Media Access with Spatial Reuse for Cooperative Spectrum Sensing

Xiao Shao

Department of Electrical & Computer Engineering
McGill University
Montreal, Canada

November 2015

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Master of Engineering.

Abstract

Cognitive Radio (CR) is an important trend for a solution to the stringent requirements and scarcity of radio spectrum resources. Spectrum sensing is a vital function in a CR system, which is necessary for a more flexible and efficient usage of the radio spectrum. The utilization of several cooperating sensors can overcome multipath fading and shadowing effects, and increase the reliability of primary signal detection in spectrum sensing. In this work, we consider a system model of a dedicated detect-and-forward wireless sensor network (DetF WSN) for cooperative spectrum sensing. We analyse the degradation introduced by reporting channel errors on cooperative spectrum sensing performance with the *k-out-of-n* decision fusion rule. Moreover, a proper media access control (MAC) protocol is required to resolve conflicts and conserve resources for the information exchange between sensors in such a system. The influence of the MAC protocol on spectrum sensing performance of the WSN is a key consideration in this work. We focus on designing a spatial reuse MAC protocol based on TDMA/OFDMA for the intra-WSN communication in a bandwidth-limited system. Two design approaches, using a greedy and an adaptive simulated annealing (ASA) algorithm, are illustrated in detail. Moreover, performance numerical results for a specific grid network in a Rician fading environment are presented.

Sommaire

La Radio Cognitive (CR) est une tendence importante pour une solution adaptée aux exigences contraignantes et à la rareté des ressources du spectre radioélectrique. La détection du spectre est une fonction vitale dans un système CR, ce qui est nécessaire pour un usage plus flexible et efficace du spectre radioélectrique. L'utilisation de plusieurs capteurs coopérants peut résoudre les problèmes comme les évanouissements par trajets multiples et l'effet d'ombre et elle peut aussi augmenter la fiabilité de la détection des signaux primaires dans la détection du spectre. Dans cet article, on considère un modèle de système d'un réseau sans fil, détecter-et-transmettre (DeF WSN), dédié à la coopérative détection du spectre. On analyse la dégradation introduit en rendant compte d'erreurs sur la performance de la coopérative détection du spectre avec le règle de la décision fusion *k-out-of-n*. D'ailleurs, un protocole de control d'accès des médias (MAC) est exigé afin de résoudre les conflits et conserver les ressources pour l'échange d'information entre capteurs dans un tel système.L'influence de MAC protocole sur la performance de la détection du spectre du WSN est une considération essentielle dans ce projet. On se concentre sur la conception d'un MAC protocole de réutilisation spatiale qui est fondé sur TDMA/OFDMA pour la communication intra-WSN dans un système à bande passante limitée. Deux approaches de conception, utilisant un algorithme glouton et un algorithme de recuit simulé qui est adaptif (ASA), sont illustrés en detail. En plus, les résultats de la performance numéricale pour un réseau maillé spécifique dans un environnement d'évanouissements à distribution de Rician sont présentés.

Acknowledgments

First and foremost, I would like to thank my supervisor, Professor Harry Leib, for his instruction, patience, and encouragement throughout my studies and research. This work would not be complete without his guidance and tolerance. Meanwhile, I am also grateful for Professor Harry Leib for his financial support from the BlackBerry and NSERC grants.

Secondly, I would like to send my gratitude to my colleagues and friends, both in and outside of our Telecommunication & Signal Processing Lab, for their technical and emotional support during my study and research at McGill University. In particular, I am grateful to Yi Wang, Xiaofei Shao, Guanglei Dai, Tianpei Chen, Scott Monk, and Djelili Radji.

Last but not least, I would like to express my deep gratitude to my parents for their endless support towards my graduate studies at McGill University.

Contents

List of Figures

List of Tables

List of Acronyms

CR	Cognitive Radio
SU	Secondary User
PU	Primary User
WRAN	Wireless Regional Area Network
MAC	Media Access Control
LTE	Long-Term Evolution
WSN	Wireless Sensor Network
DefF	Detect-and-Forward
ASA	Adaptive Simulated Annealing
PDF	Probability Density Function
SNR	Signal-to-Noise Ratio
SINR	Signal to Interference plus Noise Ratio
BEP	Bit Error Probability
AWGN	Additive White Gaussian Noise
BSC	Binary Symmetric Channel
BPSK	Binary Phase Shift Keying
CDF	Cumulative Distributed Function
ROC	Receiver Operating Characteristic
TDMA	Time Division Multiple Access
OFDMA	Orthogonal Frequency-Division Multiple Access
T-F	Time-Frequency

Chapter 1

Introduction

With the rapid development of wireless techniques and growing number of innovative telecommunication services, the scarcity of spectrum resources has become a critical issue in wireless communications. Within the conventional spectrum management framework, all of the frequency bands are exclusively allocated to licensed users, and no violation from unlicensed devices is allowed. Such a static spectrum allocation policy protects licensed users from any intersystem interference. However, it caused a bottleneck for efficient utilization of radio frequency bands, resulting in a largely under-utilized licensed spectrum [1, p. 2]. For instance, spectrum occupancy measurement conducted in New York City has indicated that the maximum spectrum occupancy is only 13.1% in the range between 30MHz to 3GHz [2]. Another measurement undertaken in downtown Washington D.C. has shown an occupancy of less than 35% of the spectrum below 3GHz [2]. In addition, these measurement studies also show that the spectrum utilization varies significantly in time periods, frequency ranges, as well as geographical locations.

To address the problem of spectrum scarcity and exploit its under-utilization, Cognitive Radio (CR) technology arises to be a feasible solution via dynamic and opportunistic spectrum access technologies. In a CR system, spectrum utilization is improved by allowing a secondary user (SU) to access spectral white spaces without introducing harmful interference to the primary user (PU) [3]. In [4], CR is defined as an intelligent wireless communication system which can learn from its surrounding environment, and adapt certain operating parameters to provide highly reliable communications and realize efficient utilization of the radio spectrum.

A typical example of CR's applications is the opportunistic use of the white spaces in the television (TV) bands in the IEEE 802.22 for wireless regional area network (WRAN) [5]. IEEE 802.22 WRAN is the first worldwide standard defining the wireless air interface based on CR techniques in physical (PHY) and media access control (MAC) layers [6]. It allows sharing of geographically unused spectrum allocated to the TV broadcast bands on a non-interfering basis. Another example is research on the coexistence of cellular technologies and wireless local area networks in the same unlicensed bands. The authors of [7, 8] discussed the possible mechanisms for the concurrent operation of LTE and Wi-Fi sharing unlicensed bands through dynamic spectrum access techniques.

An important function for such dynamic spectrum sharing is the so-termed spectrum sensing, which identifies the unused portion of the spectrum in a certain geographical area at a certain period of time. Some well-known spectrum sensing techniques are based on *energy detection* [9], *matched filter detection* [10], *cyclostationary detection* [11], *wavelet detection* [12], and *covariance detection* [13].

Single-user's sensing performance is often compromised by multipath fading, shadowing, and the uncertainty of the device's noise [14]. When encountering the hidden primary user problem [15], a SU may fail to sense the presence of the PU operating in the vicinity, and cause severe interference to the PU system by accessing the licensed band. The sensing performance, however, can be significantly improved by using multiple sensing nodes, a scheme termed *Cooperative Spectrum Sensing* [16–18]. Local spectrum measurement is conducted at each cooperating nodes, and a final decision about the presence of the PU is made after fusing the information provided by sensing. Cooperative spectrum sensing provides better spatial coverage as well as multi-user diversity gains to enhance the detection reliability, and it also helps lower the sensitivity requirements of single detectors [19].

A simple and intuitive implementation method of cooperative spectrum sensing is to integrate sensing function into SU terminals, such as personal cellphones. This approach may cause several limitations [20, 21]. Firstly, a sufficient number of widely distributed SUs is required to obtain accurate detection results, which SU terminals cannot guarantee in most cases. Secondly, the SU device cannot sense the spectrum when it is transmitting data, or transmit data in the sensing period. Thus, incorporation of a spectrum sensing mechanism in a SU device will degrade the data transport efficiency, which is called *lost transmit opportunity cost* and explained in [21]. Moreover, the cost, complexity and power consumption of an SU device are increased when the sensing function is added. An alternative approach is *External Sensing*, which relies on a dedicated wireless sensor network (WSN) to perform spectrum sensing. This dedicated WSN can be deployed by the service provider of the SUs, and an exchange of information between the SU and WSN can aid a SU to access the licensed spectrum. Such a WSN can provide sufficient diversity to cope with the hidden primary user problem, and guarantees robustness against model uncertainty induced by fading and path loss [22]. Furthermore, sensors neither need to be mobile nor battery-powered, and the cost, complexity and power consumption of SU devices decrease.

Therefore, in this work, we consider a Detect-and-Forward (DetF) distributed WSN for cooperative spectrum sensing. It is a fully distributed cooperative spectrum sensing system, whose operation doesn't rely on centralized control and a separate decision fusion center, which are necessary in a centralized system [23]. Each sensor chooses its cooperating partners independently, collects the local spectrum sensing results from them, and then makes its own spectrum sensing decision. A SU can receive the spectrum sensing decision directly from any nearby sensor in this WSN. The WSN in such a distributed form has better flexibility and scalability than a centralized system. For example, in a centralized WSN, when adding more sensors in the network to cover a larger area, we have to consider relaying or add more fusion centres to collect the spectrum sensing results, and computation and storing capability of the fusion centre is also affected.

To perform cooperative spectrum sensing, sensors in the DetF WSN need to exchange their local detection results over a common wireless channel, generating non-negligible traffic. Thus, a proper media access control protocol is required for the dedicated WSN, whose primary role is coordinating transmissions so as to efficiently utilize the resources and resolve contention. Compared with conventional MAC protocols, the peculiarities of WSNs, including the large number of nodes, low data load, and volatile links, requires paradigm shifts in MAC design [24]. The most significant design constraint is usually the limited energy budget of a sensor node together with the requirement of longevity of the network. Therefore, in [24–26], energy efficiency is the primary design consideration when discussing the MAC concepts in relation to WSNs. However, in our DetF WSN, sensors are assumed to be fixed in the region and do not depend on limited battery power. Therefore, spectrum efficiency and interference, instead of energy efficiency, become the primary issues. Mobility needs no consideration, however, low system complexity is expected. Another special part of our MAC design of this DetF WSN is its performance as a cooperative spectrum sensing system.

The original contributions in this thesis are summarized as follows:

- We propose a spatial reuse MAC protocol for exchanging sensing results within the WSN to exclude primary conflict and save bandwidth resource. Regarding the design approach, we propose and interpret a greedy algorithm as well as an adaptive simulated annealing (ASA) algorithm.

- We analyse the degradation introduced by the reporting channel errors on cooperative spectrum sensing performance with the *k-out-of-n* decision fusion rule.

- We present numerical results to analyse the spectrum sensing performance of the DetF WSN with spatial reuse MAC protocol in a grid network form.

The rest of this thesis is organized as follows. Chapter 2 presents the network architecture, channel model, and system setup assumptions. The performance of cooperative spectrum sensing using k-out-of-n decision fusion rule with imperfect reporting channels is considered in Chapter 3. The design of the spatial reuse MAC protocol via greedy algorithm as well as ASA algorithm are presented in Chapter 4. Performance numerical results for a grid network are presented in Chapter 5, and the conclusions are drawn in Chapter 6.

Chapter 2

System Model of the DetF Distributed WSN

2.1 Network Architecture

Fig. 2.1 shows the architecture of the distributed DetF WSN considered in this work. Sensors are equipped with spectrum sensing as well as communication capability. The entire operation of the WSN can be divided into two stages: 1) Intra-WSN Sensing and 2) WSN-SU Handshaking. The distributed DefF WSN proposed in this work operates without a separate fusion center, and thus each sensor is responsible to make its own detection decision. It makes its local spectrum measurement, selects the cooperating partners, gathers the local detection results from its partners, and finally makes its own spectrum sensing decision. Each sensor can deliver the spectrum sensing decision to the SU independently as requested.

The operation of the first stage is summarized as follows:

a) *Measurement Phase*: Each sensor measures the received signal through the frequency channel of interest.

b) *Local Decision Phase*: Each sensor makes a decision based on its measurement via a local spectrum sensing scheme. This local decision may be *hard* (e.g. a 1-bit decision through energy detection) or *soft* (e.g. a quantized version of log-likelihood ratio) [27].

c) *Communication Phase*: Each sensor communicates with its cooperating sensors, which are determined by a certain *partner selection* scheme, through dedicated reporting channels to exchange local decisions.

d) *Final Decision Phase*: A final decision is made by each sensor through combining the decisions gathered, according to a specific *decision fusion* rule.

e) The four phases above are repeated for another channel of interest.

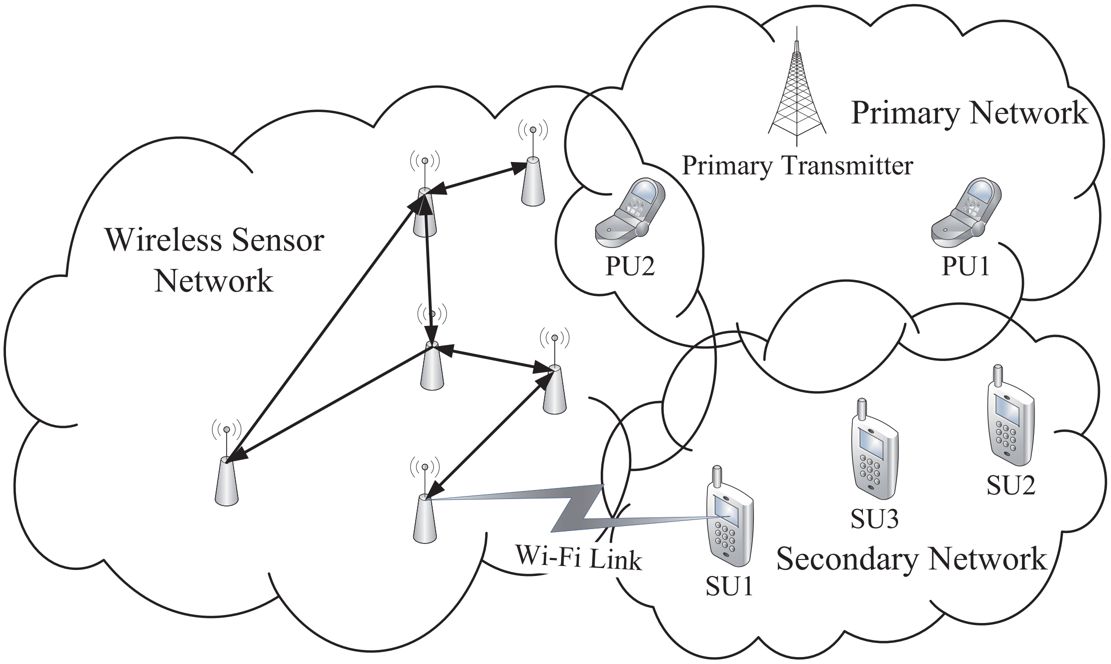

Fig. 2.1 The architecture of the WSN which is dedicated for cooperative spectrum sensing.

The cooperation between sensors can be specified by a *directed graph* or an *adjacency matrix*. An example of the directed graph representation is shown in Fig. 2.2, where the circles represent the sensors, and the directed lines represent the cooperation relations. Take s_2 in Fig. 2.2 for example. There are two incident directed link, $s_1 \to s_2$ and $s_2 \to s_2$, which means when doing decision fusion, s_2 combines the local decision received from s_1 and its own local decision. $s_2 \to s_1$ and $s_2 \to s_3$ represent that s_2 provides its local sensing result to s_1 and s_3 for their decision fusion.

The corresponding *adjacency matrix* **R** of this *directed graph* is

$$\mathbf{R} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \tag{2.1}$$

where the element $R_{i,j} = 1 (i \neq j)$ represents the existence of a cooperative relation $\mathbf{r}_{i,j}$, from the *partner* s_i to *recipient* s_j. Otherwise, $R_{i,j} = 0$. It is worth noting that if $i = j$, $R_{i,j} = 1$ means that this sensor employs its local sensing result during decision fusion, and s_i is also called a *partner* or *recipient* of itself. Thus, there is still a directed line starting and ending at this sensor. If s_i does not employ its local sensing result and only combines the sensing results from other *partners*, we will have $R_{i,i} = 0$. For example, in order to find the *partners* of s_2, look at the second column of **R**. There are two 1's, $R_{1,2} = 1$ and $R_{2,2} = 1$, and thus the two *partners* of s_2 are s_1 and s_2. If we look at the second row, we will find all the *recipients* of s_2 by picking all the 1's, which are $R_{2,1} = R_{2,2} = R_{2,3}$. So the three *recipients* of s_2 are s_1, s_2, and s_3.

In the second stage, a handshaking process takes place between the WSN and SUs to inform the sensing results whether the channel is free or not (through a Wi-Fi link for example). The second stage procedure is a topic beyond the scope of this thesis, and we only consider the first stage in this work.

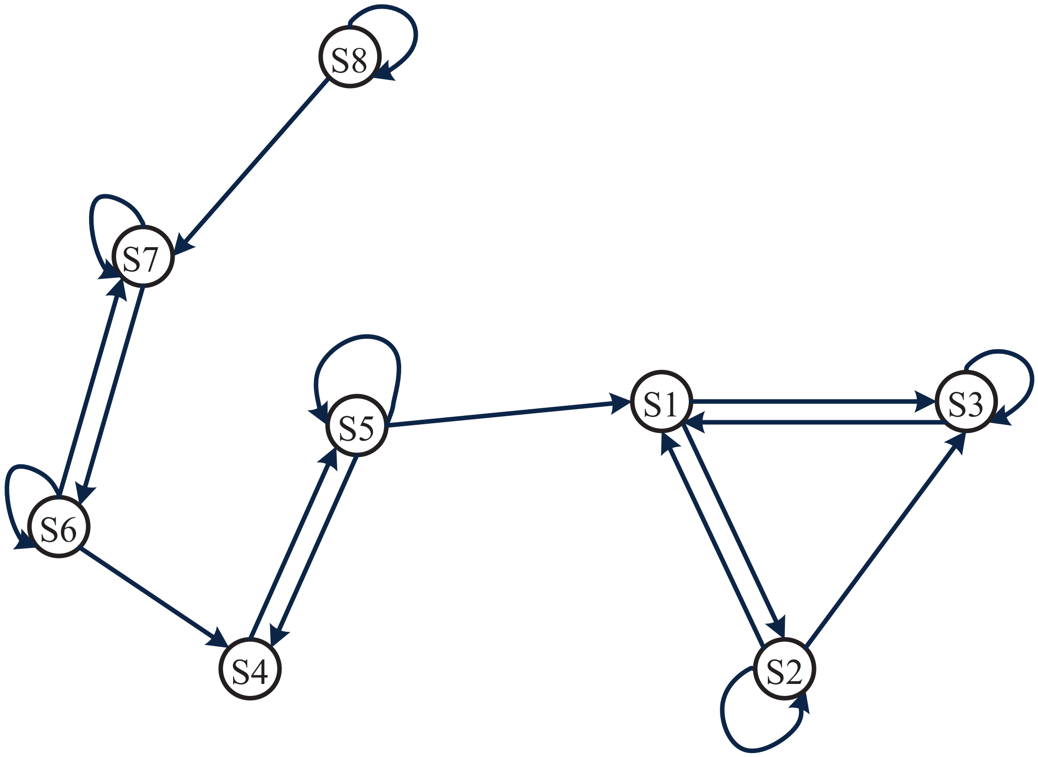

Fig. 2.2 Directed graph representation of cooperation between sensors.

2.2 Channel Model

To be more close to the realistic wireless propagation environment, we consider a time-invariant channel model including the effects of *path loss*, *multipath fading* and *additive* *interference*. Moreover, all wireless channels, including the reporting channels between sensors, as well as the detecting channels between the primary transmitter and sensors, follow the same model.

2.2.1 Path-loss Model

We use the *Simplified Path-loss Model* proposed in [28, p. 47] to characterize the variation in the received signal power with distance due to dissipation of the power radiated by the transmitter as well as effects of the propagation channel. Compared with the empirical path-loss models, such as *Okumura Model*, *Hata Model*, and *Piecewise Linear Model* [28, pp. 42–46], this *Simplified Path-loss Model* is more suitable for general trade-off analysis of system design:

$$P_L = \frac{P_T}{P_R} = \frac{1}{K}\left(\frac{d}{d_0}\right)^{\mu} \tag{2.2}$$

where the path loss P_L is defined as the ratio of the transmit power P_T over the received power P_R, K is a unitless constant which depends on the antenna characteristics and the average channel attenuation, d is the distance between the transmitter and receiver, d_0 is a reference distance for the antenna far field, and μ is the path-loss exponent. Table 2.1 gives a summary of typical μ-values for different environments [28, p. 47].

Replacing ${Kd_0}^{\mu}$ with A, we obtain an equivalent exponential form of path-loss model described by (2.2) is

$$P_L = \frac{P_T}{P_R} = \frac{1}{A}d^{\mu} \tag{2.3}$$

When d_0 is set as the unit length, $A = K$. It is worth noting that the simplified path-loss model (2.2) or (2.3) is valid only when the transmission distance d is larger than a threshold due to scattering phenomena in the antenna near field.

Table 2.1 Typical Path-Loss Exponent Values.

Environment	μ range
Urban macrocells	3.7 - 6.5
Urban microcelss	2.7 - 3.5
Office building (same floor)	1.6 -3.5
Office building (multiple floors)	2 - 6
Store	1.8 - 2.2
Factory	1.6 - 3.3
Home	3

2.2.2 Multipath Fading Model

In this thesis, we use Rician fading for each communication link [29, p. 23]. The probability density function (PDF) of the instantaneous signal-to-noise ratio (SNR) γ over a Rician fading channel is

$$f_{\mathrm{Ric}}(\gamma)=\frac{K+1}{\bar{\gamma}}\exp\left[-K-\frac{(K+1)\gamma}{\bar{\gamma}}\right]I_0\left(2\sqrt{\frac{K(K+1)\gamma}{\bar{\gamma}}}\right) \tag{2.4}$$

where $\gamma \geqslant 0$, K is the Rician fading parameter, and $\bar{\gamma}$ is the average received SNR. When combining the effects of path loss and multipath fading, $\bar{\gamma}$ equals to the received SNR based on path loss (2.3) alone [28, p.77], i.e. $\bar{\gamma} = \gamma_T A d^{-\mu}$, where γ_T represents the transmit SNR. $I_0(\cdot)$ is the 0th order modified Bessel function of the first kind,

$$I_0(x) = \frac{1}{\pi}\int_0^{\pi} \mathrm{e}^{x\cos\theta} d\theta = \sum_{m=0}^{\infty} \frac{x^{2m}}{2^{2m}\cdot(m!)^2}. \tag{2.5}$$

Besides Rician fading, our analysis procedure can also be effectively applied in other typical fading models, e.g. Nakagami, when (2.4) is replaced by the corresponding PDF. In addition, we ignore the impact of shadowing on the operation of the DetF WSN in this work, since it is assumed that the sensor positions were optimized to make this effect insignificant.

2.2.3 Interference Model

We assume an *Additive Interference Model*, in which a wireless communication link treats all the other on-going transmissions on the same channel as noise. In [30], this interference model is compared with the *Capture Threshold Model*, *Protocol Model*, and *Interference Range Model*, and showed to be more accurate in modelling the cumulative interference, which is significant in wireless network design and assessment. Let $G_{i,j}$ be the channel gain from s_i to s_j, $P_T^{(i)}$ the transmit power of s_i, and P_N the thermal noise power in the frequency of operation. The received Signal to Interference plus Noise Ratio (SINR) of the transmission from s_i to s_j is

$$\gamma_{i,j} = \frac{G_{i,j} P_T^{(i)}}{P_N + \sum_{k \in \mathcal{K} \backslash \{i\}} G_{k,j} P_T^{(k)}} \tag{2.6}$$

where $\mathcal{K} \backslash \{i\}$ denotes the set of sensors transmitting simultaneously on the same channel other than s_i. The SINR at the receiver determines the received bit error probability (BEP), which in turn affects the success or failure probability of decoding with certain modulation and coding schemes.

Combined with the path-loss model introduced before, (2.6) can be written as

$$\gamma_{i,j} = \frac{A d_{i,j}^{-\mu} P_T^{(i)}}{P_N + \sum_{k \in \mathcal{K} \backslash \{i\}} A d_{k,j}^{-\mu} P_T^{(k)}} \tag{2.7}$$

2.3 Other System Setup Assumptions

In addition to the models and assumptions we have introduced in the previous sections, in this work for network design purposes, we also make the following assumptions:

- Sensors are static after deployed.

- Each sensor employs an omni-directional antenna.

- A partner selection scheme has been utilized to generate the *adjacency matrix* of cooperation.

- Each sensor is in direct transmission range of its partners. Thus, we only focus on next neighbour transmissions, and routing is not considered in this work.

- The transmit power of each sensor is a constant, and we don't take power tuning into account.

- Sensors communicate by broadcasting, and can dynamically switch to different sub-carriers.

- Sensors are equipped with full-duplex radio, enabling transmitting and receiving at the same time on different sub-carriers.

- Sensors are synchronized with reference to a global clock for time synchronized operation.

Chapter 3

Cooperative Spectrum Sensing through Imperfect Reporting Channels

3.1 Energy Detection under Rician Fading

The essential of local spectrum sensing performed individually at each sensor is a binary hypothesis-testing problem:

$$\begin{array}{lll} \mathcal{H}_0: & y_i(t) = n_i(t) & \text{for primary signal absence} \\ \mathcal{H}_1: & y_i(t) = h_i(t) \cdot x(t) + n_i(t) & \text{for primary signal presence} \end{array}$$

where $y_i(t)$ is the received signal at s_i, $n_i(t)$ is assumed to be circularly symmetric complex additive white Gaussian noise (AWGN), $h_i(t)$ denotes the amplitude gain from the primary transmitter to s_i, and $x(t)$ is the transmitted primary signal at time instant t.

For facilitating the analysis of sensing performance, we assume that each sensor employs *Energy Detection* as the local sensing scheme, which does not need any prior information of primary signals. The received signal passes through an input bandpass filter, where the center frequency f_s and one-sided bandwidth, W_D are selected. Then a squaring device and integrator measure the received energy over an observation interval, T. Finally, the test statistic, Λ, produced by the integrator is compared to a predefined threshold, λ, to make the local decision [31]. Regarding the local spectrum sensing at s_i, the probabilities of detection, false alarm and missed detection over AWGN channels are given, respectively, by

$$P_{d,i} = \text{Prob}\{\Lambda_i > \lambda_i | \mathcal{H}_1\} = Q_u(\sqrt{2\gamma_i}, \sqrt{\lambda_i}) \tag{3.1}$$

$$P_{f,i} = \text{Prob}\{\Lambda_i > \lambda_i | \mathcal{H}_0\} = \frac{\Gamma(u, \lambda_i/2)}{\Gamma(u)} \tag{3.2}$$

$$P_{m,i} = \text{Prob}\{\Lambda_i < \lambda_i | \mathcal{H}_1\} = 1 - P_{d,i} = 1 - Q_u(\sqrt{2\gamma_i}, \sqrt{\lambda_i}) \tag{3.3}$$

where λ_i denotes the energy detection threshold, $\gamma_i = \frac{P_{R,i}}{N_0 W_D}$ is the instantaneous received SNR of primary signal at s_i ($P_{R,i}$ is the received primary signal power at s_i, and N_0 is the one-sided noise power spectral density), $u = TW_D$ is the time-bandwidth product, which is assumed to be an integer for simplicity, $\Gamma(\cdot)$, $\Gamma(\cdot, \cdot)$ are the complete and upper incomplete gamma functions respectively, and $Q_u(\cdot, \cdot)$ is the generalized Marcum Q-function, which are defined as follows

$$\Gamma(a) = \int_0^\infty t^a e^{-t} dt \tag{3.4}$$

$$\Gamma(a, b) = \int_b^\infty t^{a-1} e^{-t} dt \tag{3.5}$$

$$Q_u(a, b) = \int_b^\infty \frac{x^u}{a^{u-1}} \exp\left(-\frac{x^2 + a^2}{2}\right) I_{u-1}(ax) dx \tag{3.6}$$

where I_{u-1} is the $(u-1)$th order modified Bessel function of the first kind. Note that $P_{f,i}$ does not depend on γ_i since there is no primary signal under $\mathcal{H}_0$.

We average (3.1) and (3.2) over Rician fading with the PDF of (2.4), and obtain the average probabilities of detection and false alarm [32]:

$$P_{d,i} = \int_{\text{all } \gamma_i} Q_u(\sqrt{2\gamma_i}, \sqrt{\lambda_i}) f_{\text{Ric}}(\gamma_i) d\gamma_i, \tag{3.7}$$

$$P_{f,i} = \int_{\text{all } \gamma_i} \frac{\Gamma(u, \lambda_i/2)}{\Gamma(u)} f_{\text{Ric}}(\gamma_i) d\gamma_i = \frac{\Gamma(u, \lambda_i/2)}{\Gamma(u)}. \tag{3.8}$$

where $P_{f,i}$ is the same as that over an AWGN channel since it is independent of received SNR γ_i.

