of the aforementioned rules are shown to be optimal. In [25], the fusion rule is shown to not be optimal, and its performance compared to optimal has not been analytically characterized.

In this work, sensing quality is our main concern, which has two components: misdetection probability and false alarm probability. As a unifying objective, we adopt system throughput as a means to quantify the effects of misdetection and false alarm probabilities on the system performance. We design an optimal data fusion rule to (hard) combine the reported sensing results. More specifically, we aim to maximize the system throughput in a CRN composed of multiple PUs operating on orthogonal channels and SUs where SUs are allowed to sense all the channels in the network. We reflect two practical system requirements in the constraints: PU throughput above a threshold and limited sensing overhead. We assume that sensing decisions are made on a per channel basis and each SU can sense all the PUs in the system. Thus, we only need to solve the system throughput maximization problem per channel. Our main contributions can be summarized as follows:

- In contrast to previous works that restrict the class of fusion rules, we propose a Bayesian decision rule based algorithm to solve the throughput maximization problem optimally with constant time complexity.
- To guarantee that the PU throughput is at least a prescribed fraction δ of the PU capacity, we study the system throughput maximization problem with a constraint on the PU throughput. This constrained problem is shown to be strongly NP-hard by a reduction from the product partition problem [17]. A greedy algorithm is developed with the time complexity $O(\frac{N^5}{\log^2 \frac{1}{1-\epsilon}})$ where $\epsilon = 1 - 10^{-N10^{-r}}$, N is the total number of SUs, and r is the decimal places kept for the input. This approximation algorithm is analytically shown to achieve the system throughput (the sum of PU throughput and SU throughput) strictly greater than $\frac{1}{2}(1-\epsilon)$ of the optimal solution. The PU throughput fraction achieved is shown to be at least $\frac{\delta}{1-\epsilon} - \frac{\epsilon}{1-\epsilon}$.
- We investigate the system throughput maximization problem with two constraints: PU throughput is above a threshold and the number of sensing SUs is restricted. Our theoretical results show that the maximum system throughput is monotonic with respect to the sensing set. We propose a greedy heuristic whose performance is verified in simulation.

The paper is organized as follows: Related work is presented in Section II. In Section III, the system model is introduced and the sensing decision problem for maximizing system throughput subject to two constraints on PU throughput and the number of sensing SUs, respectively, is formulated. In Section IV, we solve the simple throughput maximization problem without any constraint optimally via the Bayesian decision rule, and then the PU throughput constrained maximization problem is studied, which is shown to be strongly NP-hard. A greedy algorithm is proposed with an approximation factor strictly greater than $\frac{1}{2}(1-\epsilon)$. The general problem is studied in Section V where the system throughput performance is investigated subject to two constraints on PU throughput and the number of sensing SUs used, respectively. In Section VI, numerical results are presented to illustrate the performance of our algorithms. We conclude in Section VII.

II. Related Work

Cooperative sensing solutions have been investigated in recent years. They rely on multiple SUs to exchange sensing results or a central controller to collect the sensing results from the SUs. The network is usually divided into clusters and each cluster head makes the decision on the channel occupancy. Collaborations among SUs have been shown to improve the efficiency of spectrum access and allow the relaxation of constraints at individual SUs [4][29]. One branch of the papers in cooperative sensing assume that the length of the sensing time at individual SUs is proportional to the sensing accuracy. However, longer sensing time decreases the transmission time. The trade-off is called the *sensing efficiency* problem and is discussed in [10] and [15]. In our work, we assume that the observation time at each SU is fixed so that the individual sensing accuracy does not depend on it. We focus on the optimal decision rule based on the sensing results collected.

Decision rules so far mainly focus on AND, OR, majority rules and other linear rules (the definitions provided in Section I). AND, OR are two extremes on SU transmission: AND promotes the SU activity while OR tends to protect the PU activity. Majority rule makes final decision on the majority of the local decisions. It is in the middle of AND, OR on aggressiveness. All these rules ignore the heterogeneity of SUs. Zhang et. al. [31] show that the best fusion rule among AND, OR, majority (half-voting) to minimize the cooperative sensing error rate is the half-voting rule in most cases. They show that AND or OR rules are better than half-voting only in rare cases. Based on these results, a fast spectrum sensing algorithm is proposed for a large network where not all SUs are required for sensing while satisfying a given error bound. However, the optimal number of sensing nodes and the complexity of this problem have not been discussed.

AND, OR, and Majority rules treat SUs equally and the different sensing abilities of SUs are not taken into account. In [18], the SU throughput is maximized subject to constraints on collisions with PUs. The optimal k-out-of-N fusion rule is determined and the sensing/throughput trade-off is also analyzed. As in [31], only AND, OR, and Majority rules are considered. They do not establish any conditions under which these rules are optimal. The spatial variation of SUs is considered by Shahid et. al. [22] and the fusion rule is a weighted combination of SU observations. The weight depends on the received power and path loss at each SU. Though more advanced than AND, OR, and majority rules, the weighted form is restricted to the linear function domain. In [5], optimal multi-channel cooperative sensing algorithms

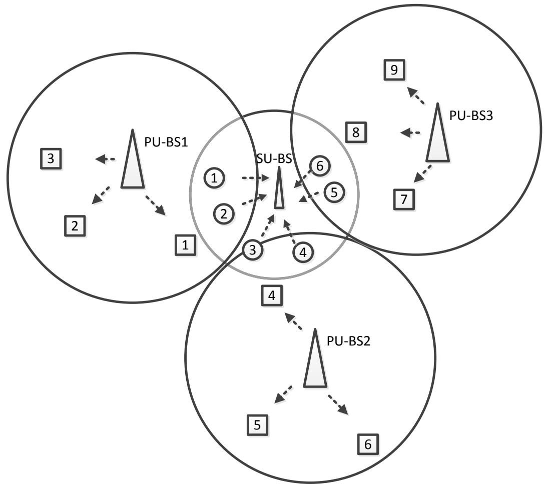


Fig. 1. System model of an SU network coexisting with three PU networks. Small circles are SUs and rectangles are PUs. Each big circle represents the transmission range of the corresponding SU-BS/PU-BS. Each SU can sense all PUs in the system.

are considered to maximize the SU throughput subject to per channel detection probability constraints. The resulting non-convex problem is solved by an iterative algorithm. Compared to [5], our work focuses on the maximization of the system throughput, including the PUs and SUs. Furthermore, our algorithm solves the system throughput maximization problem with constant time complexity, which is better than an iterative algorithm whose time complexity is high as shown in their simulations though not analytically established. Moreover, a soft decision rule is considered in [5], which requires significant amount of data to be transmitted to the coordinator while our hard decision rule requires only one bit of information to be sent from each SU. In [30], a general sensing quality metric is defined to measure the spectrum sensing accuracy. This metric does not characterize throughput as in this paper, but is rather designed to capture the correlations of sensing between SUs.

III. System Model and Problem Formulation

We consider a time-slotted cognitive radio network in which multiple PU networks, consisting of a PU base station (PU-BS) and PU receivers in each network, co-exist in the same area with an SU base station (SU-BS) and M SUs (Figure 1). Since PUs that are in the interference range of each other operate on orthogonal channels, for the purpose of the analysis, one can focus on a single PU and SU parameters corresponding to that PU. We consider the uplink part for the SU system, i.e., only one SU can be active and transmit to the SU-BS at any given time over the same channel. For each PU network, we denote the set of all SUs by $\mathcal{S}$ and the set of SUs whose uplink transmission causes interference to any PU receivers by S and $|S| = N$ ($|\mathcal{S}| = M \geq N$). They are indexed from 1 to N. SUs outside S can use the corresponding PU channel to transmit at any time slot without causing interference to the PUs. For instance, PUs 1, 4, and 8 lie in the interference range of SUs in Figure 1, and any transmission from SUs 1 and 2 may cause interference to PU 1.

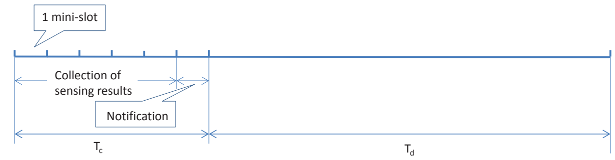


Fig. 2. Control slot T_c and data slot T_d.

A. Cooperative Sensing

SUs in S are close to the PU network and they are used to sense the channel cooperatively to reduce the sensing errors. The sensing results of individual SUs are assumed to be independent. We assume that each SU is allowed to sense any number of channels. Thus sensing decisions can be made per channel and we investigate the sensing behaviors of SUs on each channel separately. Let B represent the PU activity such that $B = 1$ if PU is active, and $B = 0$ otherwise. Let P_f^i denote the **probability of a false alarm** for SU i, which is the probability that SU i senses the PU to be active given that the PU is actually idle. P_m^i represents the **probability of mis-detection** for SU i, which is the probability that SU i senses the PU to be idle given that the PU is actually active. We assume that these probabilities are readily available as in [2][20]. In practice, they can be calculated by power level, noise power, path loss, etc, which can be learned from historical data.

Cooperative Sensing: Multiple SUs are chosen to sense the channel and the SU-BS predicts PU activity by collecting the sensing results from these SUs. We let S_0[1] denote the set of SUs that participate in cooperative sensing. Note that $S_0 \subseteq S \subseteq \mathcal{S}$. In the cooperative sensing model, we assume that the SU-BS collects sensing results from SUs in S_0.