As shown in [32], when $u = 1$, (3.7) has the form

$$P_{d,i} = Q_1\left(\sqrt{\frac{2K\bar{\gamma}_i}{K+1+\bar{\gamma}_i}}, \sqrt{\frac{\lambda_i(K+1)}{K+1+\bar{\gamma}_i}}\right) \tag{3.9}$$

where $\bar{\gamma}_i$ is the average received SNR of the primary signal at s_i, which is determined by the path loss alone in our system as explained in (2.4), i.e.

$$\bar{\gamma}_i = \frac{P_T A d_i^{-\mu}}{N_0 W_D} \tag{3.10}$$

with P_T denoting the transmit power of the primary signal, and d_i the distance between the primary transmitter and s_i. Although in other general cases, there is no closed-form expression for (3.7), we can still use numerical integration to analyse the performance. In the following analysis, we use $u = 1$ for simplicity.

3.2 Hard Decision Fusion through Imperfect Reporting Channels

In this work, we consider hard decision fusion because of its low communication cost. Specifically, followed by local spectrum sensing, each sensor gathers 1-bit hard decisions from its cooperative partners, and uses *k-out-of-n* rule for decision fusion [33, pp. 59-61]. We define the *partner set* of a sensor s_i as $\mathcal{J}^{(i)} = \{s_j | R_{j,i} = 1\}$, and use n_i to denote the number of partners, i.e. $n_i = |\mathcal{J}^{(i)}|$, where $|\cdot|$ represents the cardinality of a set. If

$R_{i,i} = 1$, s_i is also considered as a *partner* of itself. s_i declares $\mathcal{H}_1$ when at least k out of the n_i number of its partners have reported on $\mathcal{H}_1$. Otherwise, it outputs $\mathcal{H}_0$. The *OR*, *AND*, and *MAJORITY* rules are special cases of the *k-out-of-n* rule with $k = 1$, $k = n$, and $k = \lceil n/2 \rceil$ respectively, where $\lceil \cdot \rceil$ denotes the ceiling operator.

In order to analyse the impacts of reporting channel errors, we model the reporting channel from s_j to s_i as a binary symmetric channel (BSC) with cross-over probability $\varepsilon_{j,i} \in [0, 0.5)$, which is equal to the BEP of the channel [18]. Since our work is not constrained to a certain modulation scheme, without loss of generality, we choose the simplest form, binary phase shift keying (BPSK) for hard decision transmissions. The transmitted signal over one bit time period T_b is [28, p. 146]

$$s_n(t) = \sqrt{E_b} g(t) \cos[2\pi f_c t + \pi(n-1)], \quad n = 0, 1$$

where E_b is the signal energy per bit, f_c is the carrier frequency, and $g(t)$ is the baseband pulse shape satisfying [28, p. 130]

$$\int_0^{T_b} g^2(t) \cos^2(2\pi f_c t) dt = 1 \tag{3.11}$$

and

$$\int_0^{T_b} g^2(t) \cos(2\pi f_c t) \sin(2\pi f_c t) dt = 0. \tag{3.12}$$

If the bandwidth of the pulse shaping $g(t)$ is assumed to satisfy $B = 1/T_b$, then $E_b = P_R/B = P_R \cdot T_b$, and the received SNR per bit γ_b is [28, p. 173]

$$\gamma_b = \frac{P_R}{N_0 B} = \frac{P_R}{N_0(1/T_b)} = \frac{P_R T_b}{N_0} = \frac{E_b}{N_0} \tag{3.13}$$

For BPSK the received SNR per bit γ_b, is equal to the received SNR γ, so we do not need to differentiate SNR and SNR per bit in the rest of this thesis, and the average BEP for

BPSK in Rician fading on the reporting channel from s_j to s_i is [29, p. 126]

$$\varepsilon_{j,i} = \frac{1}{\pi}\int_0^{\pi/2} \frac{(1+K)\sin^2\phi}{(1+K)\sin^2\phi+\bar{\gamma}_{j,i}} \exp\left[-\frac{K\bar{\gamma}_{j,i}}{(1+K)\sin^2\phi+\bar{\gamma}_{j,i}}\right] d\phi \tag{3.14}$$

where K is the Rician parameter, and $\bar{\gamma}_{j,i}$ is the average received SNR at s_i in Rician fading. In our system, $\bar{\gamma}_{j,i}$ is affected by the path loss alone, and the noise includes both AWGN and additive interference, and thus according to (2.7),

$$\bar{\gamma}_{j,i} = \frac{A d_{i,j}^{-\mu} P_T^{(i)}}{P_N + \sum_{k\in\mathcal{K}\setminus\{i\}} A d_{k,j}^{-\mu} P_T^{(k)}} \tag{3.15}$$

where $P_T^{(i)}$ is the transmit power of s_i, P_N is the thermal noise power in the frequency of operation ($P_N = N_0 W_c$ if the communication bandwidth is W_c), and $\mathcal{K}\setminus\{i\}$ denotes the set of sensors transmitting simultaneously on the same channel other than s_i.

If an error occurs when $s_j \in \mathcal{J}^{(i)}$ transmits its binary decisions to s_i, a $\mathcal{H}_1$ (or $\mathcal{H}_0$) decision of s_j will be turned into $\mathcal{H}_0$ (or $\mathcal{H}_1$) when received by s_i. We use X_j to denote the binary decision made by s_j, and $Y_j^{(i)}$ to denote the binary decision received by s_i from s_j. Thus, $\forall s_j \in \mathcal{J}^{(i)}$, the equivalent probabilities of detection and false alarm received by

s_i are

$$\begin{aligned}
P_{d,j}^{(i)} &\triangleq \text{Prob}\{Y_j^{(i)} = 1|\mathcal{H}_1\} \\
&= \text{Prob}\{Y_j^{(i)} = 1|X_j = 1\} \cdot \text{Prob}\{X_j = 1|\mathcal{H}_1\} \\
&\quad + \text{Prob}\{Y_j^{(i)} = 1|X_j = 0\} \cdot \text{Prob}\{X_j = 0|\mathcal{H}_1\} \\
&= (1 - \varepsilon_{j,i})P_{d,j} + \varepsilon_{j,i}(1 - P_{d,j}) \\
&= P_{d,j} - 2\varepsilon_{j,i}P_{d,j} + \varepsilon_{j,i} \qquad (3.16) \\
P_{f,j}^{(i)} &\triangleq \text{Prob}\{Y_j^{(i)} = 1|\mathcal{H}_0\} \\
&= \text{Prob}\{Y_j^{(i)} = 1|X_j = 1\} \cdot \text{Prob}\{X_j = 1|\mathcal{H}_0\} \\
&\quad + \text{Prob}\{Y_j^{(i)} = 1|X_j = 0\} \cdot \text{Prob}\{X_j = 0|\mathcal{H}_0\} \\
&= (1 - \varepsilon_{j,i})P_{f,j} + \varepsilon_{j,i}(1 - P_{f,j}) \\
&= P_{f,j} - 2\varepsilon_{j,i}P_{f,j} + \varepsilon_{j,i}. \qquad (3.17)
\end{aligned}$$

We assume that each sensor experiences i.i.d. Rician fading, and performs spectrum sensing independently. Using indicator function as well as binary vector notation, [27, Eq. (11)] gives the expression for the probability of detection for *k-out-of-n* decision fusion with reporting channel errors. The probability of false alarm can be drawn by replacing P_d with P_f. However, in this thesis, we give the expressions for these probabilities in a more concise and direct form, with combination notation as specified in [34].

We use $\binom{\mathcal{J}^{(i)}}{m}$ to represent the set of all m-combinations of $\mathcal{J}^{(i)}$, which has $\binom{n_i}{m} = \frac{n_i!}{m!(n_i-m)!}$ members. $\mathcal{J}_{m,l}^{(i)} \in \binom{\mathcal{J}^{(i)}}{m}, 1 \leqslant l \leqslant \binom{n_i}{m}$, represents one m-combination of $\mathcal{J}^{(i)}$, which is a subset of $\mathcal{J}^{(i)}$ consisting of m distinct elements of $\mathcal{J}^{(i)}$, and $\mathcal{J}^{(i)} \backslash \mathcal{J}_{m,l}^{(i)}$ denotes the relative complement of $\mathcal{J}_{m,l}^{(i)}$ in $\mathcal{J}^{(i)}$. When generating $\binom{\mathcal{J}^{(i)}}{m}$ from $\mathcal{J}^{(i)}$, we always first select the sensor with the smallest index i if possible, and thus $\mathcal{J}_{m,l}^{(i)}$ are arranged in lexicographic order. For instance, if the four partners of s_1 are s_1, s_2, s_3, and s_4, i.e. $\mathcal{J}^{(1)} = \{s_1, s_2.s_3, s_4\}$, then the set of all 2-combinations of $\mathcal{J}^{(1)}$ is $\binom{\mathcal{J}^{(1)}}{2} =$

$\{\{s_1, s_2\}, \{s_1, s_3\}, \{s_1, s_4\}, \{s_2, s_3\}, \{s_2, s_4\}, \{s_3, s_4\}\}$. $\mathcal{J}_{2,1}^{(1)} = \{s_1, s_2\}$, $\mathcal{J}_{2,2}^{(1)} = \{s_1, s_3\}$, $\mathcal{J}_{2,3}^{(1)} = \{s_1, s_4\}$, …, $\mathcal{J}_{2,6}^{(1)} = \{s_3, s_4\}$. The corresponding relative complements are $\mathcal{J}^{(1)} \backslash \mathcal{J}_{2,1}^{(1)} = \{s_3, s_4\}$, $\mathcal{J}^{(1)} \backslash \mathcal{J}_{2,2}^{(1)} = \{s_2, s_4\}$, $\mathcal{J}^{(1)} \backslash \mathcal{J}_{2,3}^{(1)} = \{s_2, s_3\}$, …, $\mathcal{J}^{(1)} \backslash \mathcal{J}_{2,6}^{(1)} = \{s_1, s_2\}$.

Then, we can transform [27, Eq. (11)], and obtain s_i's probabilities of detection and false alarm after k_i-*out-of*-n_i decision fusion,

$$Q_{d,i}(k_i) = \sum_{m=k_i}^{n_i} \sum_{l=1}^{\binom{n_i}{m}} \left(\prod_{s_j \in \mathcal{J}_{m,l}^{(i)}} P_{d,j}^{(i)} \prod_{s_j \in \mathcal{J}^{(i)} \backslash \mathcal{J}_{m,l}^{(i)}} (1 - P_{d,j}^{(i)}) \right) \tag{3.18}$$

$$Q_{f,i}(k_i) = \sum_{m=k_i}^{n_i} \sum_{l=1}^{\binom{n_i}{m}} \left(\prod_{s_j \in \mathcal{J}_{m,l}^{(i)}} P_{f,j}^{(i)} \prod_{s_j \in \mathcal{J}^{(i)} \backslash \mathcal{J}_{m,l}^{(i)}} (1 - P_{f,j}^{(i)}) \right) \tag{3.19}$$

where $P_{d,j}^{(i)}$ and $P_{f,j}^{(i)}$ are given in (3.16) and (3.17), and $1 \leqslant k_i \leqslant n_i$ is s_i's decision fusion threshold. In addition, the corresponding probability of missed detection is

$$\begin{aligned} Q_{m,i}(k_i) = 1 - Q_{d,i}(k_i) &= 1 - \sum_{m=k_i}^{n_i} \sum_{l=1}^{\binom{n_i}{m}} \left(\prod_{s_j \in \mathcal{J}_{m,l}^{(i)}} P_{d,j}^{(i)} \prod_{s_j \in \mathcal{J}^{(i)} \backslash \mathcal{J}_{m,l}^{(i)}} (1 - P_{d,j}^{(i)}) \right) \\ &= \sum_{m=0}^{k_i-1} \sum_{l=1}^{\binom{n_i}{m}} \left(\prod_{s_j \in \mathcal{J}_{m,l}^{(i)}} P_{d,j}^{(i)} \prod_{s_j \in \mathcal{J}^{(i)} \backslash \mathcal{J}_{m,l}^{(i)}} (1 - P_{d,j}^{(i)}) \right). \end{aligned} \tag{3.20}$$

Furthermore, (3.18) and (3.19) can be viewed as the complements of cumulative distributed functions (CDF) for the Poisson-Binomial distribution. Please note that Poisson-Binomial distribution is the discrete probability distribution of a sum of independent Bernoulli trials when the individual probabilities of success are not necessarily identical [34, 35]. The ordinary binomial distribution can be viewed as a special case of it when all the individual probabilities of success are the same. Suppose that there are n_i Bernoulli trials, each of which has a success probability $P_{d,j}^{(i)}$ (or $P_{f,j}^{(i)}$). Then, $Q_{d,i}(k_i)$ (or $Q_{f,i}(k_i)$) is the probability that at least k_i of these n_i trials are successful, which is the complement of

Poisson-Binomial CDF as defined in [35, Eq. (8)]

$$Q(k) = \text{Prob}\{\text{At least } k \text{ successes in } N \text{ Bernoulli trials}\} \tag{3.21}$$

$$= 1 - \frac{1}{N+1}\sum_{n=0}^{N}\left\{\frac{1-\exp[-\mathbf{j}2\pi nk/(N+1)]}{1-\exp[-\mathbf{j}2\pi n/(N+1)]}\prod_{m=1}^{N}\left(1-p_m+p_m\exp[\mathbf{j}2\pi n/(N+1)]\right)\right\}. \tag{3.22}$$

where p_m is the success probability of each Bernoulli trial. Thus, in (3.22), if N is substituted with n_i, k is substituted with k_i, and each p_m is substituted with $P_{d,j}^{(i)}$ (or $P_{f,j}^{(i)}$), then we can obtain the closed-form expression for $Q_{d,i}(k_i)$ (or $Q_{f,i}(k_i)$):

$$Q_{d,i}(k_i) = 1 - \frac{1}{n_i+1}\sum_{l=0}^{n_i}\left\{\frac{1-\exp[-\mathbf{j}2\pi lk_i/(n_i+1)]}{1-\exp[-\mathbf{j}2\pi l/(n_i+1)]}\prod_{s_j\in\mathcal{J}^{(i)}}\left(1-P_{d,j}^{(i)}+P_{d,j}^{(i)}\exp[\mathbf{j}2\pi l/(n_i+1)]\right)\right\} \tag{3.23}$$

$$Q_{f,i}(k_i) = 1 - \frac{1}{n_i+1}\sum_{l=0}^{n_i}\left\{\frac{1-\exp[-\mathbf{j}2\pi lk_i/(n_i+1)]}{1-\exp[-\mathbf{j}2\pi l/(n_i+1)]}\prod_{s_j\in\mathcal{J}^{(i)}}\left(1-P_{f,j}^{(i)}+P_{f,j}^{(i)}\exp[\mathbf{j}2\pi l/(n_i+1)]\right)\right\}. \tag{3.24}$$

When n_i and k_i are large, efficient methods as well as approximation algorithms to compute (3.23) and (3.24) are discussed in [34, 35]. However, they are out of the scope of this work.

3.3 Performance Degradation Caused by Reporting Errors

3.3.1 Single Link

We start from a single link $s_j \rightarrow s_i$ to analyse the impact of reporting channel errors. Since in this subsection we only consider a single link, for simplicity we introduce the following notations in which the sensor ID i and j are omitted:

- P_d and P_f: The local probabilities of detection and false alarm under Rician fading at s_j as shown in (3.7) and (3.8), i.e.

$$P_d = \int_{\text{all } \gamma} Q_u(\sqrt{2\gamma}, \sqrt{\lambda}) f_{\text{Ric}}(\gamma) d\gamma \tag{3.25}$$

$$P_f = \int_{\text{all } \gamma} \frac{\Gamma(u, \lambda/2)}{\Gamma(u)} f_{\text{Ric}}(\gamma) d\gamma = \frac{\Gamma(u, \lambda/2)}{\Gamma(u)}. \tag{3.26}$$

- $P_{d,\varepsilon}$ and $P_{f,\varepsilon}$: The equivalent probabilities of detection and false alarm received by s_i over a reporting channel with BEP ε as shown in (3.16) and (3.17), i.e.

$$P_{d,\varepsilon} = P_d - 2\varepsilon P_d + \varepsilon \tag{3.27}$$

$$P_{f,\varepsilon} = P_f - 2\varepsilon P_f + \varepsilon. \tag{3.28}$$

- $P_d|_{P_f=\alpha}$ and $P_{d,\varepsilon}|_{P_{f,\varepsilon}=\alpha}$: The value of P_d (or $P_{d,\varepsilon}$) when the threshold λ is chosen such that $P_f = \alpha$ (or $P_{f,\varepsilon} = \alpha$). Only λ is considered as a variable here.

- $P_f|_{P_d=\beta}$ and $P_{f,\varepsilon}|_{P_{d,\varepsilon}=\beta}$: The value of P_f (or $P_{f,\varepsilon}$) when the threshold λ is chosen such that $P_d = \beta$ (or $P_{d,\varepsilon} = \beta$). Only λ is considered as a variable here.

Fig. 3.1 shows the impact of reporting channel error on a single communication link by plotting $P_{d,\varepsilon}$ vs $P_{f,\varepsilon}$ for different values of ε. We set the Rician fading parameter $K = 7$, time-bandwidth product $u = 1$, and the average received SNR of the primary signal under Rician fading $\bar{\gamma} = 4$dB. For each value of ε, we change the value of λ from 0 to ∞, and calculate $P_{d,\varepsilon}$ and $P_{f,\varepsilon}$ according to (3.25) – (3.28). Then, for each ε, we obtain an equivalent receiver operating characteristic (ROC) curves at the receiving sensor, which characterizes spectrum sensing performance. From Fig. 3.1 we can see that, when ε increases, the minimum values of $P_{d,\varepsilon}$ and $P_{f,\varepsilon}$ increases from 0 to 0.5, while the maximum values of them decreases from 1 to 0.5. In summary, the range of the ROC curve shrinks, and the ROC curve shifts down.

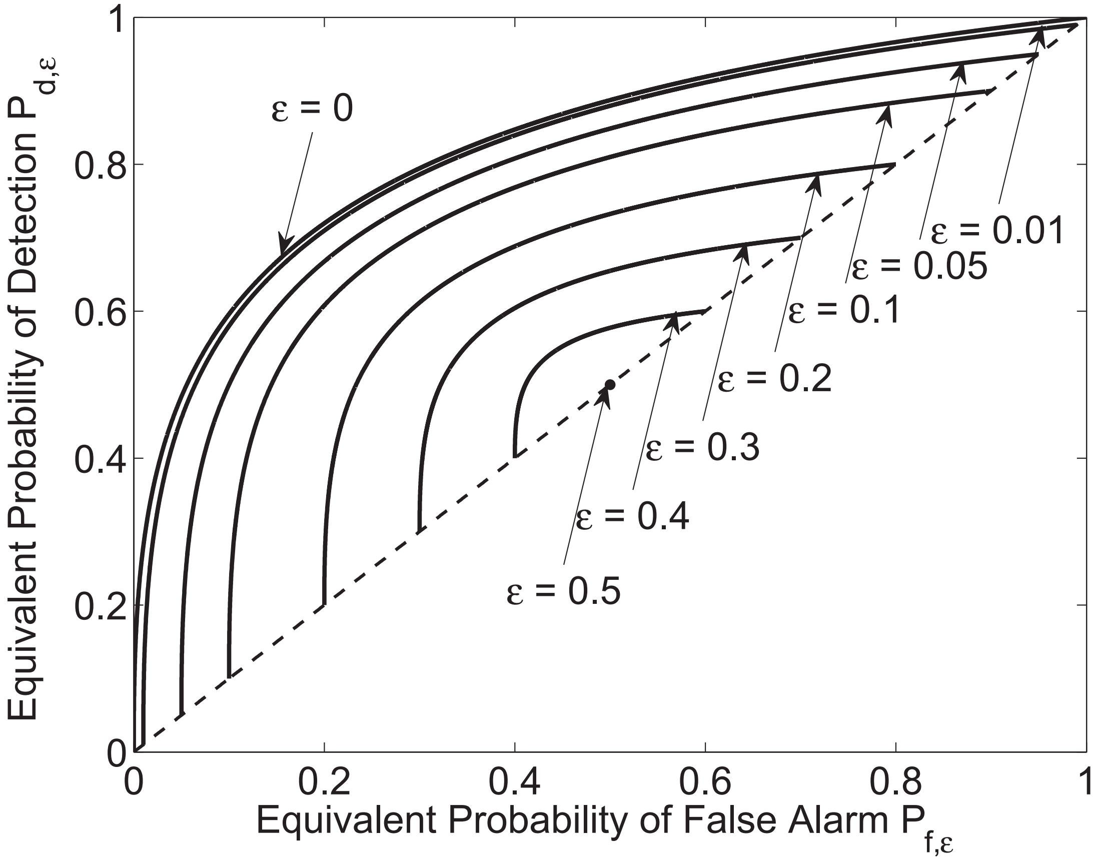

Fig. 3.1 $P_{d,\varepsilon}$ vs $P_{f,\varepsilon}$ for different values of reporting channel error ε on a single communication link, when the Rician fading parameter is $K = 7$, and the average received SNR of the primary signal is $\bar{\gamma} = 4$dB.

From (3.25) and (3.26) we get, $P_d = P_f = 1$ when $\lambda = 0$, and $P_d = P_f \to 0$ when $\lambda \to \infty$. Then from (3.27) and (3.28), we conclude the following results:

- When $\lambda \to \infty$, $P_{d,\varepsilon} \to 0 - 2\varepsilon \times 0 + \varepsilon = \varepsilon$, and $P_{f,\varepsilon} \to 0 - 2\varepsilon \times 0 + \varepsilon = \varepsilon$.
- When $\lambda = 0$, $P_{d,\varepsilon} = 1 - 2\varepsilon \times 1 + \varepsilon = 1 - \varepsilon$, and $P_{f,\varepsilon} = 1 - 2\varepsilon \times 1 + \varepsilon = 1 - \varepsilon$.

$(P_{d,\varepsilon}, P_{f,\varepsilon})$ is bounded by $(\varepsilon, \varepsilon)$ and $(1-\varepsilon, 1-\varepsilon)$. Thus, from Fig. 3.1 we can see that the range of $(P_{d,\varepsilon}, P_{f,\varepsilon})$ shrinks linearly as ε increases, and when $\varepsilon = 0.5$ there is only one point $(0.5, 0.5)$ left.

In addition, when the reporting channel error increases from ε_1 to ε_2, we are concerned about the difference of P_{d,ε_1} and P_{d,ε_2} when $P_{f,\varepsilon_1} = P_{f,\varepsilon_2}$, or the difference of P_{f,ε_1} and P_{f,ε_2} when $P_{d,\varepsilon_1} = P_{d,\varepsilon_2}$. Therefore, we define

$$\Phi_d^{(\varepsilon_1,\varepsilon_2,\alpha)} \triangleq P_{d,\varepsilon_1}|_{P_{f,\varepsilon_1}=\alpha} - P_{d,\varepsilon_2}|_{P_{f,\varepsilon_2}=\alpha} \tag{3.29}$$

$$\Phi_f^{(\varepsilon_1,\varepsilon_2,\beta)} \triangleq P_{f,\varepsilon_1}|_{P_{d,\varepsilon_1}=\beta} - P_{f,\varepsilon_2}|_{P_{d,\varepsilon_2}=\beta}. \tag{3.30}$$

Then we obtain the following properties for the impact of reporting channel errors for a single link, which are consistent with Fig. 3.1:

Property 1. *When $\varepsilon_2 > \varepsilon_1$ and $\varepsilon_2 < \alpha < 1-\varepsilon_2$, $\Phi_d^{\varepsilon_1,\varepsilon_2,\alpha}$ decreases with α when $\alpha < 0.5$, and increases with α when $\alpha > 0.5$. In addition, $\Phi_d^{(\varepsilon_1,\varepsilon_2,\alpha)} > 0$, i.e. $P_{d,\varepsilon_1}|_{P_{f,\varepsilon_1}=\alpha} > P_{d,\varepsilon_2}|_{P_{f,\varepsilon_2}=\alpha}$.*

Proof. Please see Appendix A. □

Property 2. *When $\varepsilon_2 > \varepsilon_1$ and $\varepsilon_2 < \beta < 1-\varepsilon_2$, $\Phi_f^{\varepsilon_1,\varepsilon_2,\beta}$ decreases with β when $\beta < 0.5$, and increases with β when $\beta > 0.5$. In addition, $\Phi_f^{\varepsilon_1,\varepsilon_2,\beta} < 0$, i.e. $P_{f,\varepsilon_1}|_{P_{d,\varepsilon_1}=\beta} < P_{f,\varepsilon_2}|_{P_{d,\varepsilon_2}=\beta}$.*

Proof. The proof is very similar to that of Property 1, and thus omitted. □

3.3.2 Multiple Links

When considering different BEPs on several communication links for decision fusion, the problem becomes more complicated. Fig. 3.2 shows the impacts of reporting channel errors on cooperative spectrum sensing performance through the ROC curves displaying $Q_{d,i}$ versus $Q_{f,i}$, which are calculated according to (3.18) and (3.19). There are 5 sensors in this simple scenario employing identical energy detection threshold λ, and only s_1 makes the global decision after gathering the binary decisions from others. Thus, we can omit the second subscript i in $Q_{d,i}$ and $Q_{f,i}$. The Rician fading parameter is $K = 7$, and the received SNR of the primary signal is identical at each sensor, which is denoted by $\bar{\gamma}$.

In Fig. 3.2-A, the vertical separation between curves indicates the degradation introduced by reporting channel errors for *OR* fusion rule, which is small in high Q_f range, and increases as Q_f decreases. Moreover, for ROC curves with erroneous reporting channels, when Q_f decreases to a certain level, Q_d will drastically decrease, and any Q_f lower than a threshold is not possible. In Fig. 3.2-B, we fix the average received SNR on each reporting channel, and compare the ROC curves with different $\bar{\gamma}$ as well as different fusion thresholds k_1. For a certain k_1, higher $\bar{\gamma}$ results in better detection performance with higher Q_d. However, the minimum values of Q_d and Q_f are no longer 0 as that with error-free reporting channels, and doesn't depend on $\bar{\gamma}$.

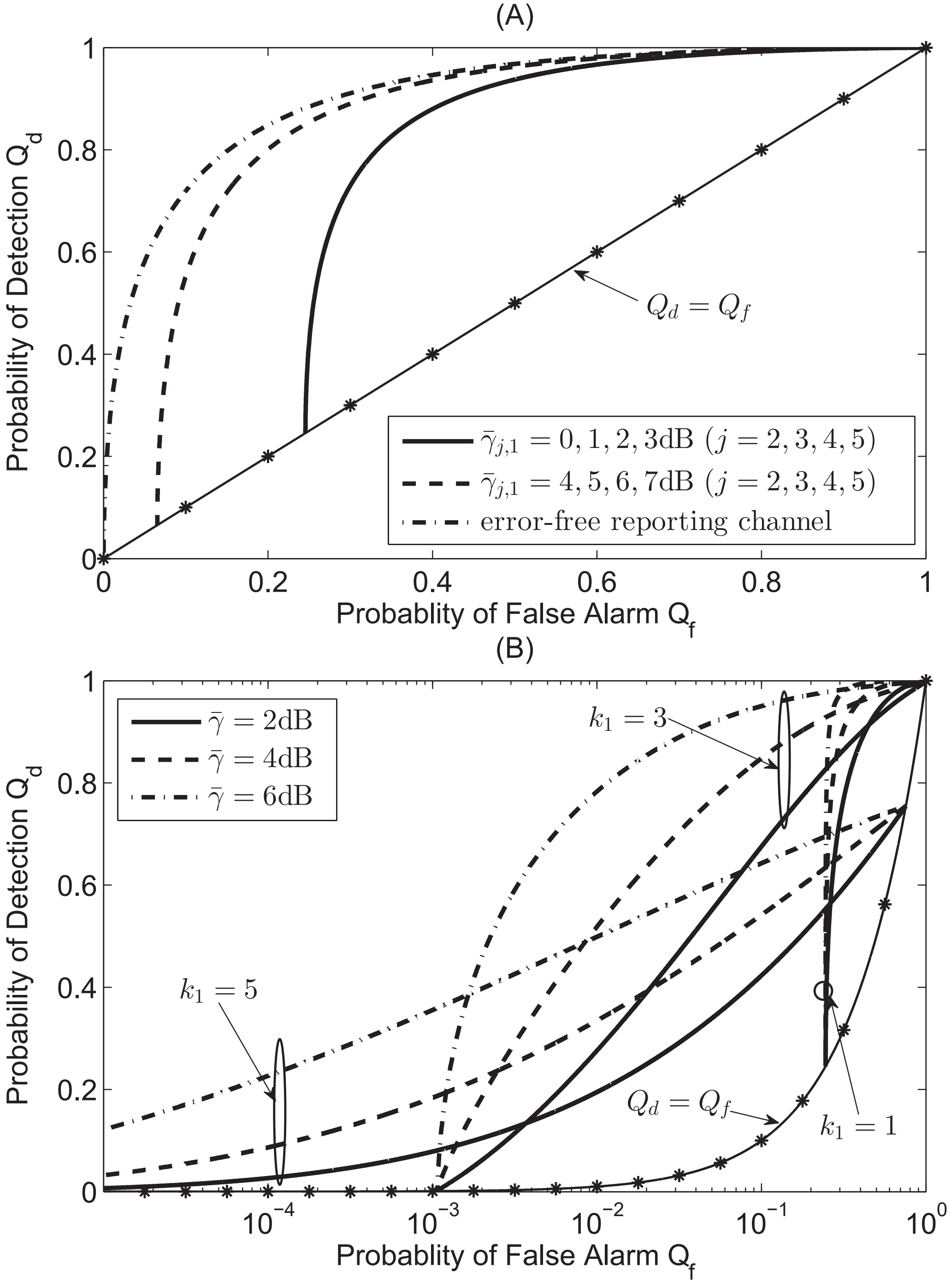

Fig. 3.2 Q_d vs Q_f at s_1 when $n_1 = 5$, and $K = 7$, $\bar{\gamma}$ is the identical received SNR of primary signal, and $\bar{\gamma}_{j,1} (j \neq 1)$ represents the average received SNR on each reporting channel. (A) shows impacts of the reporting channel errors on *OR* rule with different sets of reporting channels, where $\bar{\gamma} = 2\text{dB}$, and $k_1 = 1$. (B) compares the ROC curves with different $\bar{\gamma}$, and different decision fusion thresholds, k_1, where $\bar{\gamma}_{j,1} = 0, 1, 2, 3\text{dB}$ $(j = 2, 3, 4, 5)$.