Cooperative Sensing Indicator: Let Y_i denote how SU i senses the PU activity, which is a random variable. More precisely, $Y_i = 1$ indicates that SU i observes the PU to be active, while $Y_i = 0$ indicates that SU i observes the PU to be idle. In this paper, our objective is to maximize the system throughput by characterizing S_0 and estimating the PU activity based on observations from S_0 (called the decision rule). The decision rule is denoted as a function $F : \{0, 1\}^{|S_0|} \rightarrow \{0, 1\}$. The observations form a vector $\boldsymbol{Y}$, where $\boldsymbol{Y} \in \{0, 1\}^{|S_0|}$ while the decision is denoted by Z where $Z \in \{0, 1\}$, i.e., $Z = F(\boldsymbol{Y})$. The false alarm probability of cooperative sensing is denoted by $P_f^c = P(Z = 1|B = 0)$. The mis-detection probability of cooperative sensing is denoted by $P_m^c = P(Z = 0|B = 1)$. Each time slot is divided into a control slot T_c and a data slot T_d where $T_c + T_d = 1$ (Figure 2). In the control slot, the SU-BS collects sensing results from the set of SUs S_0 and notifies an SU in S if the cooperative sensing result is "idle" ($Z = 0$). If the PU is active (mis-detection), the PU transmission will collide with the transmission from the SU. The length T_c of the control slot is regarded as the sensing overhead and assumed to be constant throughout the paper, that is, a fixed time period is allocated for cooperative sensing in each slot.

The uplinks of SUs in S are assumed to have the same

[1]Note that $S_0 = S$ in Section IV where there is no budget constraint on the number of SUs sensing the channel; $S_0 \subseteq S$ in Section V where the size of S_0 is constrained.

TABLE I
NOTATION LIST

Symbol	Meaning
$\mathcal{S}$	The set of all SUs in the secondary network
M	Total number of SUs in the secondary network. $\lvert\mathcal{S}\rvert = M$
S	Set of SUs which cause interference to PU receivers
N	$\lvert S\rvert$
S_0	Set of SUs that are chosen to sense the channel. $S_0 \subseteq S$
P_f^i	False alarm probability of SU i
P_m^i	Mis-detection probability of SU i
P_f^c	False alarm probability of cooperative sensing
P_m^c	Mis-detection probability of cooperative sensing
T_c	Control slot
T_d	Data slot
π_0	Probability that the PU is idle
γ	Average throughput of PUs in the interference range of a SU
B	PU activity: 0 is idle and 1 is active
Z	Fusion decision on the PU activity of cooperative sensing
F	Decision rule: $\{0,1\}^{\lvert S_0\rvert} \to \{0,1\}$

capacity which is normalized to 1. We assume that SUs in S are always backlogged. The scheduling of the transmitting SUs is beyond the scope of this paper. However, any work-conserving scheduling policy operating on idle slots can be used together with the decision rule to maximize the total system throughput. We let π_0 denote the probability that the PU is idle and we assume that the prior distribution of PU activity is accurately acquired over time. Our only assumption is that state changes occur at the beginning of a time slot. The average throughput of PUs whose transmission would be interfered by SUs in S is denoted as γ. Table I summarizes the notations used in the paper.

B. Communication Model

The outline of operations for cooperative sensing is as follows.

1) Each SU i reports its probability of misdetection (P_m^i) and probability of false alarm (P_f^i) to the SU-BS;

2) The SU-BS determines the sensing set S_0 and the decision rule F based on P_m^i, P_f^i's and the optimization metric;

3) The SU-BS notifies SUs in S_0 with an ACK and also assigns each one of them a SEQ number for reporting sensing results;

4) SUs receiving an ACK sense the channel and report the results to SU-BS in the order of SEQ;

5) SU-BS makes the decision of the PU activity based on the sensing results and F and schedules an SU for transmission if the decision is 0 (PU idle).

Considering that the sensing decisions are made per channel, only a simplified single-channel problem is investigated in each section and the sum of maximum system throughput over each channel leads to the maximum system throughput of the entire PU-SU network.

C. Problem Formulation

In this section, we formulate the general cooperative sensing problem with the assumption that the sensing results from all SUs in the sensing set are reported to SU-BS within T_c. SUs outside S can transmit without causing interference to the PUs. Thus, their performance does not depend on the choice of the sensing set or the decision rule. Our goal is to maximize the sum of the expected throughput of the SUs in S and that of the PUs whose transmission may be interfered by the SUs, subject to the constraints of PU throughput and sensing budget. Instead of an abstract measure of sensing quality [30], we choose the system throughput to combine the effects of misdetection and false alarm probabilities in a meaningful manner. Misdetection and false alarm probabilities are related to PU throughput and SU throughput, respectively.

The system throughout then consists of both expected SU throughput and PU throughput defined as follows. Let $\boldsymbol{Y}$ denote a random vector of observation and $\boldsymbol{y}$ denote a concrete observation vector. Let y_i denote the observation that SU i senses the PU to be idle ($y_i = 0$) or busy ($y_i = 1$). Further, let $\boldsymbol{y}$ denote the observation vector, i.e., $\boldsymbol{y} = \{y_i\}_{i \in S}$. Now, given $B = 0$ (the PU is idle), the probability of a particular observation vector $\boldsymbol{y}$ occurring is

$$P(\boldsymbol{Y} = \boldsymbol{y} | B = 0) = \prod_{i \in S, y_i = 1} P_f^i \prod_{j \in S, y_j = 0} (1 - P_f^j). \quad (1)$$

Note that Y is a function of S_0, and we omit S_0 for simplicity. Now, the probability that the decision of cooperative sensing is idle given that the PU is idle involves summing over all values of $\boldsymbol{y}$, such that the decision $F(\boldsymbol{y}) = 0$, i.e.,

$$P(Z = 0 | B = 0) = \sum_{\boldsymbol{y}: F(\boldsymbol{y}) = 0} P(\boldsymbol{Y} = \boldsymbol{y} | B = 0). \quad (2)$$

Hence, the false alarm probability of cooperative sensing is

$$P_f^c = 1 - P(Z = 0 | B = 0) = 1 - \sum_{\boldsymbol{y}: F(\boldsymbol{y}) = 0} P(\boldsymbol{Y} = \boldsymbol{y} | B = 0). \quad (3)$$

Likewise, given $B = 1$ (the PU is active), the probability of a particular observation vector $\boldsymbol{y}$ occurring is

$$P(\boldsymbol{Y} = \boldsymbol{y} | B = 1) = \prod_{i \in S, y_i = 1} (1 - P_m^i) \prod_{j \in S, y_j = 0} P_m^j. \quad (4)$$

And

$$P(Z = 1 | B = 1) = \sum_{\boldsymbol{y}: F(\boldsymbol{y}) = 1} P(\boldsymbol{Y} = \boldsymbol{y} | B = 1). \quad (5)$$

Then, the mis-detection probability of cooperative sensing is

$$P_m^c = 1 - P(Z = 1 | B = 1) = 1 - \sum_{\boldsymbol{y}: F(\boldsymbol{y}) = 1} P(\boldsymbol{Y} = \boldsymbol{y} | B = 1). \quad (6)$$

Note that Equation (2) is the conditional probability that SU-BS correctly identifies the PU activity when it is idle so that one SU could transmit successfully; Equation (5) is the conditional probability that SU-BS correctly detects the PU is active so that no SU would transmit and the PU could transmit successfully. Hence, the *expected throughput of the SUs* can be represented by

$$(1 - T_c) P(B = 0, Z = 0) = (1 - T_c) \pi_0 P(Z = 0 | B = 0)$$

$$= (1 - T_c) \pi_0 \sum_{\boldsymbol{y}: F(\boldsymbol{y}) = 0} P(\boldsymbol{Y} = \boldsymbol{y} | B = 0), \quad (7)$$

since the uplinks of SUs in S have unit capacity and only one of them could be scheduled in each time slot. Now, let the PU capacity be γ. Then, the *expected throughput of the PU* in the interference range of the SUs is given by (8).

$$\gamma P(Z=1|B=1) = \gamma \sum_{\boldsymbol{y}:F(\boldsymbol{y})=1} P(\boldsymbol{Y}=\boldsymbol{y}|B=1) \quad (8)$$

The problem of interest to us is formulated as follows:

Problem (A):

$$\max_{F,S_0} (1-T_c)\pi_0 \sum_{\boldsymbol{y}:F(\boldsymbol{y})=0} P(\boldsymbol{Y}(\boldsymbol{S_0})=\boldsymbol{y}|B=0)$$
$$+\gamma \sum_{\boldsymbol{y}:F(\boldsymbol{y})=1} P(\boldsymbol{Y}(\boldsymbol{S_0})=\boldsymbol{y}|B=1)$$
$$\text{s.t. } \gamma \sum_{\boldsymbol{y}:F(\boldsymbol{y})=1} P(\boldsymbol{Y}(\boldsymbol{S_0})=\boldsymbol{y}|B=1) \geq \alpha, \quad (9)$$
$$|S_0| \leq k. \quad (10)$$

Note that the system throughput as the objective function in Problem (A) combines the effects of misdetection and false alarm probabilities, which is more meaningful than an arbitrary weighted sum of them. A nice property of this objective function is that the SU capacity $(1-T_c)\pi_0$ and PU capacity γ are taken into account. Note that constraint (9) indicates that the PU throughput is above a threshold α, and constraint (10) requires the sensing budge within k SUs. We study the solutions step by step in Sections IV and V.

IV. Guaranteeing a Target PU Throughput

In this section, we take the first step by investigating the maximum throughput problem under a single PU throughput constraint (Problem (B)). The reason why this is important is to ensure that the PU receives at least a guaranteed amount of throughput. We show that this constrained problem is strongly NP-hard by reducing the classical product partition problem [17] to it. Then a greedy approximation algorithm is proposed to achieve throughput that is strictly greater than $\frac{1}{2}(1-\epsilon)$ of the optimal solution. The complexity of the algorithm is shown to be $O(\frac{N^5}{\log^2 \frac{1}{1-\epsilon}})$ by solving a two-dimensional dynamic programming problem. Note that the algorithm only needs to run once until P_m^i or P_f^i changes.