From the analysis of the effects of reporting error on a single link in the last subsection, we have that if the BEP on the communication link $s_j \to s_i$ is $\varepsilon_{j,i}$, the minimum and maximum values of the equivalent probabilities of detection and false alarm received by s_i (i.e. $P_{d,j}^{(i)}$ and $P_{f,j}^{(i)}$ in (3.16) and (3.17)) are

$$\min P_{d,j}^{(i)} = \min P_{f,j}^{(i)} = \varepsilon_{j,i} \tag{3.31}$$

$$\max P_{d,j}^{(i)} = \max P_{f,j}^{(i)} = 1 - \varepsilon_{j,i} \tag{3.32}$$

The probabilities of detection and false alarm using the k_i-*out-of*-n_i decision fusion rule are given in (3.18) and (3.19),

$$Q_{d,i}(k_i) = \sum_{m=k_i}^{n_i} \sum_{l=1}^{\binom{n_i}{m}} \left(\prod_{s_j \in \mathcal{J}_{m,l}^{(i)}} P_{d,j}^{(i)} \prod_{s_j \in \mathcal{J}^{(i)} \setminus \mathcal{J}_{m,l}^{(i)}} (1 - P_{d,j}^{(i)}) \right)$$

$$Q_{f,i}(k_i) = \sum_{m=k_i}^{n_i} \sum_{l=1}^{\binom{n_i}{m}} \left(\prod_{s_j \in \mathcal{J}_{m,l}^{(i)}} P_{f,j}^{(i)} \prod_{s_j \in \mathcal{J}^{(i)} \setminus \mathcal{J}_{m,l}^{(i)}} (1 - P_{f,j}^{(i)}) \right)$$

Property 3. *For fixed partner set $\mathcal{J}^{(i)}$, and decision fusion threshold k_i, if each $P_{d,j}^{(i)}$ in (3.18) takes its minimum value (i.e. $\min P_{d,j}^{(i)} = \varepsilon_{j,i}$ as shown in (3.31)), then the probability of detection using k_i-out-of-n_i decision fusion, $Q_{d,i}(k_i)$, reaches its minimum value. If each $P_{d,j}^{(i)}$ in (3.18) takes its maximum value (i.e. $\max P_{d,j}^{(i)} = 1 - \varepsilon_{j,i}$ as shown in (3.32)), then the probability of detection using k_i-out-of-n_i decision fusion, $Q_{d,i}(k_i)$, reaches its maximum value.*

Proof. According to Lemma 2 in Appendix B, we can conclude that $Q_{d,i}(k_i)$ is monotonic non-decreasing in each $P_{d,j}^{(i)}$, and hence Property 3 is proved. □

Similarly, we have the following property regarding minimum and maximum values of $Q_{f,i}$.

Property 4. *For fixed partner set $\mathcal{J}^{(i)}$, and decision fusion threshold k_i, if each $P_{f,j}^{(i)}$ in (3.19) takes its minimum value (i.e. $\min P_{f,j}^{(i)} = \varepsilon_{j,i}$ as shown in (3.31)), then the probability of false alarm using k_i-out-of-n_i decision fusion, $Q_{f,i}(k_i)$, reaches its minimum value. If each $P_{f,j}^{(i)}$ in (3.19) takes its maximum value (i.e. $\max P_{f,j}^{(i)} = 1 - \varepsilon_{j,i}$ as shown in (3.32)), then the probability of false alarm using k_i-out-of-n_i decision fusion, $Q_{f,i}(k_i)$, reaches its maximum value.*

Proof. According to Lemma 2 in Appendix B, we can conclude that $Q_{f,i}(k_i)$ is monotonic non-decreasing in each $P_{f,j}^{i}$, and hence Property 4 is proved. □

Therefore, when substituting $P_{d,j}^{(i)}$ and $P_{f,j}^{(i)}$ in (3.18) and (3.19) with their minimum values $\varepsilon_{j,i}$ as shown in (3.31), we get s_i's minimum values of probabilities of detection and false alarm for a certain k_i-*out-of-*n_i decision fusion, which are denoted by $Q_{d,i}^{[\mathrm{L}]}(k_i)$, and $Q_{f,i}^{[\mathrm{L}]}(k_i)$ respectively.

$$Q_{d,i}^{[\mathrm{L}]}(k_i) = Q_{f,i}^{[\mathrm{L}]}(k_i) = \sum_{m=k_i}^{n_i} \sum_{l=1}^{\binom{n_i}{m}} \Bigg(\prod_{s_j \in \mathcal{J}_{m,l}^{(i)}} \varepsilon_{j,i} \prod_{s_j \in \mathcal{J}^{(i)} \setminus \mathcal{J}_{m,l}^{(i)}} (1 - \varepsilon_{j,i}) \Bigg). \tag{3.33}$$

where k_i represents the chosen fusion threshold. For fixed partner set $\mathcal{J}^{(i)}$, and decision fusion threshold k_i, any $Q_{d,i}(k_i)$ (or $Q_{f,i}(k_i)$) lower than $Q_{d,i}^{[\mathrm{L}]}(k_i)$ (or $Q_{f,i}^{[\mathrm{L}]}(k_i)$) is not achievable.

Similarly, substituting $P_{d,j}^{(i)}$ and $P_{f,j}^{(i)}$ in (3.18) and (3.19) with their maximum values $1-\varepsilon_{j,i}$ as shown in (3.32), yields s_i's maximum values of probabilities of detection and false alarm for a certain k_i-*out-of-*n_i decision fusion, which are denoted by $Q_{d,i}^{[\mathrm{U}]}(k_i)$, and $Q_{f,i}^{[\mathrm{U}]}(k_i)$ respectively.

$$Q_{d,i}^{[\mathrm{U}]}(k_i) = Q_{f,i}^{[\mathrm{U}]}(k_i) = \sum_{m=k_i}^{n_i} \sum_{l=1}^{\binom{n_i}{m}} \Bigg(\prod_{s_j \in \mathcal{J}_{m,l}^{(i)}} (1 - \varepsilon_{j,i}) \prod_{s_j \in \mathcal{J}^{(i)} \setminus \mathcal{J}_{m,l}^{(i)}} \varepsilon_{j,i} \Bigg) \tag{3.34}$$

For fixed partner set $\mathcal{J}^{(i)}$, and decision fusion threshold k_i, any $Q_{d,i}(k_i)$ (or $Q_{f,i}(k_i)$) higher than $Q_{d,i}^{[\mathrm{U}]}(k_i)$ (or $Q_{f,i}^{[\mathrm{U}]}(k_i)$) is not achievable.

The *OR* fusion rule is a special case when $k_i = 1$. Since $\binom{n_i}{0} = 1$, $\mathcal{J}_{0,l}^{(i)} = \emptyset$, and $\mathcal{J}^{(i)} = \{s_j | R_{j,i} = 1\}$, from (3.33) and (3.34), we get

$$\begin{aligned} Q_{d,i}^{[\mathrm{L}]}(1) = Q_{f,i}^{[\mathrm{L}]}(1) &= 1 - \sum_{l=1}^{\binom{n_i}{0}} \left(\prod_{s_j \in \mathcal{J}_{0,l}^{(i)}} \varepsilon_{j,i} \prod_{s_j \in \mathcal{J}^{(i)} \setminus \mathcal{J}_{0,l}^{(i)}} (1 - \varepsilon_{j,i}) \right) \\ &= 1 - \prod_{s_j \in \{s_j | R_{j,i}=1\}} (1 - \varepsilon_{j,i}) \end{aligned} \tag{3.35}$$

$$\begin{aligned} Q_{d,i}^{[\mathrm{U}]}(1) = Q_{f,i}^{[\mathrm{U}]}(1) &= 1 - \sum_{l=1}^{\binom{n_i}{0}} \left(\prod_{s_j \in \mathcal{J}_{0,l}^{(i)}} (1 - \varepsilon_{j,i}) \prod_{s_j \in \mathcal{J}^{(i)} \setminus \mathcal{J}_{0,l}^{(i)}} \varepsilon_{j,i} \right) \\ &= 1 - \prod_{s_j \in \{s_j | R_{j,i}=1\}} \varepsilon_{j,i}. \end{aligned} \tag{3.36}$$

Then, we define the *Achievable Range* of $Q_{d,i}(k_i)$ and $Q_{f,i}(k_i)$ as

$$\Delta_{d,i}(k_i) \triangleq Q_{d,i}^{[\mathrm{U}]}(k_i) - Q_{d,i}^{[\mathrm{L}]}(k_i) \tag{3.37}$$

$$\Delta_{f,i}(k_i) \triangleq Q_{f,i}^{[\mathrm{U}]}(k_i) - Q_{f,i}^{[\mathrm{L}]}(k_i) \tag{3.38}$$

Based on (3.33) and (3.34), we can see that $\Delta_{d,i}(k_i) = \Delta_{f,i}(k_i)$. Moreover, we can obtain the following properties about $Q_{d,i}^{[\mathrm{L}]}(k_i)$, $Q_{d,i}^{[\mathrm{U}]}(k_i)$, and $\Delta_{d,i}(k_i)$. It is worth noting that $Q_{f,i}^{[\mathrm{L}]}(k_i) = Q_{d,i}^{[\mathrm{L}]}(k_i)$, $Q_{f,i}^{[\mathrm{U}]}(k_i) = Q_{d,i}^{[\mathrm{U}]}(k_i)$, and $\Delta_{f,i}(k_i) = \Delta_{d,i}(k_i)$. Therefore, all the following properties hold for $Q_{f,i}^{[\mathrm{L}]}(k_i)$, $Q_{f,i}^{[\mathrm{U}]}(k_i)$ and $\Delta_{f,i}(k_i)$, and we do not need to repeat the properties and proofs.

Property 5. *For fixed partner set $\mathcal{J}^{(i)}$, and decision fusion threshold k_i, $Q_{d,i}^{[\mathrm{L}]}(k_i)$, $Q_{d,i}^{[\mathrm{U}]}(k_i)$, and $\Delta_{d,i}(k_i)$, only depend on the BEP of each reporting channel, $\varepsilon_{j,i}$.*

Proof. It can be seen directly from the expressions of $Q_{d,i}^{[\mathrm{L}]}(k_i)$, $Q_{d,i}^{[\mathrm{U}]}(k_i)$, and $\Delta_{d,i}(k_i)$ as shown in (3.33), (3.34) and (3.37). □

Property 6. *For fixed partner set $\mathcal{J}^{(i)}$, and reporting channel BEP $\varepsilon_{j,i}$, $Q_{d,i}^{[\mathrm{L}]}(k_i)$, and $Q_{d,i}^{[\mathrm{U}]}(k_i)$ all decrease with increasing k_i.*

Proof. For any $k_{i,1} < k_{i,2}$, using (3.33) and (3.34), we get

$$Q_{d,i}^{[\mathrm{L}]}(k_{i,1}) - Q_{d,i}^{[\mathrm{L}]}(k_{i,2}) = \sum_{m=k_{i,1}}^{k_{i,2}-1} \sum_{l=1}^{\binom{n_i}{m}} \left(\prod_{s_j \in \mathcal{J}_{m,l}^{(i)}} \varepsilon_{j,i} \prod_{s_j \in \mathcal{J}^{(i)} \setminus \mathcal{J}_{m,l}^{(i)}} (1 - \varepsilon_{j,i}) \right) > 0$$

$$Q_{d,i}^{[\mathrm{U}]}(k_{i,1}) - Q_{d,i}^{[\mathrm{U}]}(k_{i,2}) = \sum_{m=k_{i,1}}^{k_{i,2}-1} \sum_{l=1}^{\binom{n_i}{m}} \left(\prod_{s_j \in \mathcal{J}_{m,l}^{(i)}} (1 - \varepsilon_{j,i}) \prod_{s_j \in \mathcal{J}^{(i)} \setminus \mathcal{J}_{m,l}^{(i)}} \varepsilon_{j,i} \right) > 0.$$

□

Property 7. $\Delta_{d,i}(k_i) = \Delta_{d,i}(n_i - k_i + 1), \quad 1 \leqslant k_i \leqslant n_i.$

Proof. See Appendix C for the proof of Property 7. □

Chapter 4

Spatial Reuse MAC Protocol based on Hybrid TDMA/OFDMA

4.1 MAC Design Concept

4.1.1 Basic Concept of Spatial Reuse

The MAC protocol considered in this work is based on hybrid TDMA/OFDMA since it is simple and collision-free. It is structured as a TDMA frame having S time slots and C orthogonal sub-carriers as illustrated in Fig. 4.1. We assume that the frame length and sub-carrier bandwidth are fixed throughout system operation, and the channel resource is quantified by the number of time-frequency (T-F) slots occupied.

An intuitive approach is to schedule each sensor into a particular T-F slot as in [36]. Despite its concise implementation form and conflict-free property, such a scheme is unrealistic since the number of required T-F slots will increase linearly with the number of sensors. This problem can be solved by introducing the concept of *spatial reuse*, which allows a transmitter to share a slot with other far enough transmitters. For example, in [37] the MAC protocol reduces the number of slots by dividing sensors into groups which can communicate simultaneously.

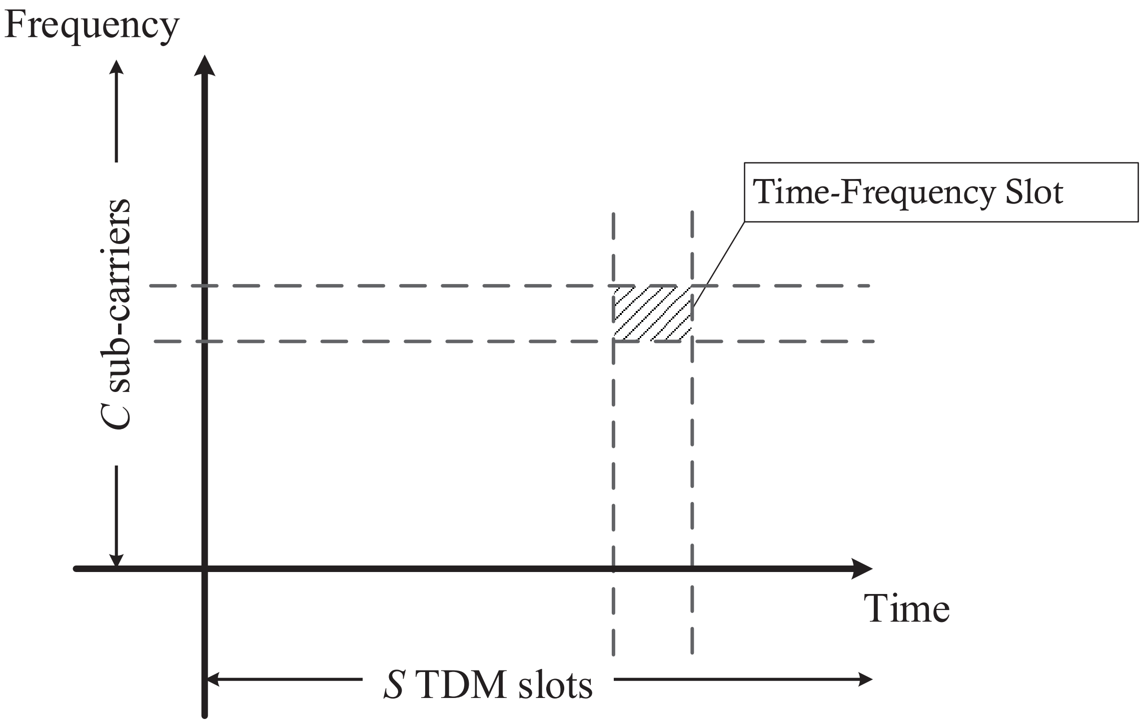

Fig. 4.1 MAC structure based on hybrid TDMA/OFDMA.

The basic concept of spatial reuse is utilized in the sensor scheduling procedure of our MAC protocol. The objective is to divide the N sensors, s_1, s_2, $\dots$, s_N, into M separate sets, and assign one T-F slot to the sensors in the same set. The scheduling result is defined as an N-dimensional vector $\mathbf{p} = (p_1, p_2, \dots, p_N)$, whose component $p_i \in \{1, 2, \dots, M\}$ $(1 \leqslant i \leqslant N)$, represents the ID of the T-F slot assigned to s_i.

4.1.2 Avoidance of Repetition

Since the T-F slots in our system are assumed to have identical frame length as well as sub-carrier bandwidth, we only need to care about the T-F slot sharing relationship between sensors, rather than a sensor's specific slot ID. For instance, if $N = 6$ and $M = 3$, the following three scheduling results, $\mathbf{p} = (1, 1, 2, 3, 2, 1)$, $\mathbf{p}' = (2, 2, 1, 3, 1, 2)$, and $\mathbf{p}'' =$

$(3,3,1,2,1,3)$ are actually the same in our system. In order to avoid such repetition, we introduce the following restriction for the scheduling result vector $\mathbf{p} = (p_1, p_2, \ldots, p_N)$,

$$\begin{aligned} &\forall m_1 < m_2, \quad m_1, m_2 = 1, 2, \ldots, M : \\ &\min\{i \in \mathbb{Z}^+ \,|\, p_i = m_1\} < \min\{i \in \mathbb{Z}^+ \,|\, p_i = m_2\} \end{aligned} \tag{4.1}$$

where $\{i \in \mathbb{Z}^+ \,|\, p_i = m_1\}$ is the set containing the indices of p_i whose value is m_1, and $\min\{i \in \mathbb{Z}^+ \,|\, p_i = m_1\}$ represents the smallest index among them. Take $\mathbf{p} = (1,1,2,3,2,1)$ for example. $\{i \in \mathbb{Z}^+ \,|\, p_i = 2\} = \{3,5\}$, and $\min\{i \in \mathbb{Z}^+ \,|\, p_i = 2\} = 3$. Then, (4.1) means that if $m_1 < m_2$, the smallest index in the set $\{i \in \mathbb{Z}^+ \,|\, p_i = m_2\}$ is less than the smallest index in the set $\{i \in \mathbb{Z}^+ \,|\, p_i = m_1\}$. Take $\mathbf{p}' = (2,2,1,3,1,2)$ for example. We find that $\min\{i \in \mathbb{Z}^+ \,|\, p_i = 1\} = 3 > \min\{i \in \mathbb{Z}^+ \,|\, p_i = 2\} = 1$, which does not satisfy condition (4.1). In addition, for $\mathbf{p} = (1,1,2,3,2,1)$, we have

$$\min\{i \in \mathbb{Z}^+ \,|\, p_i = 1\} = 1 < \min\{i \in \mathbb{Z}^+ \,|\, p_i = 2\} = 3 < \min\{i \in \mathbb{Z}^+ \,|\, p_i = 3\} = 4.$$

Thus, $\mathbf{p} = (1,1,2,3,2,1)$ satisfies condition (4.1).

4.1.3 Two Types of Conflicts

When s_i and s_j are scheduled into the same slot, there may exist two types of conflicts. A *primary conflict* occurs when there is a communication link between them, or they have a common cooperating *recipient*, which must be avoided. We form an $N \times N$ *adjacency matrix* $\mathbf{R}$ to characterize the cooperation relationship between sensors as explained in Section 2.1, and then build an $N \times N$ *conflict relation matrix* $\mathbf{F}$ with the element

$$F_{i,j} = \begin{cases} 1 & \text{if} \quad (i \neq j) \wedge (R_{i,j} = 1 \vee R_{j,i} = 1 \vee \exists k : R_{i,k} = R_{j,k} = 1) \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

where $\wedge$ and $\vee$ denote the logical conjunction and logical disjunction respectively, $i \neq j$ means a sensor cannot have conflict with itself (i.e. $F_{i,i} \neq 1$), $R_{i,j} = 1 \vee R_{j,i} = 1$ means there is a communication link between two different sensors, and $\exists k : R_{i,k} = R_{j,k} = 1$ means s_i and s_j have a common cooperating *recipient* s_k. Therefore, when two different sensors s_i and s_j share the same T-F slot, (4.2) means that a *primary conflict* occurs between two different sensors s_i and s_j (i.e. $F_{i,j} = 1, i \neq j$), if there exists

(1) a communication link from s_i to s_j (i.e. $R_{i,j} = 1$) **or**

(2) a communication link from s_f to s_i (i.e. $R_{j,i} = 1$) **or**

(3) a common cooperating *recipient* s_k of s_i and s_j (i.e. $\exists k : R_{i,k} = R_{j,k} = 1$).

In other cases, there exists a *secondary conflict*, which can be permitted if the impact caused by mutual interference is acceptable. Given that s_j is a partner of s_i, i.e. $R_{j,i} = 1$, when s_j broadcasts its package, other sensors who share the same T-F slot with s_j become the interference sources. We define a *distance matrix* $\mathbf{D}$, whose element $d_{i,j}$ stores the Euclidean distance between s_i and s_j. According to (2.3) and (2.6), the average received SINR of the transmission from s_j to s_i is

$$\bar{\gamma}_{j,i}(\mathbf{p}) = \frac{A d_{j,i}^{-\mu} P_{\text{tra}}^{(j)}}{P_{\text{noi}} + \sum_{k \in \{k \in \mathbb{Z}^+ \,|\, p_k = p_j \,\wedge\, k \neq j\}} A d_{k,i}^{-\mu} P_{\text{tra}}^{(k)}} \tag{4.3}$$

where $\mathbf{p}$ represents the scheduling result, which is an N-dimensional vector $\mathbf{p} = (p_1, p_2, \ldots, p_N)$, whose component $p_i \in \{1, 2, \ldots, M\}$ $(1 \leqslant i \leqslant N)$, represents the ID of the T-F slot assigned to s_i.

Then, we can continue to calculate the BEP $\varepsilon_{j,i}(\mathbf{p})$ on the communication link $s_j \rightarrow s_i$, using (3.14)

$$\varepsilon_{j,i}(\mathbf{p}) = \frac{1}{\pi} \int_0^{\pi/2} \frac{(1+K)\sin^2\phi}{(1+K)\sin^2\phi + \bar{\gamma}_{j,i}(\mathbf{p})} \exp\left[-\frac{K\bar{\gamma}_{j,i}(\mathbf{p})}{(1+K)\sin^2\phi + \bar{\gamma}_{j,i}(\mathbf{p})}\right] d\phi. \tag{4.4}$$

4.1.4 Problem Formulation

Firstly, we explain how to determine the distance matrix **D** and adjacency matrix **R**. The distance matrix **D** is determined by the locations of the sensors, which are assumed to be fixed in our work. The sensors should be spatially separated in the target area, such that a SU at any location in the target area can get access to the spectrum information around it. On one hand, if the cooperating sensors are located too close to each other, they will experience spatially correlated shadowing. [19, 23] analyse the degradation on spectrum detection performance caused by correlated shadowing, showing that the distance between sensors should be larger than a decorrelation distance. On the other hand, the sensors cannot be deployed too far from each other to avoid a high BEP on each reporting channel and compromise the cooperative spectrum sensing performance as discussed in last chapter. In addition, there may be more restrictions on sensors' locations in practice. For example, when the WSN is deployed by a service provider, the sensors may be installed in existing base stations, where the locations cannot be changed.

The adjacency matrix **R** characterizes the cooperating relationship between sensors, and we have shown that cooperative spectrum sensing performance is degraded by the BEP of communication links. Thus, if the distance between sensors are sufficient large to avoid severe correlated shadowing, a sensor s_i always chooses the cooperating partners with the lowest reporting channel BEP first. As we cannot decide the scheduling of T-F slots before **R** is determined, we do not consider the mutual interference between sensors, and only consider the thermal noise when setting up **R**. If all the sensors use the same transmit power, and experience identical multipath fading, a sensor always chooses the nearest sensors around it as cooperating partners in our system. Moreover, the number of partners of each sensor should be similar, such that the spectrum sensing performance does not vary a lot between different sensors. It is worth noting that **R** can be changed in the network design process. We can set it as an input of the MAC protocol, decide the spatial reuse of T-F slots, and then adjust **R** according to the network spectrum sensing performance (e.g. increase or decrease the number of partners of each sensor). In Chapter 5, we will discuss how to set up $\mathbf{R}$ in a grid network.

Given a certain number of slots M, our spatial reuse MAC protocol aims at providing a partition of the N sensors to avoid any *primary conflict*, and minimize the degradation caused by the BEPs on communication links between sensors. We focus on maximizing the *Achievable Range* of probability of detection or false alarm, $\Delta_{d,i}(k_i, \mathbf{p}) = \Delta_{f,i}(k_i, \mathbf{p})$, as defined in (3.37) and (3.38), which characterizes the reporting error effect, and doesn't depend on the local spectrum sensing quality. The second argument $\mathbf{p}$ is added because $\varepsilon_{j,i}$ in (3.37) and (3.38) will change over $\mathbf{p}$. Since $\Delta_{d,i}(k_i, \mathbf{p}) = \Delta_{f,i}(k_i, \mathbf{p})$, in the rest of the thesis, we omit the first subscript d or f, and just use $\Delta_i(k_i, \mathbf{p})$ as the notation.

The *OR* fusion rule ($k_i = 1$) and *AND* fusion rule ($k_i = n_i$) are two special cases with the same *Achievable Range* (from Property 7), which can be obtained from (3.35) and (3.36)

$$\Delta_i(1, \mathbf{p}) = \Delta_i(n_i, \mathbf{p}) = \prod_{s_j \in \{s_j | R_{j,i}=1\}} \big(1 - \varepsilon_{j,i}(\mathbf{p})\big) - \prod_{s_j \in \{s_j | R_{j,i}=1\}} \varepsilon_{j,i}(\mathbf{p}). \tag{4.5}$$

From the network perspective, since the number of *partners* of each sensor may not be identical, we first define the network *fusion factor* Ω to set the fusion threshold k_i of each sensor, as $k_i = \lceil \Omega n_i \rceil$, where $n_i = |\{s_j | R_{j,i} = 1\}|$ is the number of *partners* of s_i. Then, averaging $\Delta_i(k_i, \mathbf{p})$ over sensor locations, and according to (3.33), (3.34) and (3.37), we obtain the following *Network Achievable Range* of probability of detection or false alarm:

$$\overline{\Delta}(\Omega, \mathbf{p}) = \frac{1}{N} \sum_{i=1}^{N} \Delta_i(k_i, \mathbf{p}) \tag{4.6}$$

where

$$n_i = |\{s_j | R_{j,i} = 1\}|, \quad k_i = \lceil \Omega n_i \rceil$$

$$\Delta_i(k_i, \mathbf{p}) = \sum_{m=k_i}^{n_i} \sum_{l=1}^{\binom{n_i}{m}} \left(\prod_{s_j \in \mathcal{J}^{(i)}_{m,l}} \varepsilon_{j,i}(\mathbf{p}) \prod_{s_j \in \mathcal{J}^{(i)} \setminus \mathcal{J}^{(i)}_{m,l}} \big(1 - \varepsilon_{j,i}(\mathbf{p})\big) - \prod_{s_j \in \mathcal{J}^{(i)}_{m,l}} \big(1 - \varepsilon_{j,i}(\mathbf{p})\big) \prod_{s_j \in \mathcal{J}^{(i)} \setminus \mathcal{J}^{(i)}_{m,l}} \varepsilon_{j,i}(\mathbf{p}) \right). \tag{4.7}$$

Therefore, with the *adjacency matrix* and *distance matrix* of a DetF WSN with N sensors given as $\mathbf{R}$ and $\mathbf{D}$ respectively, the number of T-F slots fixed at M, and the network *fusion factor* chosen as Ω, the spatial reuse MAC protocol can be formulated as the solution of the following combinatorial optimization problem:

$$\max_{\mathbf{p}} \ \overline{\Delta}(\Omega, \mathbf{p}) \tag{4.8}$$

or equivalently

$$\min_{\mathbf{p}} \ \big(1 - \overline{\Delta}(\Omega, \mathbf{p})\big) \tag{4.9}$$

s.t.

$$\mathbf{p} = (p_1, p_2, \ldots, p_N), \text{ with } p_i \in \{1, 2, \ldots, M\} \tag{4.10}$$

$$\forall (i,j) \in \{(i,j) \,|\, p_i = p_j\} : F_{i,j} = F_{j,i} = 0 \tag{4.11}$$

$$\forall m_1 < m_2, \quad m_1, m_2 = 1, 2, \ldots, M :$$

$$\min\{i \in \mathbb{Z}^+ \,|\, p_i = m_1\} < \min\{i \in \mathbb{Z}^+ \,|\, p_i = m_2\}. \tag{4.12}$$

where

- (4.9) is the equivalent cost function to (4.8);
- (4.10) is the scheduling result $\mathbf{p}$ defined in Section 4.1.1, which is used to compute $\varepsilon_{j,i}(\mathbf{p})$ through (4.3) and (4.4);

- (4.11) is the constraint for *primary conflicts*. $F_{i,j}$ is the element in the *conflict relation matrix* $\mathbf{F}$ as defined in (4.2), and $F_{i,j} = 0$ means that there is no *primary conflict* if s_i and s_j are scheduled into the same T-F slot (i.e. $p_i = p_j$). Thus, constraint (4.11) eliminates *primary conflicts* as explained in Section 4.1.3;

- (4.12) is the constraint for *repetition*, which is defined in (4.1). For any two T-F slot ID $m_1 < m_2, \quad m_1, m_2 = 1, 2, \ldots, M$, the smallest index in the set $\{i \in \mathbb{Z}^+ \,|\, p_i = m_2\}$ is less than the smallest index in the set $\{i \in \mathbb{Z}^+ \,|\, p_i = m_1\}$, where p_i is the ith element of the scheduling result $\mathbf{p}$ in (4.10). Thus, constraint (4.12) eliminates *repetition* as explained in Section 4.1.2.