Problem (B):

$$\max_{F} (1-T_c)\pi_0 \sum_{\boldsymbol{y}:F(\boldsymbol{y})=0} P(\boldsymbol{Y}=\boldsymbol{y}|B=0)$$
$$+\gamma \sum_{\boldsymbol{y}:F(\boldsymbol{y})=1} P(\boldsymbol{Y}=\boldsymbol{y}|B=1)$$
$$\text{s.t. } \gamma \sum_{\boldsymbol{y}:F(\boldsymbol{y})=1} P(\boldsymbol{Y}=\boldsymbol{y}|B=1) \geq \alpha. \quad (11)$$

Equation (11) is the constraint we put on Problem (B) where the expected PU throughput must be no less than a preset system-dependent threshold. Note that $\alpha = \delta \cdot \gamma$ where δ is the fraction of the full PU throughput without any SU transmitting.

Algorithm 1 Bayesian Decision Rule Based Algorithm for maximizing the system throughput (given $\boldsymbol{Y}=\boldsymbol{y}$, decide Z)

```
1: if (1 − T_c)π_0 ∏_{y_i=1} P_f^i ∏_{y_j=0} (1 − P_f^j) ≥ γ ∏_{y_i=1} (1 − P_m^i) ∏_{y_j=0} P_m^j then
2:     Z ← 0
3: else
4:     Z ← 1
```

Problem (B) maximizes the expected system throughput given that the lowest PU throughput can be met. This is important because in cognitive radio applications, PU transmissions need to have higher importance than SU transmissions. Note that for the multichannel formulation, α in Equation (11) varies over difference PUs. By solving the constrained optimization problem on each channel and summing the throughput, we get the optimal system throughput across all channels subject to the throughput constraints of all PUs.

A. Bayesian Decision Rule Based Algorithm

To solve Problem (B), we start with Problem (C) which is an unconstrained problem and propose an optimal solution with Bayesian decision rule. Based on it, we will investigate Problem (B) in Section IV-B.

Problem (C):

$$\max_{F} (1-T_c)\pi_0 \sum_{\boldsymbol{y}:F(\boldsymbol{y})=0} P(\boldsymbol{Y}=\boldsymbol{y}|B=0)$$
$$+\gamma \sum_{\boldsymbol{y}:F(\boldsymbol{y})=1} P(\boldsymbol{Y}=\boldsymbol{y}|B=1)$$

We show that Problem (C) can be converted to a Bayesian Decision problem. Algorithm 1 is then developed based on the Bayesian decision rule to minimize the posterior expected loss [3] and it is of constant time complexity.

To solve Problem (C), we formulate the equivalent problem as follows.

$$\min_{F} L(B=0,Z=1)\left[\pi_0 \sum_{\boldsymbol{y}:F(\boldsymbol{y})=1} P(\boldsymbol{Y}=\boldsymbol{y}|B=0)\right]$$
$$+L(B=1,Z=0)\left[(1-\pi_0) \sum_{\boldsymbol{y}:F(\boldsymbol{y})=0} P(\boldsymbol{Y}=\boldsymbol{y}|B=1)\right], \quad (12)$$

where $L(B,Z)$ is the loss of decision Z based on observation $\boldsymbol{x}$, which is a non-negative number. $L(B=0,Z=1) = (1-T_c)$ and $L(B=1,Z=0) = \frac{\gamma}{1-\pi_0}$. Thus Equation (12) is the *posterior expected loss* of decision Z (Definition 8 of Chapter 4.4 in [3]). Using the Bayesian decision rule, Problem (12) can be solved optimally [3]: given $\boldsymbol{Y}=\boldsymbol{y}$, the decision $Z=1$ if $L(B=0,Z=1)\pi_0 P(\boldsymbol{Y}=\boldsymbol{y}|B=0) < L(B=1,Z=0)(1-\pi_0)P(\boldsymbol{Y}=\boldsymbol{y}|B=1)$ and $Z=0$ otherwise. Algorithm 1 is designed accordingly.

B. Achieving a Target PU Throughput

Based on the optimal solution to Problem (C), we propose a greedy approximation algorithm to solve Problem (B). We define $\mathcal{Y}_0 = \{\boldsymbol{y}|F_b(\boldsymbol{y})=0\}$, $\mathcal{Y}_1 = \{\boldsymbol{y}|F_b(\boldsymbol{y})=1\}$ where F_b

is the Bayesian rule. We also define $\mathcal{Y}_0^* = \{\boldsymbol{y}|F^*(\boldsymbol{y}) = 0\}$ and $\mathcal{Y}_1^* = \{\boldsymbol{y}|F^*(\boldsymbol{y}) = 1\}$ where F^* is the optimal solution to Problem (B). For convenience, we define $G(\boldsymbol{y}) = (1 - T_c)\pi_0 P(\boldsymbol{Y} = \boldsymbol{y}|B = 0)$ and $H(\boldsymbol{y}) = \gamma P(\boldsymbol{Y} = \boldsymbol{y}|B = 1)$. Note that if $G(\boldsymbol{y}) \geq H(\boldsymbol{y})$, we have $F_b(\boldsymbol{y}) = 0$; otherwise $F_b(\boldsymbol{y}) = 1$. By observing the structure of Problem (B), we state Lemma 4.1 and show that $\mathcal{Y}_1 \subseteq \mathcal{Y}_1^*$. *In other words, observations that have decision* 1 *by the Bayesian rule have decision* 1 *as well in the optimal decision.*

Lemma 4.1: In the optimal solution to Problem (B), all observations $\boldsymbol{y}$ with $F_b(\boldsymbol{y}) = 1$ has the property that $F^*(\boldsymbol{y}) = 1$.

(By contradiction) Assume that $\mathcal{Y}_1 \not\subseteq \mathcal{Y}_1^*$, that is, $F_b(\boldsymbol{y}) = 1$ and $F^*(\boldsymbol{y}) = 0$ for some $\boldsymbol{y}$ ($\boldsymbol{y} \in \mathcal{Y}_1 \cap \mathcal{Y}_0^*$). By Bayesian rule, it means $G(\boldsymbol{y}) < H(\boldsymbol{y})$. We find another rule $\tilde{F}$ where $\tilde{F}(\boldsymbol{y}) = 1$ if $\boldsymbol{y} \in \mathcal{Y}_1 \cap \mathcal{Y}_0^*$; otherwise $\tilde{F}(\boldsymbol{y}) = F^*(\boldsymbol{y})$. Obviously, $\sum\limits_{\boldsymbol{y}:\tilde{F}(\boldsymbol{y})=1} P(\boldsymbol{Y} = \boldsymbol{y}|B = 1) > \sum\limits_{\boldsymbol{y}:F^*(\boldsymbol{y})=1} P(\boldsymbol{Y} = \boldsymbol{y}|B = 1)$ so that this operation still results in a feasible solution. Furthermore, the expected system throughput increases considering $G(\boldsymbol{y}) < H(\boldsymbol{y})$, which results in a better solution than the current optimal one. It causes a contradiction. Hence, we have $F^*(\boldsymbol{y}) = 1$ for all $\boldsymbol{y} \in \mathcal{Y}_1$.

With the property of Lemma 4.1, to solve Problem (B), we only need to find the set $\mathcal{Y}_0 \setminus \mathcal{Y}_0^*$ that is composed of observations $\boldsymbol{y}$ with $F_b(\boldsymbol{y}) = 0$ and $F^*(\boldsymbol{y}) = 1$.

a) **Proof of Strong NP-hardness:** We show that Problem (B) is strongly NP-hard. By Lemma 4.1, it suffices to show the following problem to be strongly NP-hard: finding all $\boldsymbol{y}$ with $F_b(\boldsymbol{y}) = 0$ and $F^*(\boldsymbol{y}) = 1$. Recall that a problem is said to be strongly NP-complete, if it remains so even when all of its numerical parameters are bounded by a polynomial in the length of the input. A problem is strongly NP-hard if a strong NP-complete problem can be reduced to it in polynomial time [7].

Theorem 4.2: Problem (B) is strongly NP-hard.

Proof: We will reduce the product partition problem to the equivalent problem stated above and the strong NP-hardness of Problem (B) can be proved accordingly. We first state the product partition problem [17] - Given N positive integers: y_1, $\cdots$, y_N, is there a way to have them partitioned into two equal-sized subsets that have the same product? For the reduction, we construct an instance of Problem (B) by setting $(1-T_c)\pi_0 = \gamma$, $\alpha = \epsilon + \sum\limits_{\boldsymbol{y}:G(\boldsymbol{y})<H(\boldsymbol{y})} H(\boldsymbol{y})$ with $\epsilon \leq \min\limits_{\boldsymbol{y}:G(\boldsymbol{y})\geq H(\boldsymbol{y})} H(\boldsymbol{y})$. For this instance, putting any $\boldsymbol{y}$ with $G(\boldsymbol{y}) \geq H(\boldsymbol{y})$ to $Z = 1$ would make a feasible solution given that observations with $G(\boldsymbol{y}) < H(\boldsymbol{y})$ have all been put in $Z = 1$. Choosing the observation with the minimum non-negative $G(\boldsymbol{y}) - H(\boldsymbol{y})$ would be the optimal solution. Note that $G(\boldsymbol{y}) - H(\boldsymbol{y}) = 0$ is equivalent to $\frac{G(\boldsymbol{y})}{H(\boldsymbol{y})} = 1$. By setting $\frac{1-P_f^i}{P_m^i} = \frac{1-P_m^i}{P_f^i} = \eta_i$ for all i, we have $\frac{G(\boldsymbol{y})}{H(\boldsymbol{y})} = \prod\limits_{y_i=0, i=1,\cdots,N} \eta_i \cdot \prod\limits_{y_j=1, j=1,\cdots,N} \frac{1}{\eta_j}$. Now the instance becomes: given N pairs of integers $(\eta_1, \frac{1}{\eta_1}), \cdots, (\eta_N, \frac{1}{\eta_N})$, exactly one number should be chosen from each pair; with this constraint, what is the minimum product that is no less than 1? Note that the operations above take polynomial time.

To verify the correctness of the reduction, we can check: if the minimum $\frac{G(\boldsymbol{y})}{H(\boldsymbol{y})}$ no less than 1 is 1, that is, the optimal solution of the instance is 1, we can answer "Yes" to the partition problem; if it is greater than 1, we can answer "No" to the partition problem. If Problem (B) can be solved in polynomial time, then the product partition problem can be solved in polynomial time as well. The product partition problem is well-known to be strongly NP-complete [17]. Assuming P≠NP, Problem (B) has been proven to be strongly NP-hard. ■

Algorithm 2 Greedy Approximation Algorithm for Problem (B)

Input: N, T_c, π_0, γ, α, P_m^i, P_f^i for all i
Output: F or "infeasible"
1: $G(\boldsymbol{y}) \leftarrow (1 - T_c)\pi_0 \prod\limits_{i\in S, y_i=1} P_f^i \prod\limits_{j\in S, y_j=0} (1 - P_f^j)$ for all $\boldsymbol{y}$
2: $H(\boldsymbol{y}) \leftarrow \gamma \prod\limits_{i\in S, y_i=1} (1 - P_m^i) \prod\limits_{j\in S, y_j=0} P_m^j$ for all $\boldsymbol{y}$
3: **if** $\gamma < \alpha$ **then**
4: output "infeasible" and return
5: $F(\boldsymbol{y}) \leftarrow 1$ for all $\boldsymbol{y}$, $sum1 \leftarrow \sum\limits_{\boldsymbol{y}:G(\boldsymbol{y})<H(\boldsymbol{y})} H(\boldsymbol{y})$
6: **if** $sum1 \geq \alpha$ **then**
7: $F(\boldsymbol{y}) = 0$ for all $\boldsymbol{y}$ with $G(\boldsymbol{y}) \geq H(\boldsymbol{y})$ and return
8: Sort $\boldsymbol{y}$'s with $G(\boldsymbol{y}) \geq H(\boldsymbol{y})$ in non-increasing order of $\frac{G(\boldsymbol{y})}{H(\boldsymbol{y})}$ and denote them as $\{\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_l\}$
9: $sum2 \leftarrow 0$
10: **for** $i = 1$ to l **do**
11: **if** $sum2 + H(\boldsymbol{y}_i) > \gamma - \alpha$ **then** break
12: $sum2 \leftarrow sum2 + H(\boldsymbol{y}_i)$, $F(\boldsymbol{y}_i) \leftarrow 0$
13: **if** $\sum\limits_{n=1}^{i-1} (G(\boldsymbol{y}_n) - H(\boldsymbol{y}_n)) < G(\boldsymbol{y}_i) - H(\boldsymbol{y}_i)$ **then**
14: $F(\boldsymbol{y}_n) \leftarrow 0$ for all $n = 1, \cdots, i-1$

It has been shown in Theorem 4.2 that finding the observation with $G(\boldsymbol{y})$ closest to $H(\boldsymbol{y})$ from above is strongly NP-hard. Hence, unless P=NP, one cannot even find a pseudo-polynomial time algorithm to solve Problem (B) [28]. Hence, we next focus on designing a good approximation algorithm.

b) **Greedy Approximation Algorithm:** We propose a greedy algorithm (Algorithm 2) that initially assigns all observations to $Z = 1$ and then moves observations with $G(\boldsymbol{y}) \geq H(\boldsymbol{y})$ by $\frac{G(\boldsymbol{y})}{H(\boldsymbol{y})}$ from the highest to lowest to $Z = 0$ until the feasibility constraint of Problem (B) is violated. By transforming Problem (B) into the Knapsack Problem [28], we will show that the algorithm achieves strictly greater than $1/2$ of the optimal solution for Problem (B). Although the sum of $G(\boldsymbol{y})$ or $H(\boldsymbol{y})$ in the worst case has an exponential number of terms, we will design an approximation algorithm.

In Algorithm 2, observations are chosen by $\frac{G(\boldsymbol{y})}{H(\boldsymbol{y})}$ from the highest to the lowest and assigned to $Z = 0$ after those with $G(\boldsymbol{y}) < H(\boldsymbol{y})$ are assigned to $Z = 1$. Ties are broken by putting observations with smaller $H(\boldsymbol{y})$ in the front. In Lines 3-4, the algorithm checks whether a feasible solution exists for the given input by comparing the extreme case where all observations are assigned to $Z = 1$ ($\sum\limits_{\boldsymbol{y}} H(\boldsymbol{y}) = \gamma$) with the threshold α. In Line 5, observations are initialized to $Z = 1$. Lines 5-7 checks whether the feasibility constraint in Problem (B) has been satisfied under the initial assignment. If yes, observations with $G(\boldsymbol{y}) \geq H(\boldsymbol{y})$ are assigned to $Z = 0$ by Bayesian decision rule. Lines 8-12 searches for observations with $G(\boldsymbol{y}) \geq H(\boldsymbol{y})$ from the highest $\frac{G(\boldsymbol{y})}{H(\boldsymbol{y})}$ to lowest until $sum2 + H(\boldsymbol{y}_i) \leq \gamma - \alpha$ is violated (Line 11).

Note that $\sum_{\boldsymbol{y}:F(\boldsymbol{y})=0} H(\boldsymbol{y}) \leq \gamma - \alpha$ and $\sum_{\boldsymbol{y}:F(\boldsymbol{y})=1} H(\boldsymbol{y}) \geq \alpha$ (feasibility constraint) are equivalent since $\sum_{\boldsymbol{y}} H(\boldsymbol{y}) = \gamma$. $F(\boldsymbol{y})$ of these observations are set to be 0 (Line 12) in the searching process. To guarantee the $\frac{1}{2}$ approximation ratio, we have to do the comparison in Lines 13-14 (shown in the proof of Theorem 4.3). Next, we state Theorem 4.3 that gives the approximation factor of Algorithm 2.

Theorem 4.3: Algorithm 2 achieves strictly greater than $1/2$ of the optimal solution to Problem (B).

Proof: We define $A = \sum_{\boldsymbol{y}:\boldsymbol{y}\in\mathcal{Y}_1} H(\boldsymbol{y})$, $B = \sum_{\boldsymbol{y}:\boldsymbol{y}\in\mathcal{Y}_0\setminus\mathcal{Y}_0^*} G(\boldsymbol{y})$, $B' = \sum_{\boldsymbol{y}:\boldsymbol{y}\in\mathcal{Y}_0\setminus\mathcal{Y}_0^*} H(\boldsymbol{y})$ ($B \geq B'$ by Bayesian rule), $C = \sum_{\boldsymbol{y}:\boldsymbol{y}\in\mathcal{Y}_0^*} G(\boldsymbol{y})$, and $C' = \sum_{\boldsymbol{y}:\boldsymbol{y}\in\mathcal{Y}_0^*} H(\boldsymbol{y})$ ($C \geq C'$ by Bayesian rule). Then, $A + B + C$ is the optimal solution to Problem (B) without the PU throughput constraint since

$$A + B + C = A + (B + C) = \sum_{\boldsymbol{y}:\boldsymbol{y}\in\mathcal{Y}_1} H(\boldsymbol{y}) + \sum_{\boldsymbol{y}:\boldsymbol{y}\in\mathcal{Y}_0} G(\boldsymbol{y})$$

and $A + B' + C$ is the optimal solution to Problem (B) since

$$A + B' + C = (A + B') + C = \sum_{\boldsymbol{y}:\boldsymbol{y}\in\mathcal{Y}_1^*} H(\boldsymbol{y}) + \sum_{\boldsymbol{y}:\boldsymbol{y}\in\mathcal{Y}_0^*} G(\boldsymbol{y})$$

which is no greater than $A + B + C$. Note that $A + B' + C' = \sum_{\boldsymbol{y}} H(\boldsymbol{y}) = \gamma$. Let APX be the solution to Problem (B) output by Algorithm 2. Let OPT be the optimal solution to Problem (B). Then, we have

$$\begin{aligned} OPT &= A + B' + C \\ &= \gamma + (C - C'). \end{aligned} \tag{13}$$

We then show that $\mathcal{Y}_0^*$ is the optimal solution to Problem (14) and $C - C' = \sum_{\boldsymbol{y}:\boldsymbol{y}\in\mathcal{Y}_0^*} (G(\boldsymbol{y}) - H(\boldsymbol{y}))$ is the optimal objective value.

$$\begin{aligned} \max_{W:W\subseteq\mathcal{Y}_0} & \sum_{\boldsymbol{y}:\boldsymbol{y}\in W} (G(\boldsymbol{y}) - H(\boldsymbol{y})) \\ \text{s.t.} & \sum_{\boldsymbol{y}:\boldsymbol{y}\in W} H(\boldsymbol{y}) \leq \gamma - \alpha \end{aligned} \tag{14}$$

Clearly, we only need to show that the constraint of Problem (14) and that of Problem (B) are equivalent. By definitions, we have

$$\begin{aligned} & \gamma \sum_{\boldsymbol{y}:F(\boldsymbol{y})=1} P(\boldsymbol{Y} = \boldsymbol{y} | B = 1) \geq \alpha \\ \Leftrightarrow & \sum_{\boldsymbol{y}:F(\boldsymbol{y})=1} H(\boldsymbol{y}) \geq \alpha \\ \overset{(a)}{\Leftrightarrow} & \sum_{\boldsymbol{y}:\boldsymbol{y}\in\mathcal{Y}_1} H(\boldsymbol{y}) + \sum_{\boldsymbol{y}:\boldsymbol{y}\in\mathcal{Y}_0\setminus W} H(\boldsymbol{y}) \geq \alpha \\ \Leftrightarrow & \sum_{\boldsymbol{y}:\boldsymbol{y}\in\mathcal{Y}_1} H(\boldsymbol{y}) + \sum_{\boldsymbol{y}:\boldsymbol{y}\in\mathcal{Y}_0\setminus W} H(\boldsymbol{y}) - \alpha \geq 0 \\ \Leftrightarrow & \sum_{\boldsymbol{y}:\boldsymbol{y}\in\mathcal{Y}_1} H(\boldsymbol{y}) + \sum_{\boldsymbol{y}:\boldsymbol{y}\in\mathcal{Y}_0\setminus W} H(\boldsymbol{y}) \\ & + \sum_{\boldsymbol{y}:\boldsymbol{y}\in W} H(\boldsymbol{y}) - \alpha \geq \sum_{\boldsymbol{y}:\boldsymbol{y}\in W} H(\boldsymbol{y}) \end{aligned}$$