If a scheduling result satisfies constraint (4.11), it is defined as a *feasible* solution. If it satisfies both (4.11) and (4.12), it is called a *valid* solution. Moreover, we define the cost function as

$$\zeta(\mathbf{p}) = 1 - \overline{\Delta}(\Omega, \mathbf{p}) \tag{4.13}$$

which is also called the *Network Achievable Range Loss* of probability of detection or false alarm.

The optimization problem specified in (4.8) is a combinatorial optimization problem, which is usually computationally intractable when related to practice [38]. Many combinatorial optimization problems of scheduling or channel assignment in wireless networks using similar channel models are too complicated to find an exact algorithm. Thus, in [39–44], efforts have been spent in designing heuristic algorithms to solve various optimization objectives. The heuristic approach aims to efficiently explore the search space in order to find a high-quality solution, and the use of it in combinatorial optimization is introduced in [45, 46]. Therefore, we also focus on heuristic methods to solve the combinatorial optimization problem specified in (4.8). Common heuristic methods for solving such combinatorial optimization problem include: *Tabu Search*, *Genetic Algorithms*, *Simulated Annealing*, *Ant System*, *Neural Networks*, etc [45, 47]. In the next two sections, we consider two approaches to solving (4.9): Greedy Algorithm and Simulated Annealing.

4.2 Spatial Reuse by Graph Colouring and Greedy Algorithm

4.2.1 Initialization via Vertex Colouring

We first form the *conflict relation matrix* $\mathbf{F}$ based on the given *adjacency matrix* $\mathbf{R}$. Then all the sensors are considered as vertices, and an edge is drawn between s_i and s_j if $F_{i,j} = 1$ or $F_{j,i} = 1$. Thus, a simple undirected *conflict graph* is constructed, and finding a *feasible* solution $\mathbf{p}$ can be viewed as assigning M labels, traditionally called "colours", to the vertices of this *conflict graph*, such that adjacent vertices have different colours. We choose *DSatur* algorithm (a heuristic vertex colouring algorithm) proposed in [48] to generate an initial solution.

In order to explain this algorithm, we first introduce two definitions:

i) The *degree* of a vertex is the number of edges incident to this vertex.

ii) The *saturation degree* of a vertex is the number of different colours in the neighbors of a vertex.

The initialization via *DSatur* algorithm is summarised as follows

a) Number the colours sequentially, and assign *colour* 1 to the vertex with the highest *degree*. If there are several sensors with the same highest degree, choose the one with the smallest sensor index.

b) Select the next vertex with the highest *saturation degree*, and in case of a tie, choose the vertex with the highest ordinary *degree*.

c) Search the colours in ascending order, and find the first available one which hasn't been assigned to any neighbour of the current vertex.

d) Assign the colour found in step c) to the current vertex selected in step b).

e) Keep checking the remaining uncoloured vertices in such a dynamic order described by step b)– d) until all the vertices are coloured.

Let M_0 denote the number of colours used in *DSatur* algorithm. It is worth noting that, when $M < M_0$ the problem of determining whether such an M-colouring exists is NP-complete [49], and it is not the major issue of this work. Therefore, we only consider the situation when $M \geqslant M_0$. When all the sensors are coloured, we can obtain a *feasible* initial solution $\mathbf{p}$ by setting p_i to s_i's colour index. This initialization method through *DSatur*, or any other heuristic vertex colouring algorithm, is applicable to any network form; nevertheless, if the *adjacency matrix* or *conflict matrix* has a certain regular structure, we can find other more efficient deterministic algorithms for initialization. An example will be given in Section 5.2.

4.2.2 Realignment Procedure

Since only the constraint of primary conflicts – (4.11), is considered in the vertex colouring phase, the initial solution generated may go against the constraint of repetition – (4.12). Thus, we continue to conduct the *realignment* procedure, which will be referred to as Realign($\mathbf{p}$) throughout the rest of this thesis, to transform a *feasible* solution $\mathbf{p} \in \mathbb{R}^N$ into a *valid* solution:

a) Initialization: $x \leftarrow 0$, $y \leftarrow 1$, and $\tilde{\mathbf{p}} \leftarrow \mathbf{p}$;

b) Find the smallest index i satisfying $\tilde{p}_i \neq 0$, and assign the value of $\tilde{p}_i$ to x, i.e. $i \leftarrow \min\{i \in \mathbb{Z}^+ \,|\, \tilde{p}_i \neq 0\}$;

c) For all the indices $j \in \mathbb{Z}^+$ satisfying $\tilde{p}_j = x$, set p_j to y, and then set $\tilde{p}_j$ to 0;

d) $y \leftarrow y + 1$;

e) Repeat b) – d) until $\tilde{\mathbf{p}} = \mathbf{0}$, and then output $\mathbf{p}$, which is a *valid* solution.

4.2.3 Complete Greedy Algorithm

Let $\mathcal{N}(\mathbf{p})$ denotes the *neighbourhood* of a *valid* solution $\mathbf{p}$, which is defined as the set of *valid* solutions obtained from $\mathbf{p}$ by performing a local change. The rule for *neighbourhood* generation can be chosen from a variety of existing methods provided in [50]. For a current *valid* solution $\mathbf{p}$, our greedy algorithm changes the value of only one of its N components to get a new solution $\mathbf{p}'$, and thus there are $N(M-1)$ possible results. If $\mathbf{p}'$ violates the constraint of primary conflicts, discard it. Otherwise, the *realignment* procedure is conducted, and the outcome becomes one element of $\mathcal{N}(\mathbf{p})$.

Starting from the initial solution, the greedy algorithm always selects $\hat{\mathbf{p}}$ from $\mathcal{N}(\mathbf{p})$ resulting in the minimum value of the cost function $\zeta(\mathbf{p}')$ in (4.13). If $\zeta(\hat{\mathbf{p}}) < \zeta(\mathbf{p})$, $\hat{\mathbf{p}}$ is accepted as the new *valid* solution for the next state, and this procedure is repeated. If $\zeta(\hat{\mathbf{p}}) \geqslant \zeta(\mathbf{p})$, the algorithm is terminated, and $\mathbf{p}$ is output as the final result. In a word, the greedy algorithm keeps searching in the steepest descent direction of the cost value, and terminates at a solution whose *neighbourhood* cannot offer any decrease in cost. The complete procedure is summarized in Fig. 4.2.

```
Algorithm 1: Greedy Algorithm for Spatial Reuse
Input:  Rician fading parameter: $K$
        Path-loss parameters: $A$ and $\mu$
        Thermal noise power: $P_{\mathrm{noi}}$
        Each sensor's transmit power: $P_{\mathrm{tra}}^{(i)}$
        Number of sensors: $N$
        Number of T-F slots: $M$
        Adjacency and distance matrix $\mathbf{R}$ and $\mathbf{D}$
Output: A valid scheduling result $\mathbf{p} \in \mathbb{R}^N$
Build the conflict matrix $\mathbf{F}$ based on $\mathbf{R}$
Obtain the initial solution $\hat{\mathbf{p}}$ via vertex colouring, and conduct the realignment procedure $\hat{\mathbf{p}} \leftarrow$ REALIGN($\hat{\mathbf{p}}$)
repeat
    $\mathbf{p} \leftarrow \hat{\mathbf{p}}$
    Calculate the cost value $\zeta(\mathbf{p})$
    Generate the neighbourhood $\mathcal{N}(\mathbf{p})$ through 1-exchange and realignment procedure
    $\hat{\mathbf{p}} \leftarrow \arg\min_{\mathbf{p}' \in \mathcal{N}(\mathbf{p})} \zeta(\mathbf{p}')$
until $\zeta(\hat{\mathbf{p}}) \geqslant \zeta(\mathbf{p})$
return $\mathbf{p}$
```

Fig. 4.2 Greedy Algorithm for Spatial Reuse

4.3 Spatial Reuse by Adaptive Simulated Annealing Algorithm

4.3.1 Fundamentals of Simulated Annealing

Although the greedy algorithm is easy to implement and has low complexity, it cannot guarantee a globally optimal solution. Therefore, we propose an alternative approach to the combinatorial optimization problem via *Adaptive Simulated Annealing* (ASA) [51]. *Simulated Annealing* is a probabilistic method whose random search process mimics the physical annealing process [52]. It originates from the principles of thermodynamics making a metal "freeze" into a crystalline structure at a minimum energy configuration, when cooled slowly from a state of high temperature. If its cooling process is well controlled, the algorithm can avoid being trapped in local minima, and converges to a global optimum.

The ASA algorithm permits an exponentially decreasing annealing schedule which greatly accelerates the optimization process compared with some traditional simulated annealing algorithms (e.g. *Boltzmann Annealing* and *Fast Cauchy Annealing* [53]). Moreover, the re-annealing procedure is introduced in the ASA to adapt to changing sensitivities in the multi-dimensional parameter-space. More details on ASA are provided in Appendix D.

4.3.2 Modifications in ASA

As illustrated in Appendix D, the role of the *generating* procedure is to generate the *candidate* solution $\hat{\mathbf{p}}^{(t+1)}$ for the $(t+1)$th state, from the solution $\mathbf{p}^{(t)}$ accepted at the tth state. Since the components of $\mathbf{p}$ can take only integer values, and interrelate with each other subject to the constraint of primary conflicts, the *generating* procedure needs to be modified. To be specific, at the beginning of the $(t+1)$th state, we form a random permutation of the indices $\{1, 2, \ldots, N\}$, which is denoted by a vector $\boldsymbol{\sigma}$. Then, at the ath step, we pick the index $i = \sigma_a$, and obtain the ith component of the *temporary candidate* solution,

$$\check{p}_i^{(t+1)} = p_i^{(t)} + y_i(M-1), \tag{4.14}$$

where $y_i \in [-1, 1]$ is a sample of a random variable Y_i with the PDF shown in (D.2). The range of the ith parameter is denoted by $[L_i, U_i]$, i.e. $L_i \leqslant p_i^{(t)} \leqslant U_i$. If $\check{p}_i^{(t+1)} \notin [L_i, U_i]$, another sample y_i of Y_i is applied until $\check{p}_i^{(t+1)} \in [L_i, U_i]$.

Next, we find $\hat{p}_i^{(t+1)}$ as follows

$$\hat{p}_i^{(t+1)} = \underset{m \in \{1,2,\ldots,M\}}{\arg\min} |\check{p}_i^{(t+1)} - m|, \tag{4.15}$$

$$\text{s.t. } \forall j \in \left\{ j \,|\, j \in \{\sigma_1, \sigma_2, \ldots, \sigma_{a-1}\} \text{ and } \hat{p}_j^{(t+1)} = m \right\} : \; F_{i,j} = F_{j,i} = 0, \tag{4.16}$$

where $\left\{ j \,|\, j \in \{\sigma_1, \sigma_2, \ldots, \sigma_{a-1}\} \text{ and } \hat{p}_j^{(t+1)} = m \right\}$ is the set of all the sensors that have been scheduled into the mth T-F slot before s_i, and thus (4.16) ensures that there is no primary conflict between s_i and any other sensors scheduled before it.

Therefore, we can generate the components of $\hat{\mathbf{p}}^{(t+1)}$ one by one according to the random order indicated by $\boldsymbol{\sigma}$ ($\boldsymbol{\sigma}$ is a random permutation of the indices $\{1, 2, \ldots, N\}$ formed at the beginning of each stage as explained in the first paragraph of this subsection), and finally obtain a *feasible candidate* solution. If there is no valid value for $\hat{p}_i^{(t+1)}$ at any step, the whole state will be started over with a new permutation vector $\boldsymbol{\sigma}$. At the end of the *generating* procedure, we conduct the *realignment* procedure on $\hat{\mathbf{p}}^{(t+1)}$, to transform it to a *valid* solution.

4.3.3 Complete ASA Algorithm

The *acceptance* and *annealing* procedures are the same as those illustrated in Appendix D. However, as to the optional *reannealing* procedure, we bypass the *reannealing* for *parameter temperatures*, and only conduct this procedure on *cost temperatures*. For one thing, since the components of $\mathbf{p}$ are integers and mutually restrictive, there is no effective numerical method to calculate the derivatives f_i in (D.7). For another, for relatively large numbers of parameters (N), skipping this procedure helps to significantly increase operational efficiency.

We can use the same initialization method of the greedy algorithm (Section 4.2.1), or conduct the *generating* procedure on a random solution, to get the initial value of $\mathbf{p}$. The ASA algorithm exits normally when the predefined maximum number of generated states (N^*_{gen}) or maximum number of accepted states (N^*_{acc}) is reached. The complete procedure of the ASA algorithm is summarized in Fig. 4.3.3. For the simplicity of illustration, we define a *ASA parameter set* $\mathcal{P}_{\text{ASA}}$ containing all the predefined ASA parameter values, such as *initial parameter temperature* $T_i(0)$, *parameter temperature scale* c_{para}, and so on.

```
Algorithm 2: ASA Algorithm for Spatial Reuse
Input:  Rician fading parameter: $K$
        Path-loss parameters: $A$ and $\mu$
        Thermal noise power: $P_{\text{noi}}$
        Each sensor's transmit power: $P_{\text{tra}}^{(i)}$
        Number of sensors: $N$
        Number of T-F slots: $M$
        Adjacency and distance matrix: $\mathbf{R}$ and $\mathbf{D}$
        ASA parameter set: $\mathcal{P}_{\text{ASA}}$
Output: A valid scheduling result $\mathbf{p} \in \mathbb{R}^N$
 1: Build the conflict matrix $\mathbf{F}$ based on $\mathbf{R}$
 2: Generate an initial solution $\mathbf{p}$, and calculate $\zeta(\mathbf{p})$
 3: Sample the parameter space to calculate the initial cost temperature $T_{\text{cost}}(0)$.
 4: $N_{\text{gen}} \leftarrow 0$, $N_{\text{acc}} \leftarrow 0$
 5: while $N_{\text{gen}} < N_{\text{gen}}^*$ and $N_{\text{acc}} < N_{\text{acc}}^*$ do
 6:     Form a random sorting vector $\boldsymbol{\sigma}$
 7:     Generate the components of the candidate solution $\hat{\mathbf{p}}$ according to the order indicated by $\boldsymbol{\sigma}$ (See Section 4.3.2 for details). Whenever a component fails to find any valid value, go to Step 6.
 8:     $\hat{\mathbf{p}} \leftarrow$ Realign($\hat{\mathbf{p}}$)
 9:     $N_{\text{gen}} \leftarrow N_{\text{gen}} + 1$
10:     Conduct acceptance procedure represented by (D.3) to determine whether $\mathbf{p} \leftarrow \hat{\mathbf{p}}$.
11:     If $\hat{\mathbf{p}}$ is accepted, $N_{\text{acc}} \leftarrow N_{\text{acc}} + 1$
12:     Conduct annealing procedure to update $T_i$ and $T_{\text{cost}}$ as shown in (D.4) and (D.6).
13:     if $N_{\text{gen}} = zm_{\text{gen}}$ or $N_{\text{acc}} = zm_{\text{acc}} (z \in \mathbb{Z}^+)$ then
14:         Conduct reannealing procedure to rescale the cost temperature as shown in (D.9)–(D.11)
15:     end if
16: end while
17: return $\mathbf{p}$
```

Fig. 4.3 ASA Algorithm for Spatial Reuse

Chapter 5

Performance Results for a Grid Network

5.1 Scenario Setup

The sensors are assumed to be deployed on the grid of a target square region, and the Euclidean distance between two nearest sensors is fixed as L_0. We establish a Cartesian coordinate plane such that $(0, 0)$ is in the centre of the square, and the coordinates of the square's vertices are

$$\begin{gathered}(-\sqrt{N}L_0/2, \sqrt{N}L_0/2), \quad (\sqrt{N}L_0/2, \sqrt{N}L_0/2), \\ (\sqrt{N}L_0/2, -\sqrt{N}L_0/2), \text{ and } (-\sqrt{N}L_0/2, -\sqrt{N}L_0/2)\end{gathered}$$

where N is the number of sensors. The coordinates of sensors are

$$x_i = \Big(a + \frac{1-\sqrt{N}}{2}\Big)L_0, \quad y_i = \Big(b + \frac{1-\sqrt{N}}{2}\Big)L_0 \tag{5.1}$$

where $a, b \in \{0, 1, \ldots, \sqrt{N} - 1\}$, and $i = 1 + a + b\sqrt{N}$. An example of the deployment of sensors when $N = 16$ is shown in Fig. 5.1.

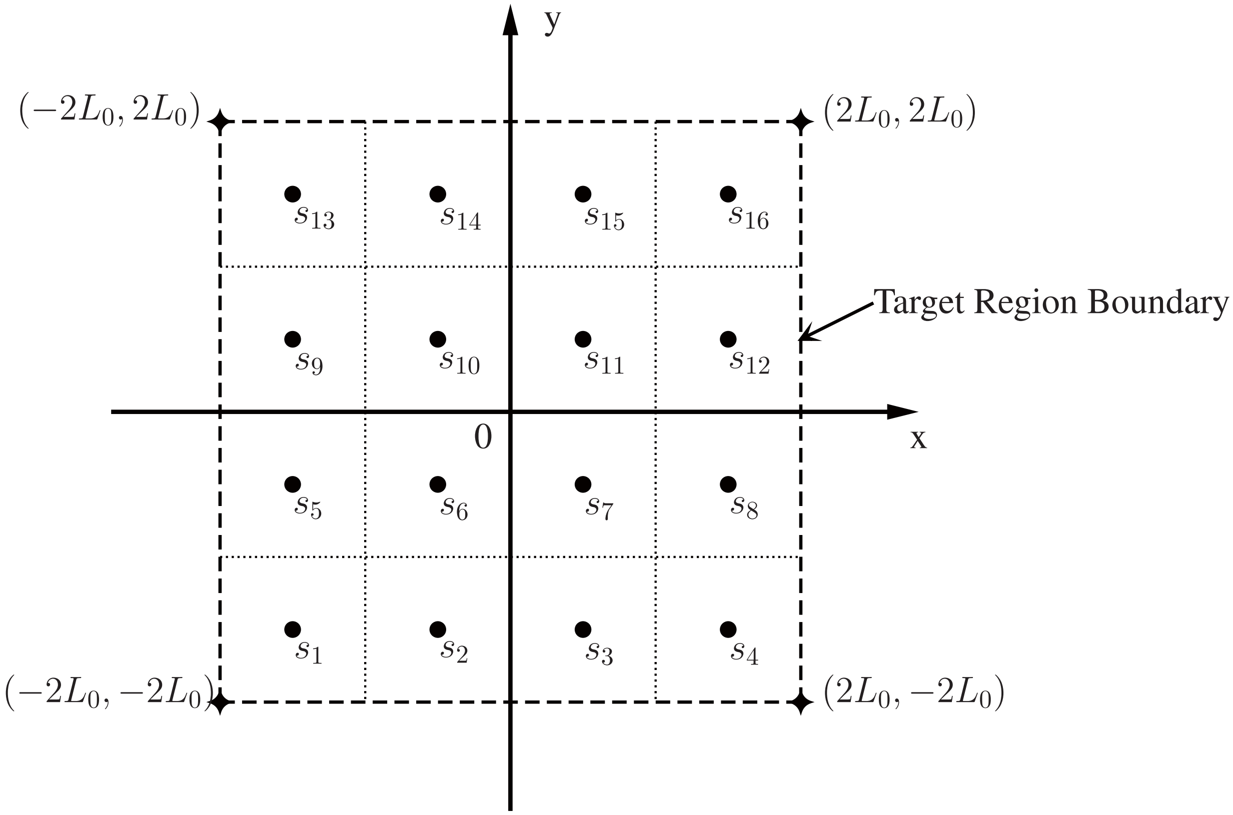

Fig. 5.1 An example of sensor's deployment when $N = 16$.

As to the *partner selection* scheme, the priority is always given to the nearest sensors. We consider four *adjacency matrix* structures, and use the following *cooperation levels* to identify them:

i) *CL0*: No cooperation between sensors, and thus **R** is an identity matrix.

ii) *CL2*: A sensor selects itself and two other nearest sensors as its partners. If there are several sensors with the same distance, select in the ascending order of there sensor IDs.

iii) *CL4*: A sensor selects all the sensors within L_0 as its partners (including itself).

iv) *CL8*: A sensor selects all the sensors within $\sqrt{2}L_0$ as its partners (including itself).

Actually, this type of *partner selection* scheme can be extended to other *cooperation levels* where each sensor has more number of *partners*, which are not listed here.

We choose the COST-Walfish-Ikegami (COST-WI) model as the empirical path-loss model in the WSN, with parameters set for the Urban Macro scenario [54]. It can be viewed as an instance of the simplified path-loss model in (2.3), with $A = 10^{-3.018}$, $\mu = 2.6$, and $20\text{m} \leqslant d \leqslant 5\text{km}$.

Table 5.1 Parameter Settings for Calculating Numerical Results

Parameter Name	Value
Minimum Distance between Sensors (L_0)	50m
Path-loss Constant (A)	$10^{-3.018}$
Path-loss Exponent (μ)	2.6
Rician Fading Parameter (K)	7
Sensor's Transmit Power (P_{tra})	100mW
Communication Subcarrier Bandwidth (W_c)	20kHz
Noise Power Spectral Density ($N_0/2$)	5×10^{-16}W/Hz

The average received SNR of the primary signal, $\bar{\gamma}$, is assumed to be identical at each sensor, which is a reasonable assumption when generating numerical or simulation results as in [18, 55]. Furthermore, each sensor employs the same transmit power P_{tra}, and energy detection threshold λ. The scenario setup parameters are summarized in Table 5.1.

The C-language implementation of the ASA algorithm is based on the source code provided by A. L. Ingber, which is available at <http://www.ingber.com/#ASA> since 1993. We made some modifications as described in Section 4.3.2, and tuned the ASA parameters and options according to the specifics of our system model and optimization problem. The key parameters and options of ASA are summarized in Table 5.2, and others are kept to their default values. In addition, an overview of the software that implements the spatial reuse MAC protocol, and generate the numerical results of spectrum sensing performance presented in this chapter is provided in Appendix E.

Table 5.2 Key Parameters and Options of ASA

Parameter or Option Name	Value
Maximum Number of Generated States (N^*_{gen})	10^7
Maximum Number of Accepted States (N^*_{acc})	10^6
Number of Cost Samples (N_{spl})	5
Temperature Ratio Scale (α_1)	10^{-4}
Temperature Anneal Scale (α_2)	100
Cost Parameter Scale Ratio (β)	1.0
Generated Frequency Modulus (m_{gen})	5000
Accepted Frequency Modulus (m_{acc})	50
Initial Parameter Temperature ($T_i(0)$)	1.0
Parameter Quenching Factor (Q_i)	$N/10$
Cost Quenching Factor (Q_{cost})	$N/10$
Include_Integer_Parameters	TRUE
Reanneal_Parameters	FALSE
QUENCH_PARAMETERS	TRUE
QUENCH_COST	TRUE
QUENCH_PARAMETERS_SCALE	FALSE
QUENCH_COST_SCALE	FALSE

5.2 Features of the Grid Network

We first build an undirected grid graph by considering all the sensors as vertices, and drawing an edge between any two sensors whose Euclidean distance is L_0. The *grid distance* between s_i and s_j is defined as the number of edges in a shortest path connecting them in the grid graph.

- For *CL2* and *CL4*, the *grid distance* between any *partner–receiver* pair is 1. If no two sensors within *grid distance* 2 share the same slot, no primary conflict exists in the WSN.

- For *CL8*, the longest *grid distance* between any *partner–receiver* pair is 2. If no two sensors within *grid distance* 4 share the same slot, no primary conflict exists in the WSN.

If the T-F slots are viewed as "colours", then for *CL2* and *CL4*, a 2-*distance colouring* of the grid graph is a *feasible* solution (no primary conflict), and for *CL8*, a 4-*distance colouring* of the grid graph is a *feasible* solution. Here, the k-*distance colouring* is defined as a vertex colouring of the grid graph such that no two vertices lying at the *grid distance* less than or equal to k are assigned the same colour. Furthermore, the minimum number of colours necessary for the k-*distance colouring* of a grid graph, denoted by χ_k, is given in [56]:

$$\chi_k = \begin{cases} \big((k+1)^2+1\big)/2 & \text{if } k \text{ is even} \\ (k+1)^2/2 & \text{if } k \text{ is odd} \end{cases} \tag{5.2}$$

In summary, for the grid network with the *partner selection* scheme described above, if the longest *grid distance* between any *partner–receiver* pair in the network is κ, then a 2κ-*distance colouring* of the grid graph is a feasible solution (no primary conflict) for the slot scheduling problem. The minimum number of colours (T-F slots) needed to complete such a 2κ-*distance colouring* is $\chi_{2\kappa} = \big((2\kappa+1)^2+1\big)/2$. Thus, if $M \geqslant \big((2\kappa+1)^2+1\big)/2$, we can guarantee that there exists a scheduling solution without primary conflict.

Then, we can set the minimum value of M (number of T-F slots) as $\chi_{2\kappa}$ for the grid network, which is 5 for *CL2* and *CL4*, and 13 for *CL8*. This minimum value is reasonable as it is relatively small compared with the network size, and independent of the number of sensors.

In addition, for the grid network, we introduce an alternative method to generate the initial *valid* solution for the greedy algorithm as well as ASA algorithm, which is based on a deterministic approach of k-*distance colouring* with the following three steps:

a) Find $\hat{k} = \max\{k \in \mathbb{Z}^+ \mid \chi_k \leqslant M\}$, where χ_k is calculated according to (5.2).

b) Obtain an initial solution $\mathbf{p}$ by setting its components as follows:

$$p_i = \begin{cases} a + (\hat{k}+1)b \bmod ((\hat{k}+1)^2+1)/2 & \text{if } \hat{k} \text{ is even} \\ a + \hat{k}b \bmod (\hat{k}+1)^2/2 & \text{if } \hat{k} \text{ is odd} \end{cases} \tag{5.3}$$

where $a, b \in \{0, 1, \ldots, \sqrt{N}-1\}$, $i = 1 + a + b\sqrt{N}$, and "mod" represents the modulo operator.

c) *Realignment* procedure: $\mathbf{p} \leftarrow$REALIGN($\mathbf{p}$)

The scheduling result $\mathbf{p}$ obtained from (5.3) is equivalent to a $\hat{k}$-*distance colouring* of the grid graph, which is proved in [56]. In addition, $\hat{k} \geqslant 2\kappa$ because we have set $M \geqslant \chi_{2\kappa}$. Therefore, the solution $\mathbf{p}$ we get in the second step is a *feasible* one, which ensures that there exists no primary conflict. Compared with the initialization method via *DSatur* colouring algorithm introduced in Section 4.2.1, the approach via *k-distance colouring* is more concise and efficient for a grid network.

5.3 Verification of the Algorithms

In this section, we present an example to show how the greedy algorithm may be trapped in local minima, and how ASA algorithm can jump out and reach the global optimum. We set up a small network with 9 sensors ($N = 9$) as illustrated in Section 5.1, and set the number of T-F slot provided as 5 ($M = 5$). The directed graph is shown in Fig. 5.2. Each sensor has two *partners* other than itself, and the corresponding *adjacency matrix* is

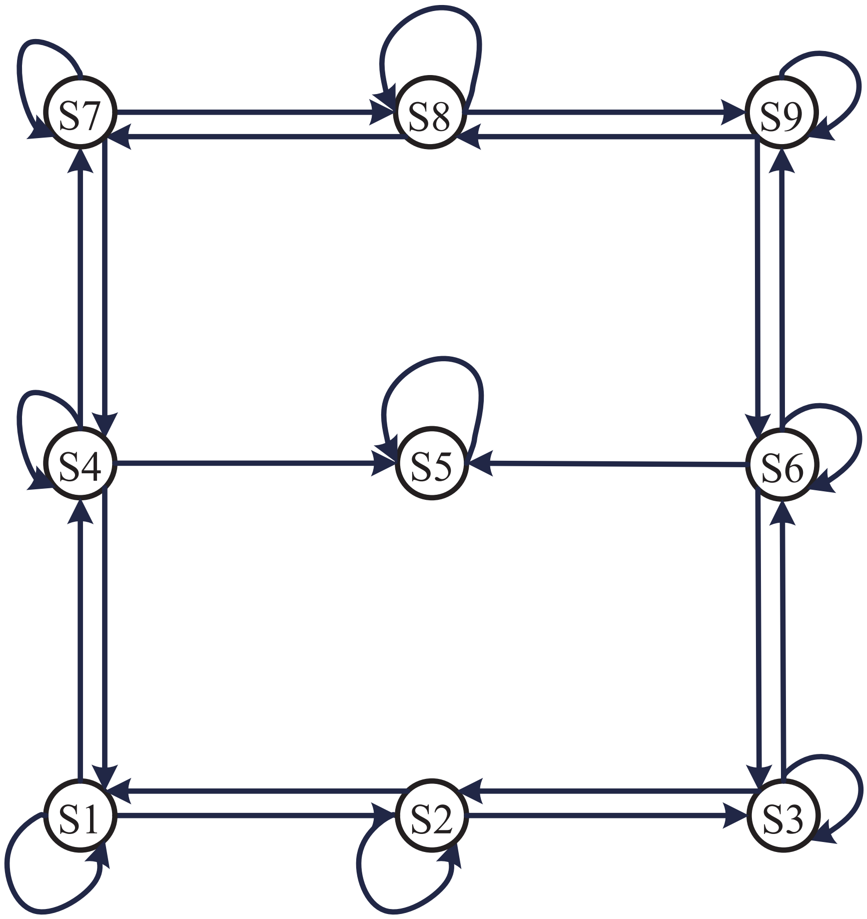

Fig. 5.2 The directed graph of the small network with 9 sensors used in Section 5.3.

constructed as

$$\mathbf{R} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \tag{5.4}$$

The number of *partners* of each sensor $n_i = 3$, and we set the *fusion factor* Ω to be 0.1 or 0.5. Then, the fusion threshold k_i of each sensor is $k_i = \lceil \Omega n_i \rceil = 1$ or 2. When $k_i = 3$, the cost function (4.13) is the same as that of $k_i = 1$, and thus we do not need to use $k_i = 3$.