Algorithm 3 Algorithm to Find the Joint Distribution of $(\log \frac{G(\boldsymbol{y})}{H(\boldsymbol{y})}, \log H(\boldsymbol{y}))$

Input: N, P_f^i, P_m^i for all i, r
Output: $C(N, j, j')$ for all j, j'
1: $a_i \leftarrow round(\log \frac{1-P_f^i}{P_m^i}, r) \times 10^r$ for all i
2: $z_i \leftarrow round(\log \frac{P_f^i}{1-P_m^i}, r) \times 10^r$ for all i
3: $\lambda_i \leftarrow round(\log P_m^i, r) \times 10^r$ for all i
4: $\mu_i \leftarrow round(\log(1 - P_m^i), r) \times 10^r$ for all i
5: $Q \leftarrow \sum_{i=1}^{N} \max\{a_i, z_i\}$, $q \leftarrow \sum_{i=1}^{N} \min\{a_i, z_i\}$
6: $Q' \leftarrow \max\{\max_i \lambda_i, \max_i \mu_i\}$, $q' \leftarrow \sum_{i=1}^{N} \min\{\lambda_i, \mu_i\}$
7: $C(i, j, j') \leftarrow 0$ for all i, j, j', $C(1, a_1, \lambda_1) \leftarrow 1$, $C(1, z_1, \mu_1) \leftarrow 1$
8: **for** $i = 1$ to $N - 1$ **do**
9:     **for** $j = q$ to Q **do**
10:         **for** $j' = q'$ to Q' **do** $C(i+1, j, j') = C(i, j - a_{i+1}, j' - \lambda_{i+1}) + C(i, j - z_{i+1}, j' - \mu_{i+1})$

$$\Leftrightarrow \sum_{\boldsymbol{y}:\boldsymbol{y}\in W} H(\boldsymbol{y}) \leq \gamma - \alpha.$$

Note that (a) holds because W corresponds to all observations $\boldsymbol{y}$ with $F(\boldsymbol{y}) = 0$. Problem (14) is a Knapsack Problem and can be solved by a greedy approach [28]: choosing observations with $G(\boldsymbol{y}_i) \geq H(\boldsymbol{y}_i)$ from the highest $\frac{G(\boldsymbol{y}_i)-H(\boldsymbol{y}_i)}{H(\boldsymbol{y}_i)}$ to the lowest until (14) is violated (the index of the observation added when the constraint is violated is labeled as s), which is exactly what we do in Algorithm 2 since $\frac{G(\boldsymbol{y}_i)-H(\boldsymbol{y}_i)}{H(\boldsymbol{y}_i)} \geq \frac{G(\boldsymbol{y}_j)-H(\boldsymbol{y}_j)}{H(\boldsymbol{y}_j)}$ if and only if $\frac{G(\boldsymbol{y}_i)}{H(\boldsymbol{y}_i)} \geq \frac{G(\boldsymbol{y}_j)}{H(\boldsymbol{y}_j)}$; A further comparison to find the maximum of $\sum_{n=1}^{s-1} (G(\boldsymbol{y}_n) - H(\boldsymbol{y}_n))$ and $G(\boldsymbol{y}_s) - H(\boldsymbol{y}_s)$ guarantees $\frac{1}{2}(C - C')$ [28]. Hence, $APX \geq \gamma + 1/2(C - C')$ holds. Since $\gamma > 0$, we always have $APX/OPT > 1/2$ for Problem (B). ■

So far, we have shown that the greedy algorithm (Algorithm 2) gives an approximation factor of strictly greater than $1/2$ for Problem (B). However, when $\gamma \gg C - C'$, this factor could be arbitrarily close to 1.

c) **Approximate Throughput Calculation:** To calculate system throughput in Algorithm 2, we need to calculate the sum of $G(\boldsymbol{y})$ or $H(\boldsymbol{y})$. The worst case complexity is exponential since there are exponential number of terms due to its combinatorial nature. To address this problem, we design an approximation algorithm via dynamic programming to estimate the joint distribution of $(\log \frac{G(\boldsymbol{y})}{H(\boldsymbol{y})}, \log H(\boldsymbol{y}))$, by counting the number of observations with the same $\log \frac{G(\boldsymbol{y})}{H(\boldsymbol{y})}$ and the same $\log H(\boldsymbol{y})$. Note that 1) We take logarithmic functions to make the recursive function additive (Line 10 in Algorithm 3); 2) We find the joint distribution of $(\log \frac{G(\boldsymbol{y})}{H(\boldsymbol{y})}, \log H(\boldsymbol{y}))$ instead of $(\log G(\boldsymbol{y}), \log H(\boldsymbol{y}))$ because we need to evaluate $\frac{G(\boldsymbol{y})}{H(\boldsymbol{y})}$ in Lines 5 and 8 in Algorithm 2. The details of the algorithm will be introduced next, followed by the complexity analysis. For simplicity, we assume $(1 - T_c)\pi_0 = \gamma$ in Algorithm 3. This assumption is only for ease of illustration, and is relaxed in our online technical report [11].

In Algorithm 3, we use dynamic programming to calculate the joint distribution of $\log \frac{G(\boldsymbol{y})}{H(\boldsymbol{y})}$ and $\log H(\boldsymbol{y})$, which counts the number of observations with the same $\log \frac{G(\boldsymbol{y})}{H(\boldsymbol{y})}$ and the same $\log H(\boldsymbol{y})$. Note that we only need to run this algorithm once in the time period where P_m^i, P_f^i, π_0 and γ are fixed. Lines 3, 6, 9 and 12 of Algorithm 2 can be calculated based on these counts. $round(b, r)$ rounds b to r decimal places by removing all digits after r decimal places. We use $round(b, r) \times 10^r$ to scale and round a real b to an integer. The values of r lead to different accuracy levels for the algorithm. Q and q specify the maximum and minimum contribution, respectively, an observation $\boldsymbol{y}$ can have to $\log \frac{G(\boldsymbol{y})}{H(\boldsymbol{y})}$, while Q' and q' specify the maximum and minimum contribution an observation $\boldsymbol{y}$ can have to $\log H(\boldsymbol{y})$ respectively. Let $\boldsymbol{y^i} = \{y_1, \cdots, y_i\}$ denote the observation vector for SUs 1 to i. $C(i, j, j')$ is defined as the number of observations with $\log \frac{G(\boldsymbol{y^i})}{H(\boldsymbol{y^i})}$ (after rounding) equal to j and $\log H(\boldsymbol{y^i})$ (after rounding) equal to j'. In particular, $C(N, j, j')$ records the number of observations with $\log \frac{G(\boldsymbol{y^N})}{H(\boldsymbol{y^N})}$ (after rounding) equal to j and $\log H(\boldsymbol{y^N})$ (after rounding) equal to j'. Lines 7-10 use iterations to find $C(i, j, j')$ for all $i = 1, \cdots, N$, $q \leq j \leq Q$ and $q' \leq j' \leq Q'$. The recursive function in Line 10 distinguishes two situations: if $y_{i+1} = 0$, $\log \frac{G(\boldsymbol{y^i})}{H(\boldsymbol{y^i})}$ is increased by a_{i+1} and $\log H(\boldsymbol{y^i})$ is increased by λ_{i+1}; on the other hand, if $y_{i+1} = 1$, $\log \frac{G(\boldsymbol{y^i})}{H(\boldsymbol{y^i})}$ is increased by z_{i+1} and $\log H(\boldsymbol{y^i})$ is increased by μ_{i+1}. Note that Line 10 may encounter $C(i, j, j')$ beyond the boundaries of j or j', the value of which will be treated as 0. Lines 3, 6, 9 and 12 of Algorithm 2 can be calculated accordingly, the time complexity of which is dominated by that of Algorithm 3. For special cases satisfying one or more of the following conditions: $P_m^i = 0$, $P_m^i = 1$, $P_f^i = 0$ and $P_f^i = 1$, the values of G or H are straightforward which does not require running Algorithm 3. For instance, when $P_m^i = 0$, $H(\boldsymbol{y}) = 0$ for all observations with $y_i = 0$. In the following, we consider only other more general cases. We will first show the tradeoff between the accuracy of $G(\boldsymbol{y})$ or $H(\boldsymbol{y})$ calculation and the time complexity of Algorithm 3 in Lemma 4.4. Next, we prove that Algorithm 2, together with Algorithm 3, can achieve strictly greater than $\frac{1}{2}(1-\epsilon)$ of the optimal solution, where $\epsilon \in (0, 1)$ is a constant, with the time complexity $O(\frac{N^5}{\log^2 \frac{1}{1-\epsilon}})$, and we also bound the feasibility gap in Theorem 4.5. As ϵ decreases, better accuracy is achieved at the cost of higher time complexity. The algorithm only needs to run once before P_m^i or P_f^i changes. We define $G'(\boldsymbol{y})$ and $H'(\boldsymbol{y})$ as the values of $G(\boldsymbol{y})$ and $H(\boldsymbol{y})$, respectively, calculated by Algorithm 3.

Lemma 4.4: With the complexity of $O(\frac{N^5}{\log^2 \frac{1}{1-\epsilon}})$, Algorithm 3 calculates $G'(\boldsymbol{y}) \geq (1-\epsilon) G(\boldsymbol{y})$ and $H'(\boldsymbol{y}) \geq (1-\epsilon) H(\boldsymbol{y})$.

Proof: The rounding in Lines 1-4 makes $\log \frac{1-P_f^i}{P_m^i}$, $\log \frac{P_f^i}{1-P_m^i}$, $\log P_m^i$ and $\log 1 - P_m^i$ lose at most 10^{-r} in their values, respectively. By the definition of $G(\boldsymbol{y})$, we have $\log G(\boldsymbol{y}) - \log G'(\boldsymbol{y}) \leq N 10^{-r}$, which is equivalent to $G'(\boldsymbol{y}) \geq 10^{-N10^{-r}} G(\boldsymbol{y})$. Similarly, we have $H'(\boldsymbol{y}) \geq 10^{-N10^{-r}} H(\boldsymbol{y})$. Let $\epsilon = 1 - 10^{-N10^{-r}}$, then given the input P_m^i and P_f^i, the complexity of Algorithm 3 is $O(N^3 10^{2r})$, which is $O(N^5 / \log^2 \frac{1}{1-\epsilon})$. ∎

Based on Lemma 4.4, we prove the approximation factor of $\frac{1}{2}(1-\epsilon)$ in the following theorem. We also characterize the feasibility gap which tends to 0 as ϵ goes to 0.

Theorem 4.5: Algorithm 2, together with Algorithm 3, achieves strictly greater than $\frac{1}{2}(1-\epsilon)$ of the optimal solution with the time complexity of $O(\frac{N^5}{\log^2 \frac{1}{1-\epsilon}})$; it also achieves a PU throughput fraction of at least $\frac{\delta}{1-\epsilon} - \frac{\epsilon}{1-\epsilon}$ where $\delta \cdot \gamma = \alpha$.

Proof: As in the proof of Theorem 4.3, we focus on the equivalent Problem (14). We denote the optimal assignment of observations without any approximations of G or H by Γ, the optimal assignment of observations with the approximations of G or H in Algorithm 3 by Γ', and the assignment generated by Algorithm 2 with Algorithm 3 by Γ'_g. We also denote the value of the objective function in Problem (14) by $\Theta(\cdot)$ and the approximated value of the objective function in Problem (14) (by the calculation of Algorithm 3) by $\Theta'(\cdot)$, given the observation assignment. Then,

$$\Theta(\Gamma'_g) \overset{(a)}{>} \Theta'(\Gamma'_g) \overset{(b)}{\geq} \frac{1}{2}\Theta'(\Gamma') \overset{(c)}{\geq} \frac{1}{2}\Theta'(\Gamma) \overset{(d)}{\geq} \frac{1}{2}(1-\epsilon)\Theta(\Gamma).$$

where (a) is by the rounding assumption, (b) is by Theorem 4.3, (c) is by the definition of Γ', and (d) is by Lemma 4.4. We denote the optimal solution to Problem (B) by OPT and the solution to Problem (B) output by Algorithm 2 together with Algorithm 3 by APX', respectively. Then, $OPT = \gamma + \Theta(\Gamma)$ and $APX' \geq \gamma + 1/2(1-\epsilon)\Theta(\Gamma)$ following a similar argument in the proof of Theorem 4.3. Hence, we always have $APX'/OPT > 1/2(1-\epsilon)$.

On the other hand, the complexity is dominated by that of Algorithm 3, which is $O(\frac{N^5}{\log^2 \frac{1}{1-\epsilon}})$ as shown in Lemma 4.4.

To check the feasible gap, we denote the set of observations assigned to $O = 0$ in Γ'_g by Δ. By Line 11 in Algorithm 2, we have $\sum_{\boldsymbol{y} \in \Delta} H'(\boldsymbol{y}) \leq \gamma - \alpha$. Also by Lemma 4.4, $\sum_{\boldsymbol{y} \in \Delta} H'(\boldsymbol{y}) \geq (1-\epsilon) \sum_{\boldsymbol{y} \in \Delta} H(\boldsymbol{y})$ holds. Then we have $\sum_{\boldsymbol{y} \in \Delta} H(\boldsymbol{y}) \leq \frac{\gamma-\alpha}{1-\epsilon}$. The PU throughput achieved can be represented by

$$\sum_{\boldsymbol{y} \notin \Delta} H(\boldsymbol{y}) \geq \gamma - \frac{\gamma - \alpha}{1-\epsilon} = \frac{\alpha}{1-\epsilon} - \gamma \frac{\epsilon}{1-\epsilon}.$$

The PU throughput fraction is then calculated as

$$\frac{\sum_{\boldsymbol{y} \notin \Delta} H(\boldsymbol{y})}{\gamma} \geq \frac{\delta}{1-\epsilon} - \frac{\epsilon}{1-\epsilon}.$$

∎

V. General Throughput Maximization Problem

In Problem (B), the SU-BS is free to choose any subset of S as the sensing set and maximizes the expected throughput of the system. However, a large number of SUs in the sensing set causes high overhead. In this section, we investigate the general problem with constraints of PU throughput above a threshold and the sensing set size below a threshold in Problem (A). To this end, we first show that the maximum throughput subject to PU throughput constraint in Problem (B) is monotonic over the number of SUs in the sensing set. By utilizing our solution to Problem (B), we then propose a greedy heuristic to Problem (A).

A. Monotonicity of Optimal System Throughput

Intuitively, sensing accuracy is increased by adding more SUs into the sensing set while guaranteeing the PU throughput above a threshold. In this section, we confirm this intuition and show that the optimal throughput for Problem (B) is monotonic over the number of SUs in the sensing set. We define

$$J^*(S_0) = \max_F \Big[(1-T_c)\pi_0 \sum_{\boldsymbol{y}:F(\boldsymbol{y})=0} P(\boldsymbol{Y}(S_0)=\boldsymbol{y}|B=0) + \gamma \sum_{\boldsymbol{y}:F(\boldsymbol{y})=1} P(\boldsymbol{Y}(S_0)=\boldsymbol{y}|B=1)\Big].$$

$$\text{s.t. } \gamma \sum_{\boldsymbol{y}:F(\boldsymbol{y})=1} P(\boldsymbol{Y}(S_0)=\boldsymbol{y}|B=1) \geq \alpha$$

Proposition 5.1: The optimal throughput for Problem (B) is monotonic over the SUs chosen in the sensing set; i.e., $J^*(S_0') \geq J^*(S_0)$, for all $S_0 \subseteq S_0'$.

Proof: Given the sensing set S_0', we design a decision rule as follows: we always ignore the observations made by SUs in $S_0' \setminus S_0$ and make the optimal decision based on observations of SUs in S_0. Then we have $J^*(S_0)$ as the system throughput with sensing set S_0' under this rule. Since $J^*(S_0')$ is the system throughput with sensing set S_0' under the optimal rule, we have $J^*(S_0') \geq J^*(S_0)$. ■

Using Proposition 5.1, we know that it is best to choose the full set as the sensing set if there is no sensing set constraint.

B. Subset Selection

We now investigate (Problem (A)) where the number of SUs in S_0 is constrained. It has been shown in [19] that no non-exhaustive search method in finding a feature subset of a given size k that has minimal Bayes risk always exists when observations are correlated. Due to the successful mapping between our problem and a Bayesian Decision problem (Problem (12)), the SU subset selection problem is equivalent to the feature subset problem in [19], except that the observations are assumed to be independent. The hardness of this problem has been a long standing open issue. It is not clear whether exhaustive search would be necessary as shown in [27] with independent observations, not to mention the general problem with the constraint of the PU throughput above a threshold as well. Although we characterize the monotonic property of maximum throughput over the number of SUs in the sensing set, the complexity of the problem is not clear. Many heuristics such as Sequential Forward Selection (SFS, [23]), Sequential Backward Selection (SBS, [23]) and their variations [9] have been proposed to solve problems of this type. We propose Algorithm 4 based on SFS [23] and Algorithm 2. In Algorithm 4, we start from an empty sensing set. At every step, only the SU that is not yet chosen and has the largest marginal increase on the maximum throughput is added to the set. The algorithm stops when the size of the set reaches k.

Algorithm 4 SFS Algorithm for Problem (C)

Output: sensing set S_0

- **if** $k \geq N$ **then**
  - $S_0 \leftarrow S$
- **else**
  - $R \leftarrow S$, $S_0 \leftarrow \emptyset$
  - **for** $i = 1$ to k **do**
    - $l^* \leftarrow \arg\max_{l \in R} \Big[J^*(S_0 \cup \{l\}) - J^*(S_0)\Big]$
    - $S_0 \leftarrow S_0 \cup \{l^*\}$
    - $R \leftarrow R \setminus \{l^*\}$

Note that the optimal system throughput with a given set is approximately calculated by Algorithm 2.

VI. Simulations

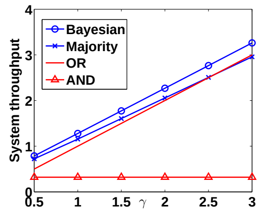
In this section, we study the throughput and analyze the sensitivity. We first compare the performance of the Bayesian decision rule (Algorithm 1), majority, AND and OR policies [31] in Section VI-A. Then we present the performance of the greedy algorithm for Problem (B) (Algorithm 2), the random selection and the optimal solution, and also compare the performance of Algorithm 4 with the optimal solution to Problem (A) in Section VI-A. Finally, we conduct sensitivity analyses with inaccurate P_m^i, P_f^i or π_0 information.