Through exhaustive search, we find that there are 111 *valid* solutions out of 1935123 *feasible* solutions. When $\Omega = 0.1$, the global minimum is:

$$\zeta(\mathbf{p}^*) = 0.01830289, \text{ with } \mathbf{p}^* = (1,2,3,3,4,5,5,2,1) \text{ or } \mathbf{p}^* = (1,2,3,4,5,1,3,2,4).$$

When $\Omega = 0.5$, the global minimum is:

$$\zeta(\mathbf{p}^*) = 7.853152 \times 10^{-6}, \text{ with } \mathbf{p}^* = (1,2,3,4,2,5,3,2,1).$$

The initial solution is obtained by *k-distance colouring* as explained in Section 5.2, which is $\mathbf{p} = (1,2,3,4,5,1,2,3,4)$. The greedy algorithm generates $N(M-1) = 45$ solutions to obtain the *neighbourhood* of a current solution, and accept the best solution from it. Thus, the greedy algorithm need to wait to obtain all of the 45 *neighbour* solutions, and then accept a new solution. If the current solution is smaller than any of the*neighbour* solutions, the greedy algorithm terminates and output the current solution.

When $\Omega = 0.1$, the initial solution is

$$\zeta(\mathbf{p}) = 0.02491, \text{ with } \mathbf{p} = (1,2,3,4,5,1,2,3,4).$$

Then the greedy algorithm generates the *neighbourhood* of this initial solution, and finds

8 new valid solutions with the following cost values:

$$\zeta(\mathbf{p_1}) = 0.02697783, \text{ with } \mathbf{p_1} = (1,2,3,4,2,1,5,3,2)$$

$$\zeta(\mathbf{p_2}) = 0.02697783, \text{ with } \mathbf{p_2} = (1,2,3,4,5,1,2,5,4)$$

$$\zeta(\mathbf{p_3}) = 0.03853168, \text{ with } \mathbf{p_3} = (1,2,3,4,1,5,2,3,4)$$

$$\zeta(\mathbf{p_4}) = 0.03853168, \text{ with } \mathbf{p_4} = (1,2,3,4,5,1,2,3,5)$$

$$\zeta(\mathbf{p_5}) = 0.04175862, \text{ with } \mathbf{p_5} = (1,2,3,4,3,1,2,5,4)$$

$$\zeta(\mathbf{p_6}) = 0.04175862, \text{ with } \mathbf{p_6} = (1,2,3,4,5,1,5,3,4)$$

$$\zeta(\mathbf{p_7}) = 0.05718572, \text{ with } \mathbf{p_7} = (1,2,3,4,2,1,2,3,4)$$

$$\zeta(\mathbf{p_8}) = 0.05718572, \text{ with } \mathbf{p_8} = (1,2,3,4,3,1,2,3,4)$$

The initial cost value 0.02491 is smaller than any of the 8 neighbours, and thus $\mathbf{p} = (1,2,3,4,5,1,2,3,4)$ is a local minimum $(0.02491 > 0.01830289)$. The greedy algorithm gets stuck at this local minimum, and can not search for other solutions.

When $\Omega = 0.5$, the initial solution is

$$\zeta(\mathbf{p}) = 0.0003457747, \text{ with } \mathbf{p} = (1,2,3,4,5,1,2,3,4).$$

This time the search process of the greedy algorithm is as follows

$$\zeta(\mathbf{p}) = 0.0003457747,\ \text{with } \mathbf{p} = (1,2,3,4,5,1,2,3,4)\ \text{[Initial Solution]}$$

$$\Downarrow \text{Generate the first 45 solutions (8 of which are valid), and accept a new solution}$$

$$\zeta(\mathbf{p}) = 0.0003136025,\ \text{with } \mathbf{p} = (1,2,3,4,2,1,5,3,4)$$

$$\Downarrow \text{Generate the second 45 solutions (8 of which are valid), and accept a new solution}$$

$$\zeta(\mathbf{p}) = 0.0002855033,\ \text{with } \mathbf{p} = (1,2,3,4,2,1,3,5,4)$$

$$\Downarrow \text{Generate the third 45 solutions (8 of which are valid), and accept a new solution}$$

$$\zeta(\mathbf{p}) = 5.70172 \times 10^{-5},\ \text{with } \mathbf{p} = (1,2,3,4,2,5,3,1,4)$$

$$\Downarrow \text{Generate the fourth 45 solutions (7 of which are valid), and output}$$

$$\zeta(\mathbf{p}) = 5.70172 \times 10^{-5},\ \text{with } \mathbf{p} = (1,2,3,4,2,5,3,1,4)$$

The greedy algorithm gets stuck in the local minimum $5.70172 \times 10^{-5} > 7.853152 \times 10^{-6}$ again. In both cases, the greedy algorithm is not able to reach the global minimum.

Next, we apply a relatively slow ASA algorithm with $\alpha_1 = 0.01$, $\beta = 0.25$, and QUENCHing options turned off. Moreover, we adjust the *generating* procedure to change only one random component instead of N at each state, which results in the same *neighbourhood* structure as the greedy algorithm. At each state, the ASA algorithm generates a candidate solution, whose cost value is represented by the solid line in Fig. 5.3(A) or Fig. 5.3(B). The *acceptance* procedure introduced in Appendix D determines whether or not the ASA algorithm takes this candidate solution as the new solution. The dark dash line in Fig. 5.3(A) or Fig. 5.3(B) shows the *acceptance* procedure. We can see that if the generated cost value is smaller than the current one, the ASA algorithm always accepts it as the new one. If the generated cost value is larger than the current one, the ASA algorithm sill allows to accept the worse cost value at certain points. This manner helps the ASA algorithm to jump out of the local minimum where the greedy algorithm is trapped. In the end, when $\Omega = 0.1$ and $\Omega = 0.5$, the ASA algorithm reaches the following global minimum when $N_{\text{gen}} = 296$ and $N_{\text{gen}} = 205$ respectively, which are the same as those obtained from exhaustive search:

$$\text{When } \Omega = 0.1, \quad \zeta(\mathbf{p}^*) = 0.01830289, \text{ with } \mathbf{p}^* = (1,2,3,4,5,1,3,2,4).$$

$$\text{When } \Omega = 0.5, \quad \zeta(\mathbf{p}^*) = 7.853152 \times 10^{-6}, \text{ with } \mathbf{p}^* = (1,2,3,4,2,5,3,2,1).$$

We also apply the normal *generating* procedure illustrated in Section 4.3.2, and the ASA parameters summarized in Table 5.2. The cost values generated and accepted at each state are shown in Fig. 5.4, which is presented in the same form as Fig. 5.3. If $\Omega = 0.1$, it arrives at the global minimum 0.01830289 when $N_{\text{gen}} = 29$, and if $\Omega = 0.5$, it arrives at the global minimum 7.853152×10^{-6} when $N_{\text{gen}} = 79$. It gets the global minimum much faster than the relatively slow ASA introduced in the last paragraph, showing that the ASA algorithm can be very efficient to solve our problem when the parameters are well-tuned.

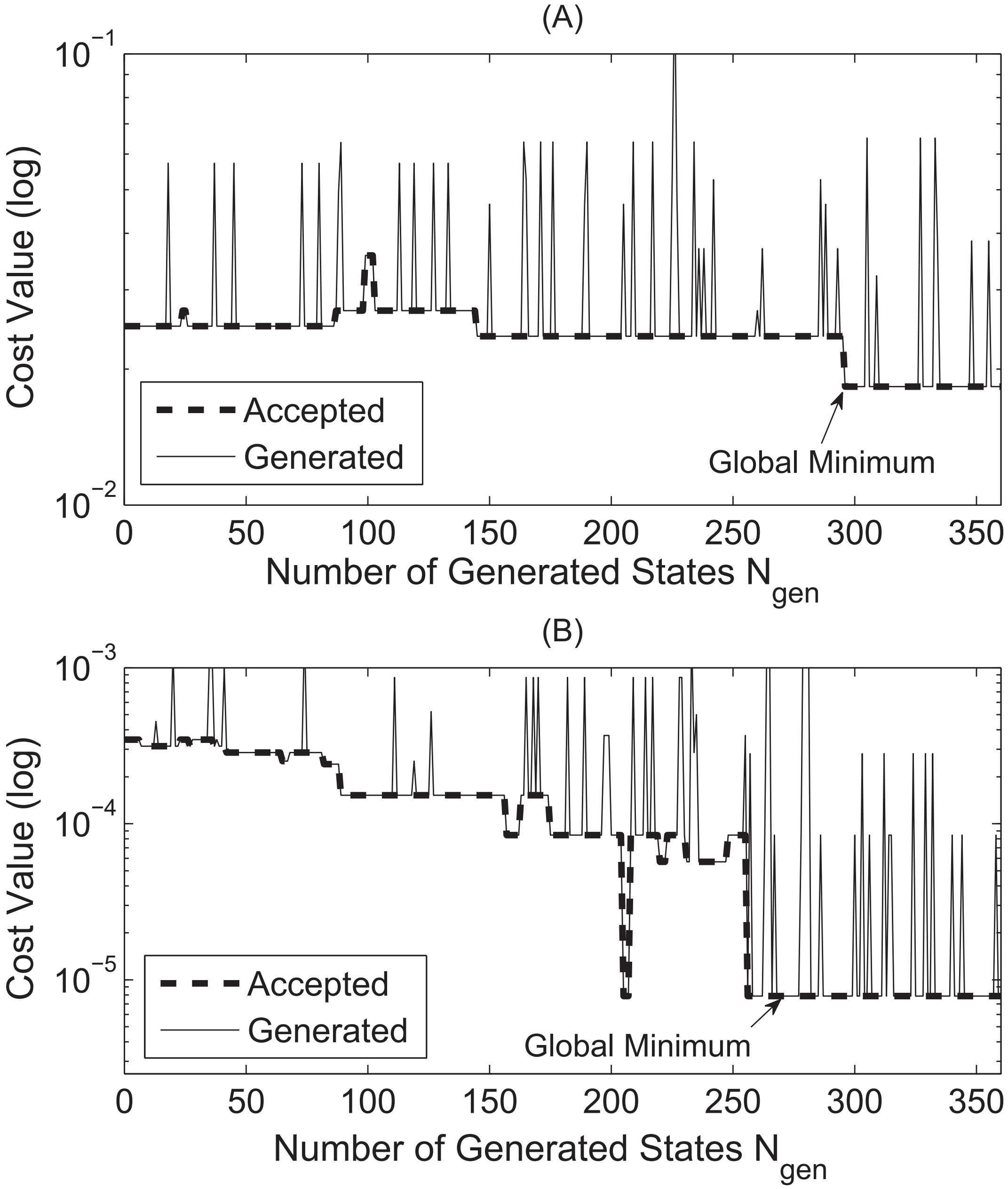

Fig. 5.3 The cost value (log) generated and accepted at each state of the relatively slow ASA algorithms for (A) $\Omega = 0.1$, and (B) $\Omega = 0.5$. The solid line with spikes shows the cost values generated at each state, and the dash line shows the cost values accepted by ASA.

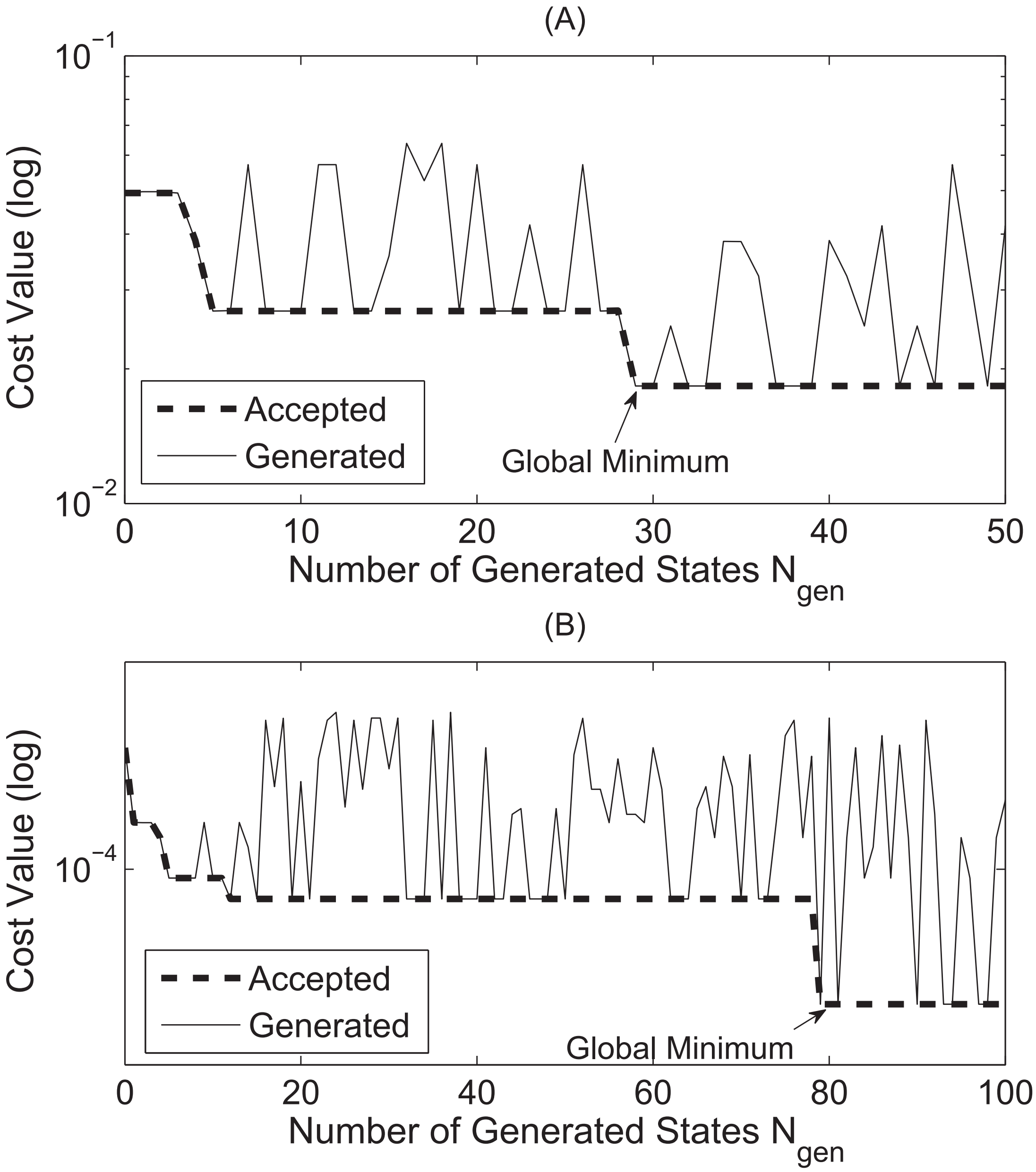

Fig. 5.4 The cost value (log) generated and accepted at each state of the normal ASA algorithms for (A) $\Omega = 0.1$, and (B) $\Omega = 0.5$. The solid line with spikes shows the cost values generated at each state, and the dash line shows the cost values accepted by ASA.

5.4 ROC Curves in Different Configurations

In this section, we analyse the spectrum sensing performance of the grid WSN through the ROC curves, which display Q_d vs Q_f. Q_d and Q_f represent the network probabilities of detection and false alarm averaged over sensor locations, i.e.

$$Q_d = \frac{1}{N}\sum_{i=1}^{N} Q_{d,i}(k_i), \text{ and } Q_f = \frac{1}{N}\sum_{i=1}^{N} Q_{f,i}(k_i) \tag{5.5}$$

where $Q_{d,i}$ and $Q_{f,i}$ are defined in (3.18) and (3.19). We use the *fusion factor* Ω to set the fusion threshold k_i of each sensor, as $k_i = \lceil \Omega n_i \rceil$, where $n_i = |\{s_j | R_{j,i} = 1\}|$ is the number of *partners* of s_i.

In each case, we change one network parameter while fixing others, and use both greedy and ASA algorithms to realize spatial reuse of T-F slots. *k-distance colouring* is applied as the initialization methods for both algorithms. The output of the spatial reuse MAC protocol is represented by an N-dimensional vector $\mathbf{p} = (p_1, p_2, \ldots, p_N)$, whose component p_i is the ID of the T-F slot assigned to s_i. After obtaining this scheduling result $\mathbf{p}$, we calculate the BEP $\varepsilon_{j,i}(\mathbf{p})$ of any communication link $s_j \to s_i$ based on (4.4). Then we can calculate Q_d and Q_f according to (3.7), (3.8), (3.16), (3.17), (3.18), and (3.19).

Moreover, it is worth noting that for the *partner selection* schemes introduced in Section 5.1, a sensor always uses its local sensing result in the decision fusion procedure, i.e. $R_{i,i} = 1$. Since there is no communication link when a sensor takes its own sensing result, $\varepsilon_{i,i} = 0$. Then when using the *OR* fusion rule, according to (3.36), the maximum value of $Q_{d,i}(1)$ or $Q_{f,i}(1)$ is

$$Q_{d,i}^{[\mathrm{U}]}(1) = Q_{f,i}^{[\mathrm{U}]}(1) = 1 - \prod_{s_j \in \{s_j | R_{j,i}=1\}} \varepsilon_{j,i} = 1 - \varepsilon_{i,i} \prod_{s_j \in \{s_j | R_{j,i}=1, j \neq i\}} \varepsilon_{j,i} = 1 - 0 = 1. \tag{5.6}$$

Thus, according to (5.5), the maximum value of Q_d or Q_f when using *OR* fusion rule is

$$Q_d^{[\mathrm{U}]} = Q_f^{[\mathrm{U}]} = \frac{1}{N}\sum_{i=1}^{N} 1 = 1. \tag{5.7}$$

5.4.1 Different Received SNR of the Primary Signal

Fig. 5.5 shows the spectrum sensing performance with different received SNR of the primary signal, $\bar{\gamma}$ = 5dB, 7dB, and 10dB. In Fig. 5.5, we set $N = 64$, $M = 15$, $K = 7$ (Rician fading parameter), and *OR* decision fusion rule ($\Omega = 0.1$) for *CL4*. The scheduling result $\mathbf{p}$ of the two algorithms, with the corresponding $Q_d^{[\mathrm{L}]}$(or $Q_f^{[\mathrm{L}]}$), $Q_d^{[\mathrm{U}]}$ (or $Q_f^{[\mathrm{U}]}$), and the *Achievable Range Loss* $1 - \overline{\Delta}$ are summarized in Table 5.3. Since the average received SNR of the primary signal $\bar{\gamma}$ is not a input of the spatial reuse MAC protocol, the scheduling result $\mathbf{p}$ will not change with $\bar{\gamma}$.

Table 5.3 Scheduling results for the scenario: $N = 64$, $M = 15$, $K = 7$, $\Omega = 0.1$ (*OR* decision fusion rule), and *CL4*.

	ASA										Greedy									
p_1 to p_{10}	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	1	6	7	7	6
p_{11} to p_{20}	7	11	12	13	2	9	12	13	14	1	8	9	10	11	12	13	11	12	13	14
p_{21} to p_{30}	15	10	3	5	8	5	6	9	8	14	2	15	3	4	15	4	5	1	7	8
p_{31} to p_{40}	4	11	15	11	4	3	2	7	1	12	9	10	8	9	10	3	12	13	14	2
p_{41} to p_{50}	2	7	10	12	5	13	6	15	14	13	14	7	2	15	4	5	1	6	5	1
p_{51} to p_{60}	1	15	11	9	10	8	3	9	6	8	6	8	9	10	11	3	3	11	12	13
p_{61} to p_{64}	14	4	3	2							14	7	15	12						
$Q_d^{[\mathrm{L}]}$ (or $Q_f^{[\mathrm{L}]}$)	0.004360498										0.005258959									
$Q_d^{[\mathrm{U}]}$ (or $Q_f^{[\mathrm{U}]}$)	1										1									
$1 - \overline{\Delta}$	0.004360498										0.005258959									

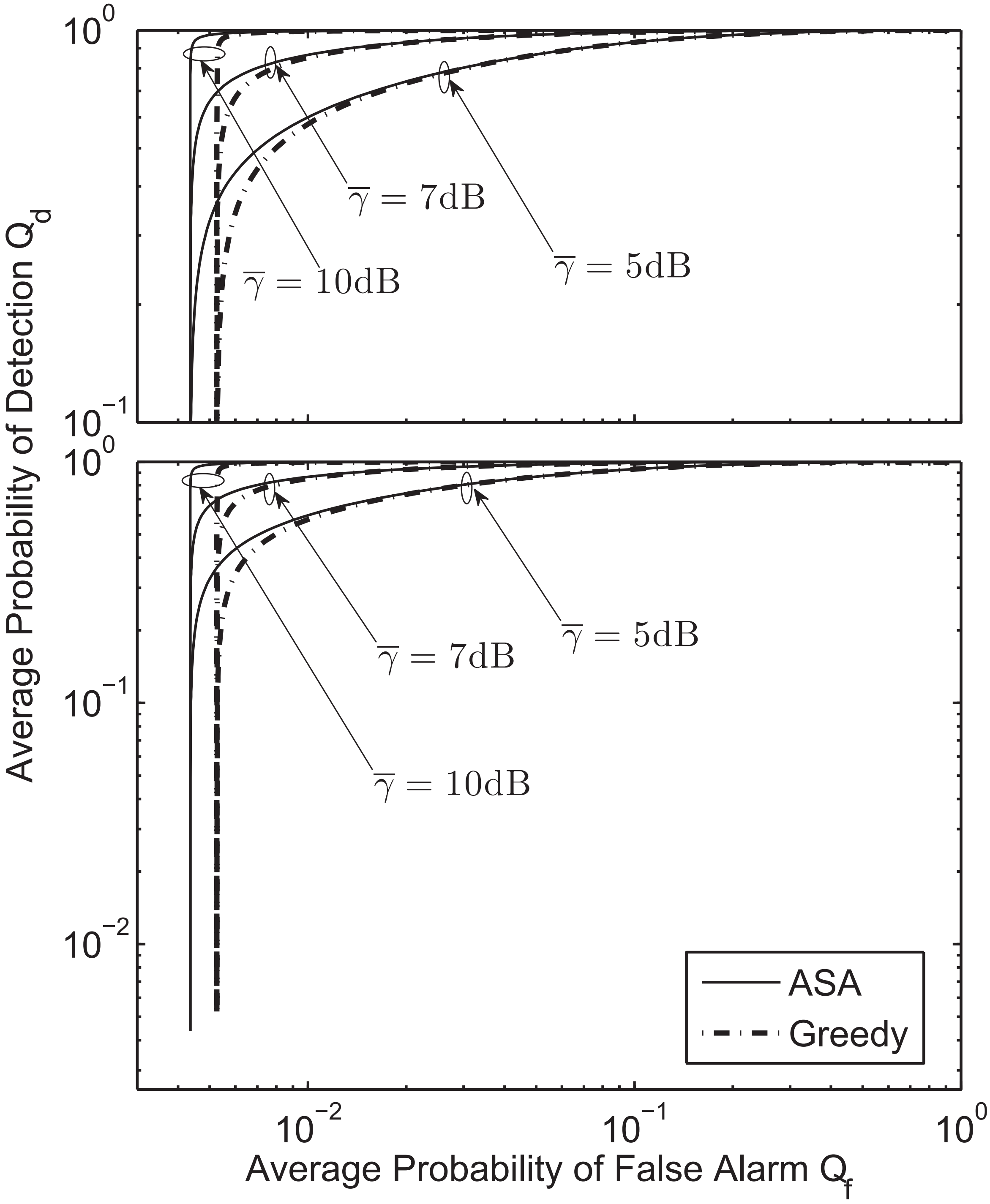

Fig. 5.5 Average Q_d vs Q_f for different different received SNR of the primary signal with $N = 64$, $M = 15$, $K = 7$, *CL4*, and *OR* decision fusion rule.

From Fig. 5.5, we see that for the same $\bar{\gamma}$, the difference between the curves using different algorithms is small for high Q_f range. The ASA algorithm gives better performance than the *Greedy* algorithm for low Q_f range. A Q_f in the range of $0.004360498 \leqslant Q_f < 0.005258959$ is achievable only by the ASA algorithm, and this range does not depend on $\bar{\gamma}$. Table 5.3 shows that ASA algorithm results in a lower *Achievable Range Loss* than the Greedy algorithm. In addition, using a certain scheduling result, $Q_d^{[\mathrm{L}]}$(or $Q_f^{[\mathrm{L}]}$) and $Q_d^{[\mathrm{U}]}$ (or $Q_f^{[\mathrm{U}]}$) don't change with $\bar{\gamma}$, following Property 5. From (3.7) and (3.8), we can see that in the local spectrum sensing performance stage, higher $\bar{\gamma}$ results in higher P_d while keeping P_f unchanged. This translates to better network sensing performance, which can be seen by comparing the ROC curves of the same line style. For the ROC curve with a higher $\bar{\gamma}$, the decrease of Q_d with decreasing Q_f is slower than that of a lower $\bar{\gamma}$. In addition, when Q_f reaches $Q_f^{[\mathrm{L}]}$, Q_d will drastically decreases to $Q_d^{[\mathrm{L}]}$.

5.4.2 Different Numbers of T-F Slots

Fig. 5.6 presents the ROC curves for different values of M with $\bar{\gamma} = 10$dB, $N = 64$, $K = 7$, *CL8*, and $\Omega = 0.1$ (*OR* decision fusion rule). The scheduling results $\mathbf{p}$ of the two algorithms are shown in Table 5.4. As M increases, the sensors sharing one slot can be more spatially separated, resulting in lower BEPs on communication links. In turn, it helps increase Q_d with the same Q_f, and lowers the minimum Q_f, denoted by $Q_f^{[\mathrm{L}]}$. This trend can be seen by comparing the curves of the same line style. The values of $Q_f^{[\mathrm{L}]}$ and $1 - \overline{\Delta}$ are also summarized in Table 5.4, and they decrease with M.

Comparing the curves of different line styles with the same M, we see that the difference between the ASA and greedy algorithms becomes more clear as Q_f decreases from 1 to $Q_f^{[\mathrm{L}]}$ of the greedy algorithm, and the ASA algorithm outperforms the greedy algorithm. As M increases, the difference between the ROC curves of the two algorithms becomes smaller. When $M = N = 64$, no spatial reuse is needed since every sensor can be scheduled into a individual T-F slot, and it provides the upper bound of spectrum sensing performance,

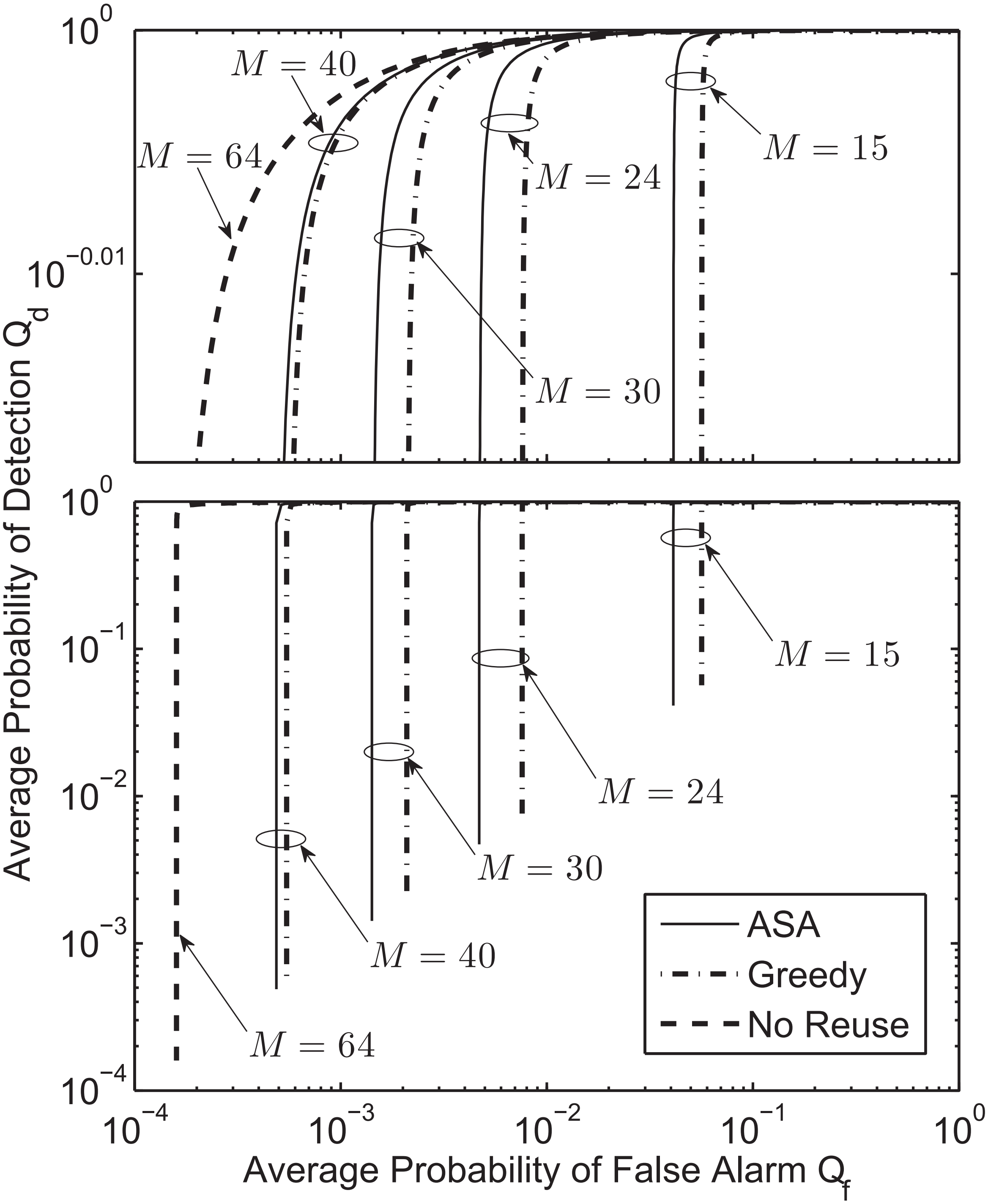

Fig. 5.6 Average Q_d vs Q_f for different numbers of T-F slots M with $\bar{\gamma} =$ 10dB, $N = 64$, $K = 7$, *CL8*, and *OR* decision fusion rule.