A. Simulation Setting

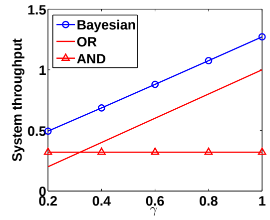
In all of the simulation studies, if not specifically mentioned, our model is that of a cognitive radio network with $N = 20$, $T_c = 0.2$, $\pi_0 = 0.4$, and $\gamma = 2$. We generate 100 groups of practical P_m^i and P_f^i based on randomly generated locations of SUs. In a 50×50 square area, the locations of the PU are randomly generated and fixed over the simulation. The power level P of the PU is also randomly generated between 1 and 10 and fixed then. In each of the 100 runs, we randomly generate the locations of N SUs within the area and calculate the distance $d(i)$ between the PU and SU i. We assume free-space path loss [2] and the SNR at SU i when the PU is transmitting is then calculated as $\frac{P/d(i)^2}{\theta(i)}$, where $\theta(i)$ is the normalized noise at SU i randomly generated between 0.01 and 0.1. The channel gain from the PU to SU i is denoted by $\frac{1}{d(i)^2}$. We let $\lambda(i)$ denote the threshold of the energy detector at SU i, which is randomly generated between 0 and 10. We use Equations (15) and (16) from [24] to generate 100 groups of P_m^i and P_f^i where the time bandwidth product $u = 3$:

$$\begin{aligned} P_m^i &= 1 - e^{-\frac{\lambda(i)}{2}} \sum_{n=0}^{u-2} \frac{1}{n!}\Big(\frac{\lambda(i)}{2}\Big)^n \\ &- \Big(\frac{1+SNR(i)}{SNR(i)}\Big)^{u-1} \times \Big[e^{-\frac{\lambda(i)}{2(1+SNR(i))}} \\ &- e^{-\frac{\lambda(i)}{2}} \sum_{n=0}^{u-2} \frac{1}{n!}\Big(\frac{\lambda(i)SNR(i)}{2(1+SNR(i))}\Big)^n\Big] \end{aligned} \quad (15)$$

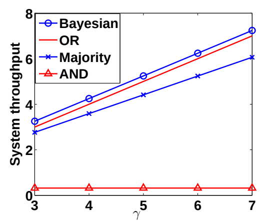
$$P_f^i = \frac{\Gamma(u, \frac{\lambda(i)}{2})}{\Gamma(u)} \quad (16)$$



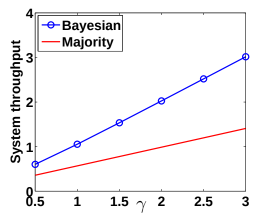
(a) All rules with varying γ.



(b) With lower γ.



(c) With higher γ.



(d) With time bandwidth product $u = 2$.

Fig. 3. Performance comparison of Bayesian decision rule, majority, AND and OR with $N = 20$, $T_c = 0.2$ and $\pi_0 = 0.4$.

In the equations above, $\Gamma(\cdot, \cdot)$ is the incomplete gamma function, and $\Gamma(\cdot)$ is the gamma function [24]. $SNR(i)$ is the SNR at SU i when the PU is transmitting.

B. Maximum System Throughput without Sensing Budget Constraint

We have shown in Section IV-A that Algorithm 1, the Bayesian decision rule based algorithm is optimal. In Figure 3, we compare its performance versus majority, AND, OR rules in terms of system throughput, which is the objective function value of Problem (C). When using the majority rule, the decision is 1 only when the majority of the SUs sense an active PU; for the AND rule, the decision is 1 only when all SUs sense an active PU; for the OR rule, the decision is 1 if any of the SUs senses an active PU. We vary γ, the average PU throughput in Figure 3. The Bayesian decision rule strictly outperforms the other algorithms. Among them, the OR and majority rules have similar performance and are both better than AND since the PU transmission is better protected by the OR rule. We show different scenarios where the Bayesian wins over other rules by a small gap in Figure 3(a) and by a significant gap in Figures 3(b) and 3(d). Also, we show the scenarios where OR is always better than AND in Figure 3(c) and AND performs better than OR when γ is low in Figure 3(b). Note that in Figure 3(a), the performance of majority is close to that of Bayesian while they are far apart in [12] with randomly generated P_m^i and P_f^i. We observe that $P_m^i + P_f^i \geq 1$ occurs often there while it never does in the practically generated P_m^i and P_f^i; majority rule over bad SUs ($P_m^i + P_f^i \geq 1$) leads to unwise decisions. In Figure 3(d) we reduce the time bandwidth product to 2 and regenerate 100 groups of P_m^i and P_f^i. We observe a significant number of SUs with $P_m^i + P_f^i \geq 1$ in each group, which leads to the big gap between Bayesian and majority compared to Figure 3(a) when $u = 3$.

C. Maximum System Throughput with General Constraints

As shown in Section IV, greedy algorithm (Algorithm 2) can achieve throughput strictly greater than $1/2$ of the optimal throughput in Problem (B). We compare it with its counterpart using the OR rule (Greedy_OR) and random selection. Greedy_OR initially assigns observations to $Z = 0$ or $Z = 1$ by OR rule (only observation $\mathbf{0}$ is assigned to $Z = 0$ in this step); if feasibility is not met, it moves observation $\mathbf{0}$ to $Z = 1$ as the last chance to satisfy feasibility; if feasibility after the initial step (only observation $\mathbf{0}$ in $Z = 0$) is met, observations are sorted in $Z = 1$ in decreasing order of $\frac{G(\boldsymbol{y})}{H(\boldsymbol{y})}$ and moved to $Z = 0$ until feasibility is violated. Random selection is based on Bayesian decision rule, which means Algorithm 1 is first executed; after that, observations with $G(\boldsymbol{y}) \geq H(\boldsymbol{y})$ are randomly selected to put in $Z = 1$ until the feasibility is satisfied. Thus, the main difference between greedy algorithm and random selection lies in the selection criterion of observations with $G(\boldsymbol{y}) \geq H(\boldsymbol{y})$ after the initial assignment based on Bayesian decision rule.

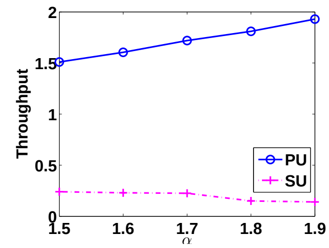
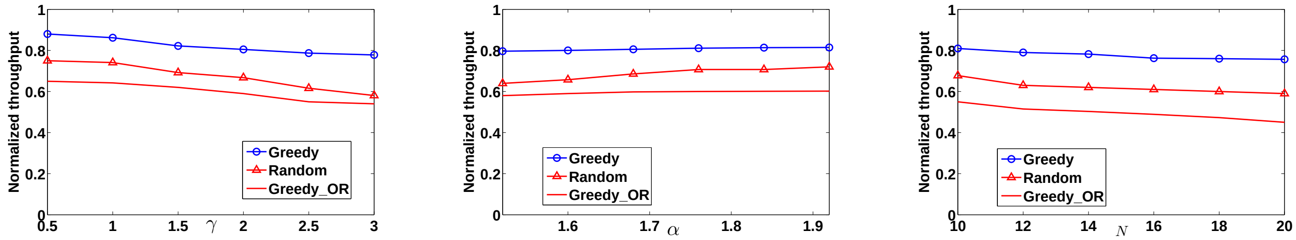


Fig. 5. Performance of PU and SU throughput over α.

In addition, we set $r = 2$. We vary parameters such as γ, the average PU throughput, α, the PU throughput constraint, and N, the number of SUs in Figure 4. *Normalized throughput* is defined as the system throughput under the algorithm over the optimal solution. Two boundary cases are excluded in the result presentation where both the greedy algorithm and random selection will give the optimal solution: 1) Bayesian decision rule gives the optimal solution; 2) It is optimal to put all observations in $Z = 1$. Hence, we only show their performance when at least one but not all observations with $G(\boldsymbol{y}) \geq H(\boldsymbol{y})$ have to be moved to $Z = 1$.

In Figure 4(a), the normalized throughput of the greedy algorithm, random selection and Greedy_OR are compared for different values of γ, the average PU throughput in the system. We set α to be 0.8γ for a fair comparison. With a higher γ, the factor decreases gradually for all three algorithms. The greedy algorithm, which has a provable lower bound, outperforms the other two algorithms. Potentially, the Bayesian decision rule assigns more SUs to $Z = 1$ compared to a lower γ case. Thus, the initial assignment is closer to α, the PU throughput constraint. Since we only consider cases where Bayesian decision rule is not optimal, all algorithms tend to have worse performance when the initial assignment approaches α because they get more sensitive to wrong observation selections. Random selection wins over Greedy_OR in that the decision rule still plays an important role in the constrained problem.

In Figure 4(b), we vary α, the minimum PU throughput constraint, and compare the performance of the algorithms.



(a) With different γ ($N = 20$, $\alpha = 0.8\gamma$). (b) With different α ($N = 20$, $\gamma = 2$). (c) With different N ($\gamma = 2$, $\alpha = 1.6$).

Fig. 4. Performance comparison of greedy algorithm and random selection when $T_c = 0.2$, $\pi_0 = 0.4$ and $r = 2$.

Again, the greedy algorithm outperforms the other two. The normalized throughput increases with α, although it is a minor increase in the two greedy algorithms. The increase can be explained similarly as in Figure 4(a): a higher α makes the initial assignment farther away from it so that the performance is less sensitive to the choice of observations.

In Figure 4(c), we test the performance of our greedy algorithm by varying the number of SUs from 10 to 20. The normalized throughput of all algorithms degrades with more SUs while the actual system throughput increases. However, it is always far above $1/2$ for the greedy algorithm, as proved in Theorem 4.3. Greedy_OR drops below $1/2$ when N is large as shown in the figure.

We also vary the PU throughput threshold α and show the corresponding PU and SU throughput achieved in Figure 5. Algorithm 2 always leads to a feasible solution as shown in the PU throughput while the SU throughput degrades as the threshold increases since the more strict requirement on PU throughput makes the decision biased toward achieving PU throughput.

As stated in Section V, the hardness of Problem (A) is unknown. Therefore, here, we focus on the performance of Algorithm 4, heuristic we proposed, and compare it with random selection. In this random selection, SUs are randomly selected for sensing. The selection is repeated till the sensing set size is no more than k. Then the PU throughput achieved is calculated based on Algorithm 1. This process is repeated till the PU throughput is above the prescribed threshold. In this way, the time complexity of the random selection algorithm is not low. In Figure 6, we vary k, the size of the sensing set, from 1 to N and show the normalized throughput of Algorithm 4 and random selection. When k increases, the performance of Algorithm 4 degrades and the performance of random selection increases. However, the former is always better than the latter and the gap is especially large when k is small. Algorithm 4 on average achieves at least 0.9 of the optimal solution achieved by exhaustive search in our simulation.

D. Sensitivity Analysis

So far, we have assumed that parameters such as P_m^i and P_f^i are accurate. However, they are collected using empirical data. Hence, the actual values could be different from those used in the calculations. We investigate the sensitivity of system throughput to these errors for Problems (A) in Figures 7 and 8. We define *Efficiency* as the throughput with inaccurate parameters over the throughput with accurate parameters. For sensitivity analysis to P_m^i and P_f^i, the value of P_m^i used falls in the range of $[P_m^i - \epsilon, P_m^i + \epsilon]$ where P_m^i is the actual value; similarly, the value of P_f^i used falls in the range of $[P_f^i - \epsilon, P_f^i + \epsilon]$ where P_f^i is the actual value. The efficiency is more than 97% even when the error range ϵ reaches 0.1. So Algorithm 4 is not sensitive to sensing inaccuracies.

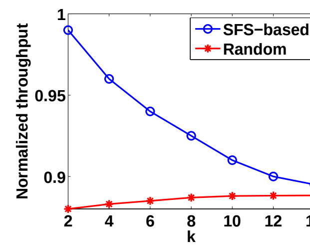


Fig. 6. Performance comparison of Algorithm 4 over different numbers of SUs in the sensing set when $N = 20$, $T_c = 0.2$, $\pi_0 = 0.4$, $\gamma = 2$ and $\alpha = 1.6$.

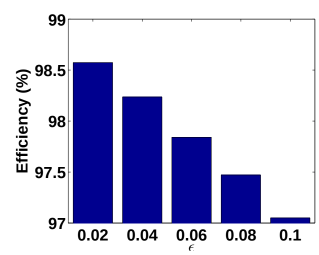


Fig. 7. Sensitivity to P_m^i, P_f^i when $N = 20$, $T_c = 0.2$, $\pi_0 = 0.4$, $\gamma = 2$, $r = 2$ and $\alpha = 1.6$.

The sensitivity analysis of system throughput to π_0 is more optimistic as in Figure 8. The value of π_0 used falls in the range of $[\pi_0 - \epsilon, \pi_0 + \epsilon]$ where π_0 is the actual value. The efficiency is always greater than 99% when $\epsilon \leq 0.2$. In both figures, we have feasible solutions even with inaccurate information in all 100 samples. These results suggest that our solution is robust to inaccuracies in P_m^i, P_f^i or π_0.

VII. CONCLUSION

In this paper, we investigated a general problem for maximizing the system throughput using cooperative sensing in cognitive radio networks. To solve it, we formulated a sensing decision problem of maximizing the system throughput. The first problem we considered is the unconstrained problem of maximizing the weighted sum of the PU and SU throughput in the cognitive radio system. We developed a Bayesian rule based algorithm to find the optimal decision. To guarantee a

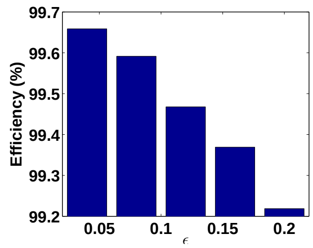


Fig. 8. Sensitivity to π_0 when $N = 20$, $T_c = 0.2$, $\pi_0 = 0.4$, $\gamma = 2$, $r = 2$ and $\alpha = 1.6$.

minimum PU throughput, we then studied a system throughput maximization problem with PU throughput constraint. We proved that the new problem is strongly NP-hard, and proposed a greedy algorithm that achieves an approximation factor strictly greater than $\frac{1}{2}(1-\epsilon)$ with the time complexity $O(\frac{N^5}{\log^2 \frac{1}{1-\epsilon}})$ where N is the total number of SUs. We also characterized the feasibility gap which tends to 0 when ϵ goes to 0. Finally, we studied the general problem with both PU throughput and sensing set size constraints. We established the monotonicity of the system throughput function, and proposed a simple greedy heuristic that performs well in the numerical results. However, proving a performance guarantee for it remains elusive, which is one of our future research directions.

References

[1] R. Balakrishnan I. F. Akyildiz, B. F. Lo. Cooperative spectrum sensing in cognitive radio networks: A survey. *Physical Communication*, March 2011.

[2] Constantine A. Balanis. *Antenna Theory: Analysis and Design*. Wiley-Interscience, 2005.

[3] J. O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, New York, 1985.

[4] T. Do and B.L. Mark. Joint spatial-temporal spectrum sensing for cognitive radio networks. In *proc. of CISS*, pages 124 –129, March 2009.

[5] R. Fan and H. Jiang. Optimal multi-channel cooperative sensing in cognitive radio networks. *IEEE Transactions on Wireless Communications*, 9(3):1128–1138, 2010.

[6] Federal Communications Commission. Notice of proposed rulemaking, in the matter of unlicensed operation in the tv broadcast bands (docket no. 04-186) and additional spectrum for unlicensed devices below 900 mhzand in the 3 ghz band (02-380), fcc 04-113. May 2004.

[7] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.

[8] http://grouper.ieee.org/groups/802/22/. Ieee 802.22, working group on wireless regional area networks (wran).

[9] M. Kudo. Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition*, 33(1):25–41, Jan 2000.

[10] W. Lee. Optimal spectrum sensing framework for cognitive radio networks. *IEEE Transactions on Wireless Communications*, 7(10):3845–3857, 2008.

[11] S. Li, Z. Zheng, E. Ekici, and N. Shroff. Technical report. Maximizing System Throughput by Cooperative Sensing in Cognitive Radio Networks, 2011. http://www.cse.ohio-state.edu/∼lish/Shuang_Li_TR_2011.pdf

[12] S. Li, Z. Zheng, E. Ekici, and N. Shroff. Maximizing System Throughput by Cooperative Sensing in Cognitive Radio Networks. In *proc. of Infocom*, Orlando, Florida, 2012.

[13] Y.G. Li and G. Ganesan. Cooperative spectrum sensing in cognitive radio networks. In *proc. of DySPAN*, pages 137 – 143, Nov. 2005.

[14] B. Mark and A. Nasif. Estimation of maximum interference-free power level for opportunistic spectrum access. *Wireless Communications, IEEE Transactions on*, 8(5):2505 –2513, May 2009.

[15] A. Min and K. Shin. An optimal sensing framework based on spatial rss-profile in cognitive radio networks. In *proc. of SECON*, pages 207–215, Piscataway, NJ, USA, 2009. IEEE Press.

[16] S. Mishra, A.t Sahai, and R. W. Brodersen. Cooperative sensing among cognitive radios. In *proc. of ICC*, pages 1658–1663, 2006.

[17] C. T. Ng, M.S.Barketau, T.C.E. Cheng and M. Y. Kovalyov. "Product Partition" and related problems of scheduling and systems reliability: Computational complexity and approximation. *European Journal of Operational Research*, 207(2):601–604, Dec 2010.

[18] E.C.Y. Peh, Y. Liang, Y. Guan, and Y. Zeng. Optimization of cooperative sensing in cognitive radio networks: A sensing-throughput tradeoff view. *Vehicular Technology, IEEE Transactions on*, 58(9):5294 –5299, Nov. 2009.

[19] J. M. Pena and R. Nilsson. On the complexity of discrete feature selection for optimal classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:1517–1522, 2010.

[20] W. Saad, Zhu Han, M. Debbah, A. Hjorungne, and T. Basar. Coalitional Games for Distributed Collaborative Spectrum Sensing in Cognitive Radio Networks. In *proc. of INFOCOM*, pages 2114 –2122, April 2009.

[21] F.W. Seelig. A description of the august 2006 xg demonstrations at fort a.p. hill. In *proc. of DySPAN*, pages 1 –12, April 2007.

[22] M. Shahid and J. Kamruzzaman. Weighted soft decision for cooperative sensing in cognitive radio networks. In *proc. of ICON*, page 6, 2010.

[23] S. D. Stearns. On selecting features for pattern classifiers. In *proc. of ICPR*, pages 71–75, Coronado, CA, 1976.

[24] Chunhua Sun, Wei Zhang, and Khaled Ben Letaief. Cluster-based cooperative spectrum sensing in cognitive radio systems. In *poc. of ICC*, pages 2511–2515, 2007.

[25] J. Unnikrishnan and V.V. Veeravalli. Cooperative sensing for primary detection in cognitive radio. *Selected Topics in Signal Processing, IEEE Journal of*, 2(1):18 –27, Feb. 2008.

[26] H. Urkowitz. Energy detection of unknown deterministic signals. In *proc. of the IEEE*, 55(4):523 – 531, April 1967.

[27] J. M. Van Campenhout. *On the problem of measurement selection*. PhD thesis, Stanford University, 1978.

[28] V. V. Vazirani. *Approximation Algorithms*. Springer, Mar 2004.

[29] E. Visotsky, S. Kuffner, and R. Peterson. On collaborative detection of tv transmissions in support of dynamic spectrum sharing. In *proc. of DySPAN*, pages 338 –345, Nov. 2005.

[30] D. Xue, E. Ekici and M. Vuran. (corn)2: Correlation-based cooperative spectrum sensing in cognitive radio networks. To appear in *proc. of WiOpt*, Paderborn, Germany, 2012.

[31] Wei Zhang, R.K. Mallik, and K. Ben Letaief. Cooperative spectrum sensing optimization in cognitive radio networks. In *proc. of ICC*, pages 3411 –3415, May 2008.