Table 5.4 Scheduling results for $M = 15, 24, 30$ and 40, $N = 64$, $K = 7$, $\bar{\gamma} = 10$dB, $\Omega = 0.1$ (*OR* decision fusion rule), and *CL8*.

$[M = 15]$	ASA	Greedy
p_1 to p_{10}	1 2 3 4 5 6 2 7 8 9	1 2 3 4 5 6 7 1 6 7
p_{11} to p_{20}	10 11 8 12 10 13 7 12 14 13	8 9 10 11 12 13 11 12 13 14
p_{21} to p_{30}	15 9 3 11 15 5 6 1 7 4	2 15 4 3 15 4 5 1 3 8
p_{31} to p_{40}	14 1 11 4 3 2 8 5 6 2	9 10 3 9 10 11 12 13 14 1
p_{41} to p_{50}	8 10 9 12 11 10 13 15 1 7	13 14 2 15 4 5 6 7 5 6
p_{51} to p_{60}	13 15 14 3 9 12 14 2 5 6	7 8 9 10 11 12 4 11 12 1
p_{61} to p_{64}	4 1 7 8	3 2 15 8
$Q_d^{[L]}$ (or $Q_f^{[L]}$)	0.04113702	0.05644103
$Q_d^{[U]}$ (or $Q_f^{[U]}$)	1	1
$1 - \overline{\Delta}$	0.04113702	0.05644103
$[M = 24]$	ASA	Greedy
p_1 to p_{10}	1 2 3 4 5 6 7 8 9 10	1 2 3 4 5 6 7 8 9 7
p_{11} to p_{20}	11 12 13 14 15 9 15 7 16 17	10 11 12 13 2 1 14 15 16 17
p_{21} to p_{30}	18 19 20 10 20 6 21 8 22 1	18 19 20 21 19 20 21 22 23 9
p_{31} to p_{40}	23 24 19 23 24 5 2 3 21 11	24 4 24 4 5 8 14 10 11 3
p_{41} to p_{50}	22 4 14 9 12 7 4 16 3 13	6 11 12 13 15 16 17 22 9 7
p_{51} to p_{60}	11 10 15 6 17 13 17 1 16 18	18 19 20 21 1 12 23 1 2 3
p_{61} to p_{64}	20 19 8 14	24 5 6 7
$Q_d^{[L]}$ (or $Q_f^{[L]}$)	0.004694146	0.007587821
$Q_d^{[U]}$ (or $Q_f^{[U]}$)	1	1
$1 - \overline{\Delta}$	0.004694146	0.007587821
$[M = 30]$	ASA	Greedy
p_1 to p_{10}	1 2 3 4 5 6 7 8 7 9	1 2 3 4 5 6 7 2 8 9
p_{11} to p_{20}	10 11 12 13 14 15 16 17 18 19	10 11 12 13 14 15 16 17 18 19
p_{21} to p_{30}	20 21 22 23 15 24 25 26 27 28	20 21 22 23 23 24 25 26 27 28
p_{31} to p_{40}	29 30 29 30 23 8 1 16 24 3	1 29 29 15 6 7 8 9 10 30
p_{41} to p_{50}	28 22 14 6 2 9 17 25 4 21	30 12 13 14 16 17 18 19 1 20
p_{51} to p_{60}	13 5 7 10 18 4 27 20 3 12	21 22 2 24 25 26 11 27 28 3
p_{61} to p_{64}	11 15 19 26	4 5 11 15
$Q_d^{[L]}$ (or $Q_f^{[L]}$)	0.001417139	0.002089166
$Q_d^{[U]}$ (or $Q_f^{[U]}$)	1	1
$1 - \overline{\Delta}$	0.001417139	0.002089166
$[M = 40]$	ASA	Greedy
p_1 to p_{10}	1 2 3 4 5 6 7 8 9 10	1 2 3 4 5 6 7 8 9 10
p_{11} to p_{20}	11 12 13 14 15 16 17 18 19 20	11 12 13 14 15 16 16 17 18 19
p_{21} to p_{30}	21 22 23 24 25 26 27 28 29 30	20 21 22 23 23 24 25 26 27 28
p_{31} to p_{40}	31 25 24 16 32 33 34 35 1 17	29 30 30 7 31 32 33 34 9 35
p_{41} to p_{50}	23 7 36 37 38 39 9 40 40 15	35 15 36 37 38 39 10 40 40 22
p_{51} to p_{60}	6 8 2 3 10 26 31 13 14 5	14 6 3 1 17 24 5 29 13 8
p_{61} to p_{64}	4 11 18 12	2 12 4 11
$Q_d^{[L]}$ (or $Q_f^{[L]}$)	0.000487242	0.0005463582
$Q_d^{[U]}$ (or $Q_f^{[U]}$)	1	1
$1 - \overline{\Delta}$	0.000487242	0.0005463582

serving as a benchmark. At $Q_f = 0.1$, the ASA and the greedy algorithm with $M = 15$ results in the Q_d nearly same as that of the benchmark. At $Q_f = 0.01$, the difference between the Q_d of the ASA and the greedy algorithm with $M = 40$ and the benchmark can also be ignored. Thus, we can save at least 76.6% and 37.5% of T-F slots if the requirements of Q_f are 0.1 and 0.01 respectively, which shows that our spatial reuse MAC can effectively save channel resource for suitable values of Q_f.

5.4.3 Different Cooperation Levels

We evaluate the performance with different *cooperation levels* in Fig. 5.7, where $N = 100$, $M = 15$, $K = 7$, $\bar{\gamma} = 10$dB, and $\Omega = 0.1$ (*OR* rule is applied for decision fusion). The *cooperation levels*: *CL0*, *CL2*, *CL4* and *CL8* are defined in Section 5.1. Since no cooperation is involved in *CL0*, no T-F slot is needed, and it acts as a benchmark for other *cooperation levels*. The scheduling results are presented in Table 5.5. Comparing the curves of different line styles at the same *cooperation level*, we can see that the difference of the ASA and greedy algorithms is mainly manifested in the Q_f range between the lower bounds of Q_f of the two algorithms. For example, using *CL8*, the difference between the curves when $Q_f > 0.08608787$ is nearly negligible, and when $0.07173913 \leqslant Q_f < 0.08608787$, the ASA algorithm outperforms the greedy algorithm. Moreover, when the *cooperation level* increases (i.e. the number of *partners* of each sensor increases), the difference between the values of $Q_f^{[\mathrm{L}]}$ becomes larger, which can be seen from Fig. 5.7 as well as Table 5.5.

Next, we focus on the curves of one algorithm at different *cooperation levels*. When the *cooperation level* increases, the existing *partners* for the lower *cooperation level* remain in sensor s_i's *partner set*, $\{s_j | R_{j,i} = 1\}$, and new *partners* are also added into it. Because the number of slots M is fixed, the interference caused by spatial reuse in the network remains nearly unchanged, and the BEP from an existing *partner* for the lower *cooperation level* varies very little. Then from (3.35), we can get that the lower bound of a sensor's Q_f increases with the *cooperation level*, which is confirmed by the curves in Fig. 5.7, as

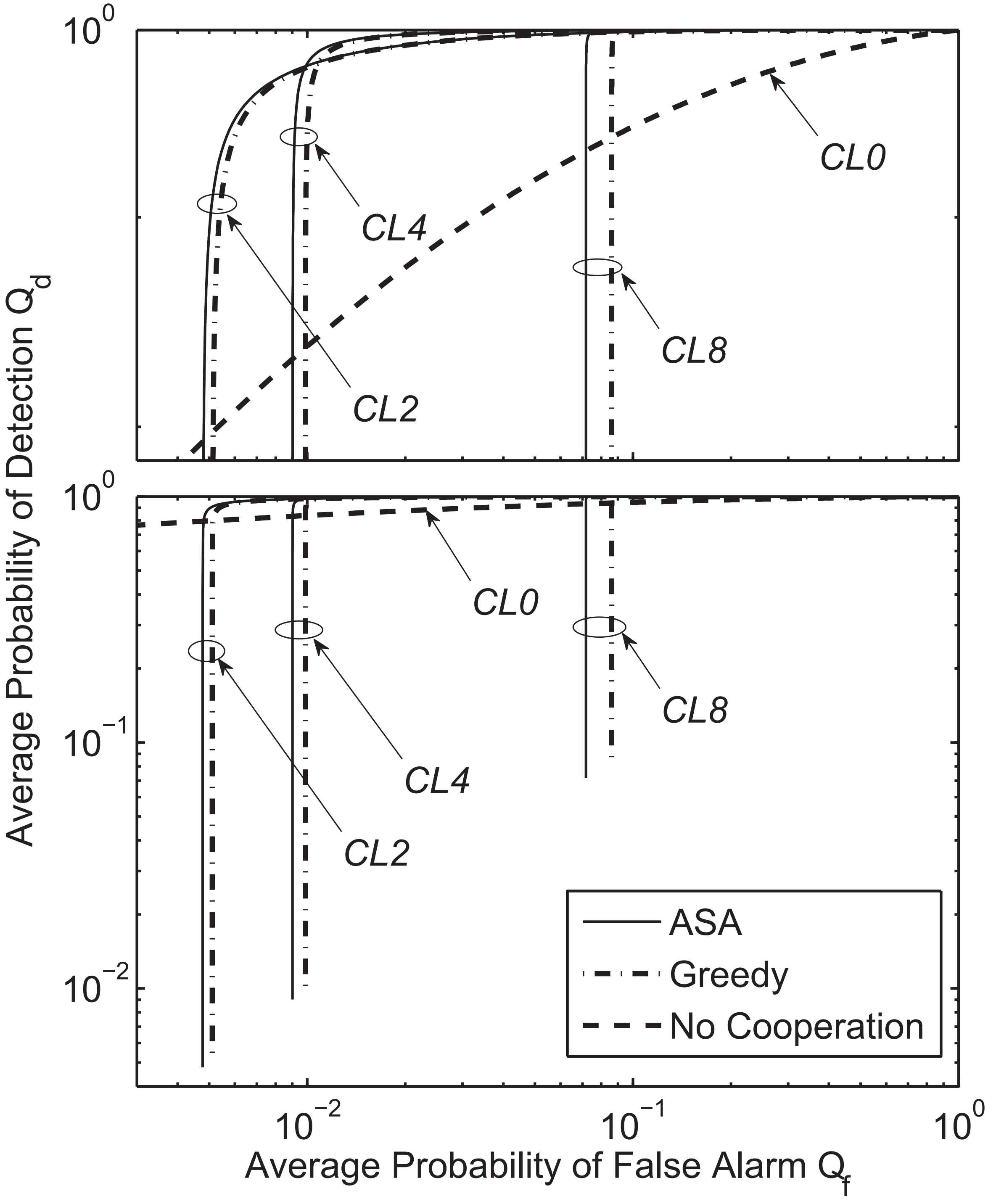

Fig. 5.7 Average Q_d vs Q_f for different cooperation levels with $N = 100$, $M = 15$, $K = 7$, $\bar{\gamma} = 10$dB and $\Omega = 0.1$ (*OR* decision fusion rule).

Table 5.5 Scheduling results for different cooperation levels (*CL2*, *CL4* and *CL8*), with $N = 100$, $M = 15$, $K = 7$, $\bar{\gamma} = 10$dB, and $\Omega = 0.1$ (*OR* decision fusion rule).

CL2	ASA										Greedy									
p_1 to p_{10}	1	2	3	4	5	6	1	2	7	8	1	2	3	4	5	6	7	8	9	10
p_{11} to p_{20}	5	9	7	8	10	11	3	12	5	13	6	7	8	11	10	12	13	14	15	11
p_{21} to p_{30}	10	11	14	12	13	15	9	4	14	11	12	13	14	15	2	3	4	5	6	7
p_{31} to p_{40}	12	6	15	1	2	7	8	6	10	15	9	4	5	1	9	8	11	1	12	13
p_{41} to p_{50}	4	13	8	9	4	14	5	13	1	2	8	11	10	12	13	14	15	2	3	10
p_{51} to p_{60}	11	14	12	7	15	6	2	4	8	14	1	6	7	9	11	10	1	13	14	15
p_{61} to p_{70}	1	2	5	3	10	11	12	3	9	7	14	15	2	3	4	5	6	7	8	9
p_{71} to p_{80}	3	6	4	13	1	9	7	1	10	11	10	12	13	14	15	2	3	4	5	6
p_{81} to p_{90}	10	9	8	2	11	14	5	13	6	12	9	3	4	5	6	7	8	11	10	12
p_{91} to p_{100}	7	15	5	3	10	12	8	15	3	4	2	8	11	1	12	13	14	9	2	1
$Q_d^{[\mathrm{L}]}$ (or $Q_f^{[\mathrm{L}]}$)	0.004778993										0.005109983									
$Q_d^{[\mathrm{U}]}$ (or $Q_f^{[\mathrm{U}]}$)	1										1									
$1 - \overline{\Delta}$	0.004778993										0.005109983									
CL4	ASA										Greedy									
p_1 to p_{10}	1	2	3	4	5	2	6	4	7	1	1	2	3	4	5	6	1	7	8	4
p_{11} to p_{20}	7	8	9	10	7	1	11	10	12	9	6	9	7	8	10	11	12	13	14	2
p_{21} to p_{30}	12	13	14	11	12	3	15	13	5	14	11	12	13	14	2	3	15	5	6	9
p_{31} to p_{40}	10	5	15	6	8	14	9	2	8	3	4	15	5	6	9	4	8	10	11	1
p_{41} to p_{50}	11	4	3	1	13	4	7	6	11	4	7	8	10	1	12	13	14	2	3	15
p_{51} to p_{60}	6	9	7	12	2	5	10	1	12	15	13	14	2	3	15	5	1	9	7	8
p_{61} to p_{70}	14	2	10	8	11	15	14	13	9	5	1	6	9	7	4	10	11	12	13	4
p_{71} to p_{80}	13	1	5	4	9	3	6	8	7	2	10	11	12	13	14	2	3	15	5	6
p_{81} to p_{90}	12	15	6	14	7	1	12	4	11	3	2	3	15	5	6	9	7	8	10	11
p_{91} to p_{100}	8	3	11	13	10	2	5	15	10	14	4	7	8	1	11	12	4	14	2	1
$Q_d^{[\mathrm{L}]}$ (or $Q_f^{[\mathrm{L}]}$)	0.009014317										0.0098602									
$Q_d^{[\mathrm{U}]}$ (or $Q_f^{[\mathrm{U}]}$)	1										1									
$1 - \overline{\Delta}$	0.009014317										0.0098602									
CL8	ASA										Greedy									
p_1 to p_{10}	1	2	3	4	5	6	3	2	1	4	1	2	3	4	5	6	1	7	8	4
p_{11} to p_{20}	7	8	9	10	11	7	12	13	5	8	6	9	7	8	10	11	12	13	14	2
p_{21} to p_{30}	11	14	12	13	1	8	9	10	14	7	11	12	13	14	2	3	15	5	6	9
p_{31} to p_{40}	15	6	5	2	15	14	4	11	15	6	4	15	5	1	9	4	8	10	11	1
p_{41} to p_{50}	13	1	4	3	7	6	5	2	3	1	7	8	10	11	12	13	14	2	3	15
p_{51} to p_{60}	9	10	8	11	9	10	12	8	13	9	13	14	2	3	15	5	6	9	7	8
p_{61} to p_{70}	2	7	12	14	13	1	15	7	14	4	1	6	9	7	4	10	1	12	13	4
p_{71} to p_{80}	6	3	15	5	2	4	3	11	6	10	10	11	12	13	14	2	3	15	5	6
p_{81} to p_{90}	13	4	1	10	6	8	9	5	2	12	2	3	15	5	6	9	7	8	10	11
p_{91} to p_{100}	14	8	9	11	7	12	14	13	15	1	4	7	8	1	11	12	4	14	2	1
$Q_d^{[\mathrm{L}]}$ (or $Q_f^{[\mathrm{L}]}$)	0.07173913										0.08608787									
$Q_d^{[\mathrm{U}]}$ (or $Q_f^{[\mathrm{U}]}$)	1										1									
$1 - \overline{\Delta}$	0.07173913										0.08608787									

well as the values of $Q_f^{[\mathrm{L}]}$ listed in Table 5.5. When Q_f is higher than the lower bound $Q_f^{[\mathrm{L}]}$, cooperation between sensors helps increase Q_d compared with *CL0*. However, the difference between *CL2*, *CL4* and *CL8* in the high Q_f range ($Q_f \geqslant 0.1$) is very small. For the case shown in Fig. 5.7, we can divide the Q_f axis into 4 intervals with boundaries close to $Q_f^{[\mathrm{L}]}$ at each *cooperation level*, and find the best choice of *cooperation levels* in each interval. Taking the ASA curves as an example, when $0.0916 < Q_f < 1$, *CL2*, *CL4* and *CL8* performs nearly the same, *CL4* is the best for $0.00993 < Q_f < 0.0916$, *CL2* is the best for $0.00478 < Q_f < 0.00993$, and if the system required $Q_f < 0.00478$, *CL0* is the only choice.

5.4.4 Different Network Sizes

Three networks sizes, $N = 36,\ 64,\ 100$ are analysed in Fig. 5.8 and Fig. 5.9, with $K = 7$, *CL4*, $\bar{\gamma} = 10$dB, and $\Omega = 0.1$ (*OR* decision fusion rule). In Fig. 5.8, we fix the value of M to 15, and in Fig. 5.9 we fix the ratio M/N to approximately 0.4. The scheduling results for these scenarios are presented in Table 5.6 and Table 5.7, in which the value of the lower bound $Q_f^{[\mathrm{L}]}$ (or $Q_d^{[\mathrm{L}]}$) as well as the *Achievable Range Loss*, $1-\overline{\Delta}$ are also listed. In any case with the same parameters, ASA algorithm gives a lower $1-\overline{\Delta}$ than the Greedy algorithm as shown in the tables. In addition, if we compare the complimentary ROC curves, we see that when $Q_f > 0.02$, the difference between two algorithms are very small. In relatively low Q_f range, for the same value of Q_f, the ASA algorithm results in the higher Q_d compared with the greedy algorithm, which means a better spectrum sensing performance. Specifically, for $N = 36$ or $N = 64$, the ASA algorithm outperforms the greedy algorithm when $Q_f < 0.01$, and for $N = 100$, the difference is identifiable when $Q_f < 0.02$.

For *CL4*, a sensor located at the corner has 3 *partners*, the one on the edge has 4 *partners*, while any other sensor has 5 *partners*. When N is increased, the proportion of sensors on the edge, $4(\sqrt{N} - 1)/N$, as well as that at the corner, $4/N$, decreases, which results in the increasing average number of *partners* of each sensor. When M is fixed, larger N means an increasing average numbers of sensors sharing the same T-F slot. Thus, the network interference level increases, resulting in higher average BEP of communication links. As shown in Fig. 5.8, for relatively high Q_f, the gain brought by more *partners* is counteracted by the impact of worse reporting channels. In addition, according to (3.35), more *partners* and higher BEPs of reporting channels both lead to a higher $Q_f^{[\mathrm{L}]}$, which is shown in both Fig. 5.8, and Table 5.6. When the ratio M/N is kept the same, Fig. 5.9 shows that the performance gets better when increasing N. This is because in a larger network, although the average number of sensors sharing one T-F slot remains almost unchanged, the average distance between them increases, which leads to a lower network interference level and lower average reporting BEP.

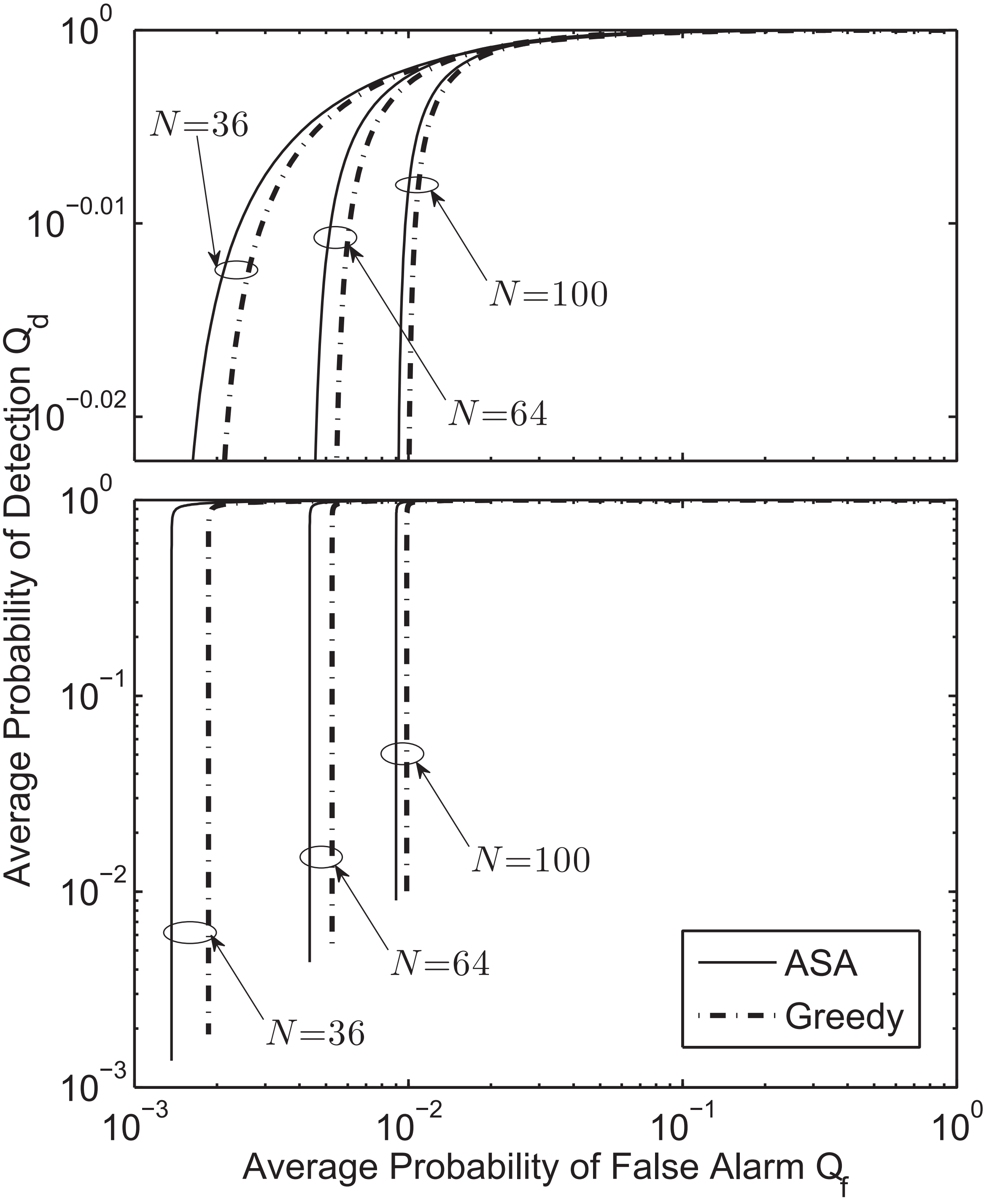

Fig. 5.8 Average Q_d vs Q_f for different numbers of sensors N with $M = 15$, $K = 7$, $\bar{\gamma} = 10$dB, *CL4*, and $\Omega = 0.1$ (*OR* decision fusion rule).

Table 5.6 Scheduling results for different numbers of sensors $N = 36, 64$ and 100, with $M = 15$, $K = 7$, $\bar{\gamma} = 10$dB, and $\Omega = 0.1$ (*OR* decision fusion rule).

$[N = 36]$	ASA										Greedy									
p_1 to p_{10}	1	2	3	4	5	6	6	7	8	9	1	2	3	4	5	6	7	8	9	10
p_{11} to p_{20}	10	11	12	13	14	15	1	7	5	10	1	11	11	12	13	14	2	15	15	6
p_{21} to p_{30}	11	2	12	3	9	3	4	6	13	14	5	7	8	9	9	10	1	11	12	13
p_{31} to p_{36}	15	1	7	8	5	9					4	14	2	3	4	6				
$Q_d^{[\mathrm{L}]}$ (or $Q_f^{[\mathrm{L}]}$)	0.001366718										0.001863652									
$Q_d^{[\mathrm{U}]}$ (or $Q_f^{[\mathrm{U}]}$)	1										1									
$1 - \overline{\Delta}$	0.001366718										0.001863652									

$[N = 64]$	ASA										Greedy									
p_1 to p_{10}	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	1	6	7	7	6
p_{11} to p_{20}	7	11	12	13	2	9	12	13	14	1	8	9	10	11	12	13	11	12	13	14
p_{21} to p_{30}	15	10	3	5	8	5	6	9	8	14	2	15	3	4	15	4	5	1	7	8
p_{31} to p_{40}	4	11	15	11	4	3	2	7	1	12	9	10	8	9	10	3	12	13	14	2
p_{41} to p_{50}	2	7	10	12	5	13	6	15	14	13	14	7	2	15	4	5	1	6	5	1
p_{51} to p_{60}	1	15	11	9	10	8	3	9	6	8	6	8	9	10	11	3	3	11	12	13
p_{61} to p_{64}	14	4	3	2							14	7	15	12						
$Q_d^{[\mathrm{L}]}$ (or $Q_f^{[\mathrm{L}]}$)	0.004360498										0.005258959									
$Q_d^{[\mathrm{U}]}$ (or $Q_f^{[\mathrm{U}]}$)	1										1									
$1 - \overline{\Delta}$	0.004360498										0.005258959									

$[N = 100]$	ASA										Greedy									
p_1 to p_{10}	1	2	3	4	5	2	6	4	7	1	1	2	3	4	5	6	1	7	8	4
p_{11} to p_{20}	7	8	9	10	7	1	11	10	12	9	6	9	7	8	10	11	12	13	14	2
p_{21} to p_{30}	12	13	14	11	12	3	15	13	5	14	11	12	13	14	2	3	15	5	6	9
p_{31} to p_{40}	10	5	15	6	8	14	9	2	8	3	4	15	5	6	9	4	8	10	11	1
p_{41} to p_{50}	11	4	3	1	13	4	7	6	11	4	7	8	10	1	12	13	14	2	3	15
p_{51} to p_{60}	6	9	7	12	2	5	10	1	12	15	13	14	2	3	15	5	1	9	7	8
p_{61} to p_{70}	14	2	10	8	11	15	14	13	9	5	1	6	9	7	4	10	11	12	13	4
p_{71} to p_{80}	13	1	5	4	9	3	6	8	7	2	10	11	12	13	14	2	3	15	5	6
p_{81} to p_{90}	12	15	6	14	7	1	12	4	11	3	2	3	15	5	6	9	7	8	10	11
p_{91} to p_{100}	8	3	11	13	10	2	5	15	10	14	4	7	8	1	11	12	4	14	2	1
$Q_d^{[\mathrm{L}]}$ (or $Q_f^{[\mathrm{L}]}$)	0.009014317										0.0098602									
$Q_d^{[\mathrm{U}]}$ (or $Q_f^{[\mathrm{U}]}$)	1										1									
$1 - \overline{\Delta}$	0.009014317										0.0098602									

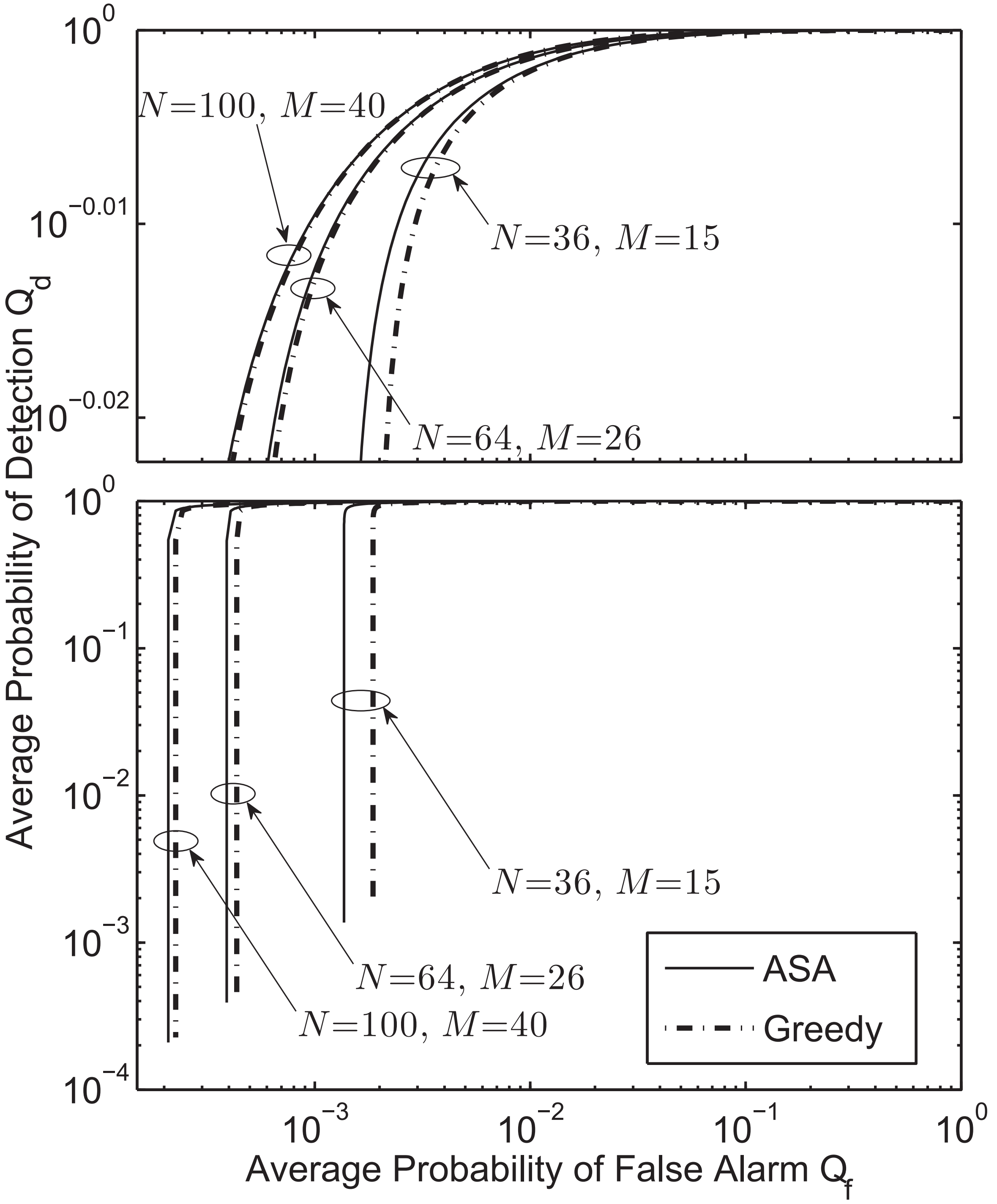

Fig. 5.9 Average Q_d vs Q_f for different numbers of sensors N with $M/N \approx 0.4$, $K = 7$, $\bar{\gamma} = 10$dB, *CL4*, and $\Omega = 0.1$ (*OR* decision fusion rule).

Table 5.7 Scheduling results for different numbers of sensors N with $M/N \approx 0.4$, $K = 7$, $\bar{\gamma} = 10\text{dB}$, *CL4*, and $\Omega = 0.1$ (*OR* decision fusion rule).

$[N = 36, M = 15]$	ASA										Greedy									
p_1 to p_{10}	1	2	3	4	5	6	6	7	8	9	1	2	3	4	5	6	7	8	9	10
p_{11} to p_{20}	10	11	12	13	14	15	1	7	5	10	1	11	11	12	13	14	2	15	15	6
p_{21} to p_{30}	11	2	12	3	9	3	4	6	13	14	5	7	8	9	9	10	1	11	12	13
p_{31} to p_{36}	15	1	7	8	5	9					4	14	2	3	4	6				
$Q_d^{[L]}$ (or $Q_f^{[L]}$)	0.001366718										0.001863652									
$Q_d^{[U]}$ (or $Q_f^{[U]}$)	1										1									
$1 - \overline{\Delta}$	0.001366718										0.001863652									
$[N = 64, M = 26]$	ASA										Greedy									
p_1 to p_{10}	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
p_{11} to p_{20}	11	12	13	14	9	1	15	16	8	17	11	12	13	14	15	16	16	17	18	19
p_{21} to p_{30}	18	19	15	20	21	20	22	23	24	21	20	21	1	22	22	8	23	24	25	2
p_{31} to p_{40}	2	25	5	6	25	7	26	16	10	3	3	26	26	5	6	7	9	10	11	12
p_{41} to p_{50}	13	19	14	1	4	11	5	22	24	18	12	13	14	15	16	17	18	19	19	20
p_{51} to p_{60}	3	9	12	17	13	23	26	10	2	15	21	1	22	8	23	24	11	25	2	3
p_{61} to p_{64}	8	20	6	18							4	5	6	13						
$Q_d^{[L]}$ (or $Q_f^{[L]}$)	0.0003903481										0.0004340288									
$Q_d^{[U]}$ (or $Q_f^{[U]}$)	1										1									
$1 - \overline{\Delta}$	0.0003903481										0.0004340288									
$[N = 100, M = 40]$	ASA										Greedy									
p_1 to p_{10}	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
p_{11} to p_{20}	11	12	10	13	14	15	16	17	2	18	11	9	12	13	14	15	16	17	18	19
p_{21} to p_{30}	17	19	20	21	22	23	24	25	26	27	20	18	21	22	23	24	25	26	2	1
p_{31} to p_{40}	28	18	26	29	30	31	1	32	12	3	27	28	29	30	31	32	33	11	34	28
p_{41} to p_{50}	7	33	34	9	35	36	37	33	20	38	35	34	36	37	38	39	40	3	35	4
p_{51} to p_{60}	6	39	38	8	27	11	40	39	21	28	6	15	19	7	10	27	12	13	14	5
p_{61} to p_{70}	40	15	25	32	16	19	4	6	29	14	13	14	5	16	17	20	21	22	23	38
p_{71} to p_{80}	14	37	24	5	3	10	18	13	22	15	8	23	24	25	26	28	29	30	31	15
p_{81} to p_{90}	22	31	12	23	28	17	2	34	24	31	3	9	32	33	11	18	36	37	1	7
p_{91} to p_{100}	21	36	20	1	26	7	30	9	8	35	39	1	2	40	4	34	6	19	8	9
$Q_d^{[L]}$ (or $Q_f^{[L]}$)	0.0002090692										0.0002263128									
$Q_d^{[U]}$ (or $Q_f^{[U]}$)	1										1									
$1 - \overline{\Delta}$	0.0002090692										0.0002263128									

5.4.5 Different Decision Fusion Rules

The ROC curves for different values of fusion factor $\Omega = 0.1, 0.3, 0.5, 0.7, 1$ are presented in Fig. 5.10, where $N = 64$, $M = 15$, $K = 7$, $\bar{\gamma} = 10$dB and *CL4* is applied. Since the cost function (4.13) for the ASA and Greedy algorithm varies with Ω, we have different scheduling results $\mathbf{p}$ for different decision fusion rules used in the system. For *CL4*, sensors at different locations in the network may have different numbers of *partners* n_i. For the 4 sensors in the corner, $n_i = 3$, for the $4(\sqrt{N} - 1) = 20$ sensors on the edge (but not in the corner), n_i=4, and the other $N - 4\sqrt{N} = 12$ sensors have $n_i = 5$. The decision fusion threshold k_i at each sensor is $k_i = \lceil \Omega n_i \rceil$, and we can have

- For $n_i = 3$, $k_i|_{\Omega=0.3} = \lceil 0.3 \times 3 \rceil = 1$, and $k_i|_{\Omega=0.7} = \lceil 0.7 \times 3 \rceil = 3$. Thus, $n_i + 1 - k_i|_{\Omega=0.7} = 3 + 1 - 3 = 1 = k_i|_{\Omega=0.3}$. $k_i|_{\Omega=0.1} = \lceil 0.1 \times 3 \rceil = 1$, and $k_i|_{\Omega=1} = \lceil 1 \times 3 \rceil = 3$. Thus, $n_i + 1 - k_i|_{\Omega=1} = 3 + 1 - 3 = 1 = k_i|_{\Omega=0.1}$.

- For $n_i = 4$, $k_i|_{\Omega=0.3} = \lceil 0.3 \times 4 \rceil = 2$, and $k_i|_{\Omega=0.7} = \lceil 0.7 \times 4 \rceil = 3$. Thus, $n_i + 1 - k_i|_{\Omega=0.7} = 4 + 1 - 3 = 2 = k_i|_{\Omega=0.3}$. $k_i|_{\Omega=0.1} = \lceil 0.1 \times 4 \rceil = 1$, and $k_i|_{\Omega=1} = \lceil 1 \times 4 \rceil = 4$. Thus, $n_i + 1 - k_i|_{\Omega=1} = 4 + 1 - 1 = 4 = k_i|_{\Omega=0.1}$.

- For $n_i = 5$, $k_i|_{\Omega=0.3} = \lceil 0.3 \times 5 \rceil = 2$, and $k_i|_{\Omega=0.7} = \lceil 0.7 \times 5 \rceil = 4$. Thus, $n_i + 1 - k_i|_{\Omega=0.7} = 5 + 1 - 2 = 4 = k_i|_{\Omega=0.3}$. $k_i|_{\Omega=0.1} = \lceil 0.1 \times 5 \rceil = 1$, and $k_i|_{\Omega=1} = \lceil 1 \times 5 \rceil = 5$. Thus, $n_i + 1 - k_i|_{\Omega=1} = 5 + 1 - 1 = 5 = k_i|_{\Omega=0.1}$.

From Property 7, the *Achievable Range* $\Delta_{d,i}(k_i)$ for each sensor satisfies $\Delta_i(k_i) = \Delta_i(n_i + 1 - k_i)$. Therefore, in this scenario, $\Omega = 0.1$ and $\Omega = 1$ result in the same *Achievable Range* Δ_i of each sensor, and $\Omega = 0.3$ and $\Omega = 0.7$ result in the same *Achievable Range* Δ_i of each sensor. Then according to (4.6) and (4.13), $\Omega = 0.1$ and $\Omega = 1$ have the same cost function for the optimization algorithm, and so do $\Omega = 0.3$ and $\Omega = 0.7$. Therefore, $\Omega = 0.1$ and $\Omega = 1$ have the same scheduling result, and $\Omega = 0.3$ and $\Omega = 0.7$ have the same scheduling result. Scheduling results for different decision fusion factors $\Omega = 0.1, 0.3, 0.5, 0.7$ and 1 in this scenario are summarised in Table 5.8. From the table, we can see that for each Ω, the scheduling result obtained through ASA algorithm gives a lower $Q_d^{[\mathrm{L}]}$ (or $Q_f^{[\mathrm{L}]}$), higher $Q_d^{[\mathrm{U}]}$ (or $Q_f^{[\mathrm{U}]}$), as well as lower *Achievable Range Loss* $1 - \overline{\Delta}$.

In Fig. 5.10, we can see that the separation between the ROC curves of ASA and greedy algorithms becomes smaller as Ω increases. When $\Omega = 1$, we can hardly differentiate the ROC curves for two algorithms in Fig. 5.10. Comparing the curves for the same algorithm in Fig. 5.10, or the values listed in Table 5.8, we can also see that the lower bound of Q_f and the upper bound of Q_d decrease with increasing Ω. This result is in accord with Property 6. Furthermore, in this case, for the same algorithm, we cannot conclude which Ω is better than others for any Q_f. We have to choose Ω according to the specific Q_f or Q_d requirements and the ROC curves shown in Fig. 5.10. Take the ROC curves for the ASA algorithm for example, when the required Q_f is in the range $Q_f > 0.01$, the ROC curves of $\Omega = 0.1$, $\Omega = 0.3$ and $\Omega = 0.5$ have very small difference, which all result in higher Q_d than $\Omega = 0.7$ or $\Omega = 1$. If the system requirement is $Q_f < 0.001$, $\Omega = 0.1$ cannot achieve this Q_f and $\Omega = 0.5$ is better than $\Omega = 0.3$. In this case, for $10^{-6} < Q_f < 1$ we can always choose $\Omega = 0.5$, and $Q_f = 1$ is the worst choice.

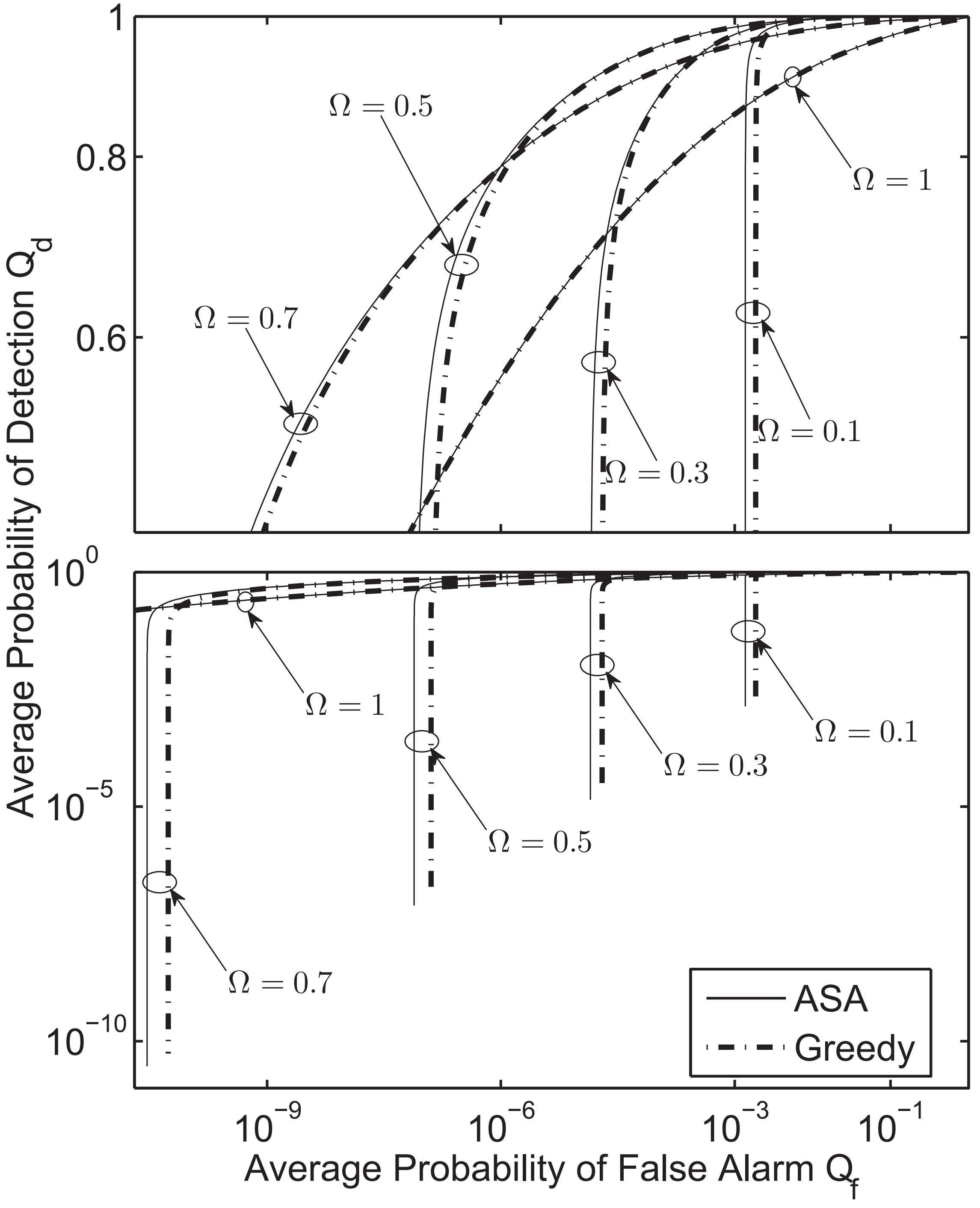

Fig. 5.10 Average Q_d vs Q_f for different decision fusion factors $\Omega = 0.1, 0.3, 0.5, 0.7$ and 1 with $N = 64$, $M = 15$, $K = 7$, $\bar{\gamma} = 10$dB, and *CL4*.

Table 5.8 Scheduling results for different decision fusion factors $\Omega = 0.1, 0.3, 0.5, 0.7$ and 1 with $N = 64$, $M = 15$, $K = 7$, $\bar{\gamma} = 10$dB, and *CL4*.

[$\Omega = 0.1$ or 1]	ASA	Greedy
p_1 to p_{10}	1 2 3 4 5 6 6 7 8 9	1 2 3 4 5 6 7 8 9 10
p_{11} to p_{20}	10 11 12 13 14 15 1 7 5 10	1 11 11 12 13 14 2 15 15 6
p_{21} to p_{30}	11 2 12 3 9 3 4 6 13 14	5 7 8 9 9 10 1 11 12 13
p_{31} to p_{36}	15 1 7 8 5 9	4 14 2 3 4 6
$Q_d^{[\mathrm{L}]}$ (or $Q_f^{[\mathrm{L}]}$)	0.001366718 for $\Omega = 0.1$	0.001863652 for $\Omega = 0.1$
	0 for $\Omega = 1$	0 for $\Omega = 1$
$Q_d^{[\mathrm{U}]}$ (or $Q_f^{[\mathrm{U}]}$)	1 for $\Omega = 0.1$	1 for $\Omega = 0.1$
	$1 - 0.001366718$ for $\Omega = 1$	$1 - 0.001863652$ for $\Omega = 1$
$1 - \overline{\Delta}$	0.001366718	0.001863652
[$\Omega = 0.3$ or 0.7]	ASA	Greedy
p_1 to p_{10}	1 2 3 4 5 1 6 7 8 9	1 2 3 4 5 6 7 8 9 10
p_{11} to p_{20}	10 11 12 13 14 15 12 13 4 10	1 11 12 6 13 14 2 3 3 15
p_{21} to p_{30}	1 3 7 14 11 15 9 8 4 6	4 12 8 13 9 10 1 11 6 7
p_{31} to p_{36}	7 5 12 13 2 10	8 5 14 3 15 10
$Q_d^{[\mathrm{L}]}$ (or $Q_f^{[\mathrm{L}]}$)	1.407227×10^{-5} for $\Omega = 0.3$	1.984026×10^{-5} for $\Omega = 0.3$
	2.893060×10^{-11} for $\Omega = 0.7$	5.418733×10^{-11} for $\Omega = 0.7$
$Q_d^{[\mathrm{U}]}$ (or $Q_f^{[\mathrm{U}]}$)	$1 - 2.893075 \times 10^{-11}$ for $\Omega = 0.3$	$1 - 2.893097 \times 10^{-11}$ for $\Omega = 0.3$
	$1 - 1.407227 \times 10^{-5}$ for $\Omega = 0.7$	$1 - 1.984026 \times 10^{-5}$ for $\Omega = 0.7$
$1 - \overline{\Delta}$	1.407230×10^{-5}	1.984032×10^{-5}
[$\Omega = 0.5$]	ASA	Greedy
p_1 to p_{10}	1 2 3 4 5 6 7 8 9 10	1 2 3 4 5 6 7 8 9 10
p_{11} to p_{20}	11 7 6 12 13 14 15 1 15 11	11 12 12 6 13 14 1 15 15 4
p_{21} to p_{30}	5 8 12 2 4 3 10 9 4 3	5 7 8 2 2 10 11 12 6 13
p_{31} to p_{36}	2 1 14 7 6 13	13 14 1 9 3 4
$Q_d^{[\mathrm{L}]}$ (or $Q_f^{[\mathrm{L}]}$)	7.720442×10^{-8}	1.271326×10^{-7}
$Q_d^{[\mathrm{U}]}$ (or $Q_f^{[\mathrm{U}]}$)	$1 - 5.127297 \times 10^{-9}$	$1 - 3.374461 \times 10^{-9}$
$1 - \overline{\Delta}$	8.233172×10^{-8}	1.305070×10^{-7}

Chapter 6

Conclusions

In this thesis, we analysed the cooperative spectrum sensing performance using *k-out-of-n* decision fusion rule with imperfect reporting channels, considering small-scale fading, path loss, and additive interference. We derived the expressions and properties for the upper and lower bounds of probabilities of detection and false alarm. Then, we introduced a DetF distributed WSN for cooperative spectrum sensing, and proposed a spatial reuse MAC protocol based on TDMA/OFDMA. Two design approaches for the MAC protocol were considered: greedy and ASA algorithms. Finally, for a grid WSN, we discussed how to construct the adjacency matrix charactering the cooperating relations between sensors. We also explain how to determine the minimum number of T-F slots, and get the initial *valid* solution via a k-distance colouring method. Numerical results are presented to analyse the spectrum sensing performance of the DetF WSN with spatial reuse MAC protocol in such a grid network form. It is worth noting the proposed spatial resue MAC protocol is not limited to the grid network presented in this work, and can be employed in any other network structures with a certain cooperating partner selection scheme.

We started from a single link $s_j \rightarrow s_i$ to analyse the impact of the reporting channel error in Section 3.3.1. As the BEP of the reporting channel ε increases, the equivalent probability of detection at the receiving sensor $P_{d,\varepsilon}$ decreases for the same equivalent probability of false alarm $P_{f,\varepsilon}$, which is explained in Property 1. The impact of the reporting channel error is also illustrated by Fig. 3.1, where the equivalent ROC curve shifts down, and the achievable range of it shrinks as ε increases. Next, we analyse *k-out-of-n* decision fusion multiple reporting channels with non-identical BEP. We obtained the expressions for the minimum and maximum value of probabilities of detection and false alarm, which do not depend on the local spectrum sensing results and are only determined by the BEP of each reporting channel errors.

The spatial reuse MAC protocol is designed to control the exchange of sensing results within the WSN in order to exclude primary conflict and save bandwidth resource. The basic concept is to divide N sensors in the WSN into M separate sets, and assign one T-F slot to the sensors in the same set. We choose the *Network Achievable Range Loss* of the probability of detection or false alarm (4.13) as the cost function to be minimized, which is equivalent to maximize the *Network Achievable Range* defined in (4.6), with the constraints of *primary conflicts* as well as *repetition*. We focused on heuristic solution methods and considered greedy and ASA algorithms. Compared with the greedy algorithm, the ASA algorithm mimics the physical annealing process to conduct the random search, and avoids being trapped in local minima by accepting worse solutions in a controlled way. We use the vertex colouring method to generate the initial solution, and the *generating* procedure is modified to generate the new solution under the constraints of *repetition* and *primary conflict*. In Chapter 5, we also explained how to set the ASA parameters for the specific grid network, and verified its effectiveness.

Numerical results for a grid network structure in Chapter 5 show how the spectrum sensing performance is influenced by system parameters, and also present a comparison of the ASA and greedy algorithms. For each network setting, we presented the scheduling results, values of the upper and lower bounds of Q_d and Q_f, and drew the ROC curve to evaluate the cooperative spectrum sensing performance. For the same network setting, the value of the cost function (4.13) resulted by the ASA algorithm is lower than the greedy algorithm. The ROC curves in Fig. 5.5 – Fig. 5.9 show that using the *OR* decision fusion rule, the difference between the curves of two algorithms is very small in relatively high Q_f range, and advantage of the ASA algorithm can be seen when Q_f is around and smaller than the lower bound $Q_f^{[\mathrm{L}]}$ of the greedy algorithm. The ROC curves in Fig. 5.10 show that the separation between the ROC curves of the ASA and greedy algorithms decreases as the decision fusion factor Ω increases, and when $\Omega = 1$, it is hard to differentiate the two curves in Fig. 5.10.

When comparing the ROC curves of one algorithm with different network settings in each figure, we can see the influence of some system parameters. Fig. 5.5 shows that higher received SNR of the primary signal $\bar{\gamma}$ results in better spectrum sensing performance, but the upper and lower bounds $Q_d^{[\mathrm{U}]}$, $Q_f^{[\mathrm{U}]}$, $Q_d^{[\mathrm{L}]}$ and $Q_f^{[\mathrm{L}]}$ do not change with $\bar{\gamma}$. The results of Fig. 5.6 show that our spatial reuse MAC protocol can save T-F slots without significant impact on the sensing performance at certain Q_f. The lower bound $Q_d^{[\mathrm{L}]}$ and $Q_f^{[\mathrm{L}]}$ decreases with increasing number of T-F slots. When the target Q_f decreases, more T-F slots are needed to achieve it. Fig. 5.7 shows that cooperation between sensors helps improve the spectrum sensing performance when Q_f is higher than the lower bound $Q_f^{[\mathrm{L}]}$. However, the lower bound of Q_f increases with the number of *partners* of each sensor. For example, in this case, the difference between the ROC curves of *CL4* and *CL8* is nearly negligible when $Q_f > 0.07$, and $Q_f < 0.07$ is not achievable by *CL8*. Therefore, we should not expect the number of *partners* of each sensor to be as large as possible. Fig. 5.8 shows that if the number of T-F slots is fixed, larger network size leads to an increasing lower bound of Q_f, and Fig. 5.9 shows that if the ratio M/N is fixed, a larger network size gives better sensing performance. In Fig. 5.10, we plotted the ROC curves for different values of fusion factor Ω, and it is not clear cut which Ω is the best choice in general. We can use the ROC curves to decide the value of Ω according to the specific required Q_f and Q_d. For instance, if the system required Q_f is 0.001 in this case, $\Omega = 0.5$ is the best choice. In addition, if the number of T-F slots cannot be changed for a certain network size, adjusting the *cooperation* *level* or decision fusion threshold is necessary to improve the sensing performance subject to Q_f and Q_d requirements. The upper and lower bounds of Q_f and Q_d are not affected by the local spectrum sensing performance (i.e. $Q_{d,i}$ and $Q_{f,i}$ of each sensor). So in the design procedure, we can first just calculate the bounds Q_f and Q_d for each parameter setting, instead of plotting all the points on the ROC curves. Then we can exclude the cases not satisfying the system requirements to save time, and analyse the performance via ROC curves in the remaining cases. Take the results in Fig. 5.10 for example. If the requirements are $Q_f < 0.001$ and $Q_d > 0.999$, we can first exclude $\Omega = 0.1$ (because its $Q_f^{[\mathrm{L}]} = 0.001366718 > 0.001$) and $\Omega = 1$ (because $Q_d^{[\mathrm{U}]} = 0.988633282 < 0.999$), without plotting their ROC curves.

For future research, since the spatial reuse MAC protocol in this work is not constrained to the grid network form, it would be interesting to implement the MAC protocol in other network forms. For example, sensors can be located on a hexagonal or even irregular net, and the corresponding spectrum sensing performance can be analysed in a similar way. Future research efforts may also include designing through other heuristic methods for solving the combinatorial optimization problem of the MAC protocol. *Genetic Algorithms*, *Ant System*, and *Neural Networks* are possible candidates, and we can continue to compare the effectiveness and complexity of different methods. Moreover, there will be other cost functions than the *Network Achievable Range Loss* chosen in this work, and designing the cost function for the spatial reuse MAC protocol will be a meaningful future research topic. We can analyse how the cost function will influence the shape of the ROC curve, and consider changing the cost function to make it adaptive to specific required probabilities of detection and false alarm.

Appendix A

Proof of Property 1

Proof.

Lemma 1. *For a given bandwidth-product u and $f_{\text{Ric}}(\gamma)$ with fixed parameters, P_d is a concave function of P_f.*

Proof of Lemma 1 is given in [57, Theorem 1], and we use $v(\cdot)$ to represent the relation between P_d and P_f, i.e. $P_d = v(P_f)$, and $v(\cdot)$ is concave function. In [57], $P_d = v(P_f)$ is shown to be differentiable.

From (3.27) and (3.28) we have

$$P_{d,\varepsilon} = P_d - 2\varepsilon P_d + \varepsilon = (1 - 2\varepsilon)P_d + \varepsilon \tag{A.1}$$

$$P_{f,\varepsilon} = P_f - 2\varepsilon P_f + \varepsilon = (1 - 2\varepsilon)P_f + \varepsilon. \tag{A.2}$$

We define the function $u_\varepsilon(\cdot)$ to represent the relation between $P_{d,\varepsilon}$ and P_d. From (A.1) we have

$$P_{d,\varepsilon} = u_\varepsilon(P_d) = (1 - 2\varepsilon)P_d + \varepsilon. \tag{A.3}$$

Thus, $P_{d,\varepsilon} = u_\varepsilon(P_d)$ is a linear function of P_d, and we have

$$\frac{\mathrm{d}P_{d,\varepsilon}}{\mathrm{d}P_d} = 1 - 2\varepsilon. \tag{A.4}$$

We define the function $w_\varepsilon(\cdot)$ to represent the relation between P_f and $P_{f,\varepsilon}$. From (A.2) we have

$$P_f = w_\varepsilon(P_{f,\varepsilon}) = \frac{P_{f,\varepsilon} - \varepsilon}{1 - 2\varepsilon} = \frac{1}{1 - 2\varepsilon} P_{f,\varepsilon} - \frac{\varepsilon}{1 - 2\varepsilon}. \tag{A.5}$$

Thus, $P_f = w_\varepsilon(P_{f,\varepsilon})$ is a linear function of $P_{f,\varepsilon}$, and we have

$$\frac{\mathrm{d}P_f}{\mathrm{d}P_{f,\varepsilon}} = \frac{1}{1 - 2\varepsilon}. \tag{A.6}$$

We have shown that $u_\varepsilon(\cdot)$ and $w_\varepsilon(\cdot)$ are linear functions, and $v(\cdot)$ is a differentiable concave function. Thus, we can use the chain rule to compute the derivative of $P_{d,\varepsilon}$ with respect to $P_{f,\varepsilon}$ at the point $P_{f,\varepsilon} = \alpha$ as follows

$$\begin{aligned}
\left.\frac{\mathrm{d}P_{d,\varepsilon}}{\mathrm{d}P_{f,\varepsilon}}\right|_{P_{f,\varepsilon}=\alpha} &= \left.\frac{\mathrm{d}P_{d,\varepsilon}}{\mathrm{d}P_d}\right|_{P_d=v(w_\varepsilon(\alpha))} \cdot \left.\frac{\mathrm{d}P_d}{\mathrm{d}P_f}\right|_{P_f=w_\varepsilon(\alpha)} \cdot \left.\frac{\mathrm{d}P_f}{\mathrm{d}P_{f,\varepsilon}}\right|_{P_{f,\varepsilon}=\alpha} \\
&= (1 - 2\varepsilon) \cdot \left.\frac{\mathrm{d}P_d}{\mathrm{d}P_f}\right|_{P_f=w_\varepsilon(\alpha)} \cdot \frac{1}{1 - 2\varepsilon} \\
&= \left.\frac{\mathrm{d}P_d}{\mathrm{d}P_f}\right|_{P_f=w_\varepsilon(\alpha)} \\
&= \left.\frac{\mathrm{d}P_d}{\mathrm{d}P_f}\right|_{P_f=\frac{\alpha-\varepsilon}{1-2\varepsilon}}.
\end{aligned} \tag{A.7}$$

When $P_{f,\varepsilon_1} = P_{f,\varepsilon_2} = \alpha$, $(\varepsilon_2 < \alpha < 1 - \varepsilon_2)$, we use

$$f_1 = w_{\varepsilon_1}(\alpha) = \frac{\alpha - \varepsilon_1}{1 - 2\varepsilon_1}, \qquad f_2 = w_{\varepsilon_2}(\alpha) = \frac{\alpha - \varepsilon_2}{1 - 2\varepsilon_2} \tag{A.8}$$

to simplify the representation, and have

$$\begin{aligned} f_1 - f_2 &= \frac{(\alpha - \varepsilon_1)(1 - 2\varepsilon_2) - (\alpha - \varepsilon_2)(1 - 2\varepsilon_1)}{(1 - 2\varepsilon_1)(1 - 2\varepsilon_2)} \\ &= \frac{\alpha - \varepsilon_1 - 2\alpha\varepsilon_2 + 2\varepsilon_1\varepsilon_2 - \alpha + \varepsilon_2 + 2\alpha\varepsilon_1 - 2\varepsilon_1\varepsilon_2}{(1 - 2\varepsilon_1)(1 - 2\varepsilon_2)} \\ &= \frac{-\varepsilon_1 - 2\alpha\varepsilon_2 + \varepsilon_2 + 2\alpha\varepsilon_1}{(1 - 2\varepsilon_1)(1 - 2\varepsilon_2)} \\ &= \frac{(1 - 2\alpha)(\varepsilon_2 - \varepsilon_1)}{(1 - 2\varepsilon_1)(1 - 2\varepsilon_2)}. \end{aligned} \tag{A.9}$$

As $0 < \varepsilon_1 < \varepsilon_2 < 0.5$,

$$\frac{\varepsilon_2 - \varepsilon_1}{(1 - 2\varepsilon_1)(1 - 2\varepsilon_2)} > 0$$

From Lemma 1, $P_d = v(P_f)$ is a concave function of P_f, and as shown in [57], the derivative function $v'(P_f) = \frac{\mathrm{d}P_d}{\mathrm{d}P_f}$ is monotonically decreasing, i.e.

$$\text{If } f_1 > f_2, \text{ then } v'(f_1) = \frac{\mathrm{d}P_d}{\mathrm{d}P_f}\Big|_{P_f = f_1} < v'(f_2) = \frac{\mathrm{d}P_d}{\mathrm{d}P_f}\Big|_{P_f = f_2}. \tag{A.10}$$

Therefore, according to (A.7), (A.8), (A.9), and Lemma 1, we have

1) $\alpha < 0.5 \Longrightarrow 1 - 2\alpha > 0 \Longrightarrow f_1 > f_2$, then

$$\frac{\mathrm{d}P_{d,\varepsilon_1}}{\mathrm{d}P_{f,\varepsilon_1}}\Big|_{P_{f,\varepsilon_1}=\alpha} = \frac{\mathrm{d}P_d}{\mathrm{d}P_f}\Big|_{P_f = f_1} < \frac{\mathrm{d}P_d}{\mathrm{d}P_f}\Big|_{P_f = f_2} = \frac{\mathrm{d}P_{d,\varepsilon_2}}{\mathrm{d}P_{f,\varepsilon_2}}\Big|_{P_{f,\varepsilon_2}=\alpha}.$$

2) $\alpha = 0.5 \Longrightarrow 1 - 2\alpha = 0 \Longrightarrow f_1 = f_2$, then

$$\frac{\mathrm{d}P_{d,\varepsilon_1}}{\mathrm{d}P_{f,\varepsilon_1}}\Big|_{P_{f,\varepsilon_1}=\alpha} = \frac{\mathrm{d}P_d}{\mathrm{d}P_f}\Big|_{P_f = f_1} = \frac{\mathrm{d}P_d}{\mathrm{d}P_f}\Big|_{P_f = f_2} = \frac{\mathrm{d}P_{d,\varepsilon_2}}{\mathrm{d}P_{f,\varepsilon_2}}\Big|_{P_{f,\varepsilon_2}=\alpha}.$$

3) $\alpha > 0.5 \Longrightarrow 1 - 2\alpha < 0 \Longrightarrow f_1 < f_2$, then

$$\frac{\mathrm{d}P_{d,\varepsilon_1}}{\mathrm{d}P_{f,\varepsilon_1}}\Big|_{P_{f,\varepsilon_1}=\alpha} = \frac{\mathrm{d}P_d}{\mathrm{d}P_f}\Big|_{P_f = f_1} > \frac{\mathrm{d}P_d}{\mathrm{d}P_f}\Big|_{P_f = f_2} = \frac{\mathrm{d}P_{d,\varepsilon_2}}{\mathrm{d}P_{f,\varepsilon_2}}\Big|_{P_{f,\varepsilon_2}=\alpha}.$$

Therefore,

$$\frac{\partial \Phi_d^{(\varepsilon_1,\varepsilon_2,\alpha)}}{\partial \alpha} = \frac{\mathrm{d}P_{d,\varepsilon_1}}{\mathrm{d}P_{f,\varepsilon_1}}\Big|_{P_{f,\varepsilon_1}=\alpha} - \frac{\mathrm{d}P_{d,\varepsilon_2}}{\mathrm{d}P_{f,\varepsilon_2}}\Big|_{P_{f,\varepsilon_2}=\alpha} \begin{cases} > 0 & \text{for} \quad \alpha > 0.5 \\ = 0 & \text{for} \quad \alpha = 0.5 \\ < 0 & \text{for} \quad \alpha < 0.5 \end{cases} \tag{A.11}$$

where $\Phi_d^{(\varepsilon_1,\varepsilon_2,\alpha)} = P_{d,\varepsilon_1}|_{P_{f,\varepsilon_1}=\alpha} - P_{d,\varepsilon_2}|_{P_{f,\varepsilon_2}=\alpha}$ as defined in (3.29), and $P_{d,\varepsilon}|_{P_{f,\varepsilon}=\alpha}$ represents the value of $P_{d,\varepsilon}$ when $P_{f,\varepsilon} = \alpha$ as described in Section 3.3.1.

(A.11) shows that $\Phi_d^{\varepsilon_1,\varepsilon_2,\alpha}$ decreases with α when $\alpha < 0.5$, and increases with α when $\alpha > 0.5$. Therefore, the first half of Property 1 is proved, and the next step is to prove the second half, which is $\Phi_d^{\varepsilon_1,\varepsilon_2,\alpha} > 0$.

From (A.11), $\Phi_d^{\varepsilon_1,\varepsilon_2,\alpha}$ is decreasing with α when $\alpha < 0.5$, and increasing with α when $\alpha > 0.5$, so when $\alpha = 0.5$, $\Phi_d^{\varepsilon_1,\varepsilon_2,\alpha}$ reaches its minimum value, i.e.

$$\Phi_d^{\varepsilon_1,\varepsilon_2,\alpha} \geqslant \Phi_d^{\varepsilon_1,\varepsilon_2,0.5}. \tag{A.12}$$

Therefore, if we prove that $\Phi_d^{\varepsilon_1,\varepsilon_2,0.5} > 0$, then we can have $\Phi_d^{\varepsilon_1,\varepsilon_2,\alpha} \geqslant \Phi_d^{\varepsilon_1,\varepsilon_2,0.5} > 0$.

Using the following notations which have been defined before

$$P_{d,\varepsilon} = u_\varepsilon(P_d), \quad P_d = v(P_f), \text{ and } P_f = w_\varepsilon(P_{f,\varepsilon}),$$

we can obtain

$$\begin{aligned} P_{d,\varepsilon}|_{P_{f,\varepsilon}=\alpha} &= u_\varepsilon(P_d) \\ &= u_\varepsilon[v(P_f)] \\ &= u_\varepsilon\{v[w_\varepsilon(\alpha)]\} \\ &= (1-2\varepsilon)v[w_\varepsilon(\alpha)] + \varepsilon. \end{aligned} \tag{A.13}$$

When $\alpha = 0.5$,

$$w_\varepsilon(\alpha) = \frac{0.5-\varepsilon}{1-2\varepsilon} = 0.5,$$

which does not depend on ε. Then according to (A.13), we obtain

$$\begin{aligned}
\Phi_d^{(\varepsilon_1,\varepsilon_2,0.5)} &= P_{d,\varepsilon_1}|_{P_{f,\varepsilon_1}=0.5} - P_{d,\varepsilon_2}|_{P_{f,\varepsilon_2}=0.5} \\
&= (1-2\varepsilon_1)v(0.5) + \varepsilon_1 - (1-2\varepsilon_2)v(0.5) - \varepsilon_2 \\
&= (\varepsilon_2 - \varepsilon_1)(2v(0.5)-1) \qquad \text{(A.14)}
\end{aligned}$$

According to (3.25) and (3.26) we get, $P_d = P_f = 1$ when $\lambda = 0$, and $P_d = P_f = 0$ when $\lambda \to \infty$, i.e. $(0,0)$, $(1,1)$ are two points on the curve $P_d = v(P_f)$. Moreover, according to Lemma 1, $P_d = v(P_f)$ is a concave function. Thus, we can get $v(0.5) > \frac{v(0)+v(1)}{2} = 0.5$, and then $2v(0.5) - 1 > 0$. Therefore, according to (A.14),

$$\Phi_d^{(\varepsilon_1,\varepsilon_2,0.5)} = P_{d,\varepsilon_1}|_{P_{f,\varepsilon_1}=0.5} - P_{d,\varepsilon_2}|_{P_{f,\varepsilon_2}=0.5} > 0. \qquad \text{(A.15)}$$

Then according to (A.12),

$$\Phi_d^{(\varepsilon_1,\varepsilon_2,\alpha)} \geqslant \Phi_d^{(\varepsilon_1,\varepsilon_2,0.5)} > 0.$$

So the second half of Property 1 is proved.

□

Appendix B

Lemma 2

We first introduce the definition of Poisson-Binomial distribution. The Poisson-Binomial distribution is the discrete probability distribution of a sum of independent Bernoulli trials when the individual probabilities of success are not necessarily identical [34, 35]. Consider N independent Bernoulli trials, where for the mth trial X_m the probability of success is denoted by p_m, i.e. $\text{Prob}\{X_m = 1\} = p_m$ and $\text{Prob}\{X_m = 0\} = 1 - p_m$. The probability of at least k successes in these N trials is the complement of Poisson-Binomial CDF as defined in [35, Eq. (8)], and a closed-form expression is given in [35, Eq. (11)],

$$\begin{aligned} Q(k) &= \text{Prob}\{\text{At least } k \text{ successes in } N \text{ Bernoulli trials}\} && \text{(B.1)} \\ &= 1 - \frac{1}{N+1}\sum_{n=0}^{N}\left\{\frac{1-\exp[-\mathbf{j}2\pi nk/(N+1)]}{1-\exp[-\mathbf{j}2\pi n/(N+1)]}\prod_{m=1}^{N}\big(1-p_m+p_m\exp[\mathbf{j}2\pi n/(N+1)]\big)\right\}. && \text{(B.2)} \end{aligned}$$

In addition, the closed-form expression for Poisson-Binomial probability mass function

(PMF) is given in [35, Eq. (5)],

$$
\begin{aligned}
P(k) &= \text{Prob}\{\text{Exactly } k \text{ successes in } N \text{ Bernoulli trials}\} && \text{(B.3)}\\
&= \frac{1}{N+1}\sum_{n=0}^{N}\left\{\exp[-\mathbf{j}2\pi nk/(N+1)]\prod_{m=1}^{N}\big(1-p_m+p_m\exp[\mathbf{j}2\pi n/(N+1)]\big)\right\}. && \text{(B.4)}
\end{aligned}
$$

Lemma 2. *For a certain series of N Bernoulli trials, and fixed value of k, the complement of Poisson-Binomial CDF, $Q(k)$ of (B.1) is monotonic non-decreasing in each $p_m, m = 1, 2, \ldots, N$.*

Proof. We first find the partial derivative of $Q(k)$ with respect to $p_l, l = 1, 2, \ldots, N$. Based on (B.2), we can obtain

$$
\begin{aligned}
\frac{\partial Q(k)}{\partial p_l} &= -\frac{1}{N+1}\sum_{n=0}^{N}\left\{\frac{1-\exp[-\mathbf{j}2\pi nk/(N+1)]}{1-\exp[-\mathbf{j}2\pi n/(N+1)]}\big(\exp[\mathbf{j}2\pi n/(N+1)]-1\big)\right.\\
&\qquad \left.\prod_{1\leqslant m\leqslant N, m\neq l}\big(1-p_m+p_m\exp[\mathbf{j}2\pi n/(N+1)]\big)\right\}\\
&= -\frac{1}{N+1}\sum_{n=0}^{N}\left\{\frac{1-\exp[-\mathbf{j}2\pi nk/(N+1)]}{1-\exp[-\mathbf{j}2\pi n/(N+1)]}\big(1-\exp[-\mathbf{j}2\pi n/(N+1)]\big)\exp[\mathbf{j}2\pi n/(N+1)]\right.\\
&\qquad \left.\prod_{1\leqslant m\leqslant N, m\neq l}\big(1-p_m+p_m\exp[\mathbf{j}2\pi n/(N+1)]\big)\right\}\\
&= \frac{1}{N+1}\sum_{n=0}^{N}\left\{\big(\exp[-\mathbf{j}2\pi nk/(N+1)]-1\big)\exp[\mathbf{j}2\pi n/(N+1)]\right.\\
&\qquad \left.\prod_{1\leqslant m\leqslant N, m\neq l}\big(1-p_m+p_m\exp[\mathbf{j}2\pi n/(N+1)]\big)\right\}. \qquad \text{(B.5)}
\end{aligned}
$$

Then we consider a certain series of N Bernoulli trials where the success probability of the lth trial is set to 1, i.e. $\text{Prob}\{X_l = 1\} = p_l = 1$. We use $P^{[p_l=1]}(k)$ to denote the

Poisson-Binomial PMF in this case, and based on (B.4) we get

$$\begin{aligned}P^{[p_l=1]}(k) &= \frac{1}{N+1}\sum_{n=0}^{N}\left\{\exp[-\mathbf{j}2\pi nk/(N+1)]\prod_{m=1}^{N}\big(1-p_m+p_m\exp[\mathbf{j}2\pi n/(N+1)]\big)\right\}\\ &= \frac{1}{N+1}\sum_{n=0}^{N}\Big\{\exp[-\mathbf{j}2\pi nk/(N+1)]\exp[\mathbf{j}2\pi n/(N+1)]\\ &\qquad \prod_{1\leqslant m\leqslant N, m\neq l}\big(1-p_m+p_m\exp[\mathbf{j}2\pi n/(N+1)]\big)\Big\}. \end{aligned} \tag{B.6}$$

Moreover, when $k=0$, according to (B.6) we have

$$P^{[p_l=1]}(0) = \frac{1}{N+1}\sum_{n=0}^{N}\left\{\exp[\mathbf{j}2\pi n/(N+1)]\prod_{1\leqslant m\leqslant N, m\neq l}\big(1-p_m+p_m\exp[\mathbf{j}2\pi n/(N+1)]\big)\right\} \tag{B.7}$$

Based on (B.6) and (B.7), we get

$$\begin{aligned}P^{[p_l=1]}(k)-P^{[p_l=1]}(0) = \frac{1}{N+1}\sum_{n=0}^{N}\Big\{&\big(\exp[-\mathbf{j}2\pi nk/(N+1)]-1\big)\exp[\mathbf{j}2\pi n/(N+1)]\\ &\prod_{1\leqslant m\leqslant N, m\neq l}\big(1-p_m+p_m\exp[\mathbf{j}2\pi n/(N+1)]\big)\Big\}.\end{aligned} \tag{B.8}$$

The right-hand side of (B.8) is exactly the same as that of (B.5). Thus, we have

$$\frac{\partial Q(k)}{\partial p_l} = P^{[p_l=1]}(k)-P^{[p_l=1]}(0). \tag{B.9}$$

When $p_l=1$, the lth trial is successful with probability 1, and hence $P^{[p_l=1]}(0)=0$. Then, we can conclude that

$$\frac{\partial Q(k)}{\partial p_l} = P^{[p_l=1]}(k)-P^{[p_l=1]}(0) = P^{[p_l=1]}(k)\geqslant 0, \quad l=1,2,\ldots,N. \tag{B.10}$$

making $Q(k)$ monotonic non-decreasing in each $p_l, l=1,2,\ldots,N$. □

Appendix C

Proof of Property 7

Proof. The expressions for $Q^{[\mathrm{L}]}_{d,i}$ and $Q^{[\mathrm{U}]}_{d,i}$ are given in (3.33) and (3.34), i.e.

$$Q^{[\mathrm{L}]}_{d,i}(k_i) = \sum_{m=k_i}^{n_i} \sum_{l=1}^{\binom{n_i}{m}} \left(\prod_{s_j \in \mathcal{J}^{(i)}_{m,l}} \varepsilon_{j,i} \prod_{s_j \in \mathcal{J}^{(i)} \backslash \mathcal{J}^{(i)}_{m,l}} \big(1 - \varepsilon_{j,i}\big) \right) \tag{C.1}$$

$$Q^{[\mathrm{U}]}_{d,i}(k_i) = \sum_{m=k_i}^{n_i} \sum_{l=1}^{\binom{n_i}{m}} \left(\prod_{s_j \in \mathcal{J}^{(i)}_{m,l}} \big(1 - \varepsilon_{j,i}\big) \prod_{s_j \in \mathcal{J}^{(i)} \backslash \mathcal{J}^{(i)}_{m,l}} \varepsilon_{j,i} \right) \tag{C.2}$$

where $\mathcal{J}^{(i)} = \{s_j | R_{j,i} = 1\}$ is the *partner set* of s_i, $n_i = |\mathcal{J}^{(i)}|$ is the number of *partners*, $\mathcal{J}^{(i)}_{m,l}, 1 \leqslant l \leqslant \binom{n_i}{m}$ is one m-combination of $\mathcal{J}^{(i)}$, and $\mathcal{J}^{(i)} \backslash \mathcal{J}^{(i)}$ is the relative complement of $\mathcal{J}^{(i)}_{m,l}$ in $\mathcal{J}^{(i)}$. Please note that all the notations above are the same as those defined in Section 3.2.

Suppose there are n_i independent Bernoulli trials, and the success probability of each trial is $\varepsilon_{j,i}$, and the failure probability of each trial is $1 - \varepsilon_{j,i}$. Thus, $Q^{[\mathrm{L}]}_{d,i}(k_i)$ in (C.1) can be viewed as the probability that at least k_i of these n_i trials are successful, i.e.

$$Q^{[\mathrm{L}]}_{d,i}(k_i) = \mathrm{Prob}\{\text{Number of successes } \geqslant k_i\}, \tag{C.3}$$

and $Q^{[\mathrm{U}]}_{d,i}(k_i)$ in (C.2) can be viewed as the probability that at least k_i of these n_i trials are failed, i.e.

$$\begin{aligned} Q^{[\mathrm{U}]}_{d,i}(k_i) &= \mathrm{Prob}\{\text{Number of failures } \geqslant k_i\} \\ &= \mathrm{Prob}\{\text{Number of successes } \leqslant n_i - k_i\} \\ &= 1 - \mathrm{Prob}\{\text{Number of successes} \geqslant n_i - k_i + 1\} \end{aligned} \tag{C.4}$$

where (C.4) is also given in [58, pp. 251-253]. Thus, based on (C.3) and (C.4), it is obvious that

$$Q^{[\mathrm{U}]}_{d,i}(k_i) = 1 - Q^{[\mathrm{L}]}_{d,i}(n_i - k_i + 1). \tag{C.5}$$

Then, based on (C.5) and the definition of *Achievable Range* $\Delta_{d,i}(k_i)$ in (3.37), we can have

$$\Delta_{d,i}(k_i) \triangleq Q^{[\mathrm{U}]}_{d,i}(k_i) - Q^{[\mathrm{L}]}_{d,i}(k_i) = 1 - Q^{[\mathrm{L}]}_{d,i}(n_i - k_i + 1) - Q^{[\mathrm{L}]}_{d,i}(k_i) \tag{C.6}$$

and

$$\begin{aligned} \Delta_{d,i}(n_i - k_i + 1) &\triangleq Q^{[\mathrm{U}]}_{d,i}(n_i - k_i + 1) - Q^{[\mathrm{L}]}_{d,i}(n_i - k_i + 1) \\ &= 1 - Q^{[\mathrm{L}]}_{d,i}(n_i - (n_i - k_i + 1) + 1) - Q^{[\mathrm{L}]}_{d,i}(n_i - k_i + 1) \\ &= 1 - Q^{[\mathrm{L}]}_{d,i}(k_i) - Q^{[\mathrm{L}]}_{d,i}(n_i - k_i + 1). \end{aligned} \tag{C.7}$$

Therefore, comparing (C.6) and (C.7) we can conclude that $\Delta_{d,i}(k_i) = \Delta_{d,i}(n_i - k_i + 1)$.

□

Appendix D

Details of ASA

The ASA algorithm [51] is a probabilistic optimization method for an N-dimensional parameter space. It permits an exponentially decreasing annealing schedule which greatly accelerates the optimization process compared with some traditional simulated annealing algorithms (e.g. *Boltzmann Annealing* and *Fast Cauchy Annealing* [53]). Moreover, the re-annealing procedure is introduced in the ASA to adapt to changing sensitivities in the multi-dimensional parameter-space.

The basic structure of the ASA algorithm consists of three major procedures: *generating*, *acceptance*, and *annealing* procedures. In a N-dimensional parameter space with the ith parameter having the range $[L_i, U_i]$, assuming that $\mathbf{p}^{(t)} \in \mathbb{R}^N$ is the tth last saved point, the *generating* procedure aims to obtain a new *candidate* point $\hat{\mathbf{p}}^{(t+1)}$ for the $(t+1)$th state. The components of $\hat{\mathbf{p}}^{(t+1)}$ are determined by

$$\hat{p}_i^{(t+1)} = p_i^{(t)} + y_i(U_i - L_i), \tag{D.1}$$

where $y_i \in [-1, 1]$ is a sample of a random variable Y_i with the following PDF,

$$g_i(y_i; T_i(t_i)) = \frac{1}{2(|y_i| + T_i(t_i)) \ln(1 + 1/T_i(t_i))}. \tag{D.2}$$

$T_i(t_i)$ in this equation is defined as the *parameter temperature*, and t_i is called the *parameter annealing index*, which will be illustrated later. If $\hat{p}_i^{(t+1)} \notin [L_i, U_i]$, a new $\hat{p}_i^{(t+1)}$ is generated with another sample y_i of Y_i, until $\hat{p}_i^{(t+1)} \in [L_i, U_i]$.

The *acceptance* procedure determines whether the *candidate* point $\hat{\mathbf{p}}^{(t+1)}$ is accepted as the new saved point for the next state. When the cost function is denoted by $\zeta(\cdot)$, and $u \in [0, 1)$ is a sample of a uniformly distributed random variable, the *acceptance* procedure is

$$\mathbf{p}^{(t+1)} = \begin{cases} \hat{\mathbf{p}}^{(t+1)} & \text{if} \quad \exp\left[-\left(\zeta(\hat{\mathbf{p}}^{(t+1)}) - \zeta(\mathbf{p}^{(t+1)})\right)/T_{\text{cost}}(t_{\text{cost}})\right] > u \\ \mathbf{p}^{(t)} & \text{if} \quad \exp\left[-\left(\zeta(\hat{\mathbf{p}}^{(t+1)}) - \zeta(\mathbf{p}^{(t+1)})\right)/T_{\text{cost}}(t_{\text{cost}})\right] \leqslant u \end{cases} \tag{D.3}$$

where $T_{\text{cost}}(t_{\text{cost}})$ is the *cost temperature*, and t_{cost} is called *cost annealing index*, which will be discussed later.

The *parameter temperature* $T_i(t_i)$ associated with the *generating* procedure, and the *cost temperature* $T_{\text{cost}}(t_{\text{cost}})$ associated with the *acceptance* procedure are controlled by the *annealing* procedures. Specifically, the annealing procedure for $T_i(t_i)$ is

$$T_i(t_i) = T_i(0) \exp\left(-c_{\text{para}} t_i^{Q_i/N}\right) \tag{D.4}$$

In (D.4), the *initial parameter temperature* $T_i(0)$ is set by users, and usually set to 1. The *parameter annealing index* t_i is initially set to 0, and increased by 1 every time a new *candidate* point is generated. Q_i is the *parameter quenching factor* for the ith parameter, which is set to 1 for normal ASA. When $Q_i > 1$, the "quenching" option in ASA is turned on. The annealing speed is faster than the normal ASA, and the search procedure cannot be proved to converge to the global optimum with probability 1. However, turning on the "quenching" option can be extremely useful in speeding up the search, and sometimes necessary in a large parameter space [51]. c_{para} denotes the *parameter temperature scale*, which is determined by

$$c_{\text{para}} = -\ln(\alpha_1) \exp\left(-\ln(\alpha_2)/N\right), \tag{D.5}$$

where α_1 and α_2 are called the *temperature ratio scale* and *temperature anneal scale* respectively.

The *annealing* procedure for the *cost temperature* $T_{\text{cost}}(t_{\text{cost}})$ is

$$T_{\text{cost}}(t_{\text{cost}}) = T_{\text{cost}}(0) \exp\left(-c_{\text{cost}} t_{\text{cost}}^{Q_{\text{cost}}/N}\right) \tag{D.6}$$

which is very similar to that for the *parameter temperature* $T_i(t_i)$. However, In (D.6), the *initial cost temperature* $T_{\text{cost}}(0)$ is not set as a constant by users. The ASA algorithm first generates N_{spl} states and calculates the cost function of each state, where N_{spl} is named *number of cost samples*. Then, $T_{\text{cost}}(0)$ takes the average of the absolute values of these N_{spl} sampled cost functions. The *cost annealing index* t_{cost} is increased by 1 every time a new *candidate* point is accepted, after initially set to 0. Q_{cost} is called the *cost quenching factor*, which is similar to Q_{i}, but will not affect the proof that ASA converges asymptotically to the global optimum. The *cost temperature scale* c_{cost} is set as $c_{\text{cost}} = \beta c_{\text{para}}$, where β is an ASA parameter named *cost parameter scale ratio*.

An optional *reannealing* procedure in ASA will periodically rescale the *parameter temperatures* as well as *cost temperature* to adapt to different sensitivities among parameter dimensions and current status of cost function. This procedure takes place every m_{gen} (*generated frequency modulus*) *candidate* points generated, or every m_{acc} (*acceptance frequency modulus*) *candidate* points accepted. Assuming that this *reannealing* procedure is conducted after t states are generated, $T_i(t_i)$ is reannealed as

$$T_i(t_i) \leftarrow \big(\max_{1 \leqslant i \leqslant N}\{f_i\}/f_i\big) T_i(t_i), \text{ with } f_i = \left| \frac{\partial \zeta(\mathbf{p})}{\partial p_i}\bigg|_{\acute{\mathbf{p}}^{(t)}} \right| \tag{D.7}$$

where $\acute{\mathbf{p}}^{(t)}$ corresponds to the best cost value found as of the current state. If this new value of $T_i(t_i)$ is less than $T_i(0)$, the *parameter annealing index* t_i is reset as

$$t_i \leftarrow \left(\frac{|\ln T_i(0) - \ln T_i(t_i)|}{c_{\text{para}}} \right)^{N/Q_i} \tag{D.8}$$

Moreover, the *initial cost temperature* $T_{\text{cost}}(0)$ is rescaled as

$$T_{\text{cost}}(0) \leftarrow \min \left\{ T_{\text{cost}}(0), \max\{ |\zeta(\mathbf{p}^{(t-1)})|, |\zeta(\acute{\mathbf{p}}^{(t)})|, |\zeta(\mathbf{p}^{(t-1)}) - \zeta(\acute{\mathbf{p}}^{(t)})| \} \right\} \tag{D.9}$$

When $T_{\text{cost}}(0)$ gets its new value, the current *cost temperature* is reset as

$$T_{\text{cost}}(t_{\text{cost}}) \leftarrow \min \left\{ T_{\text{cost}}(0), \max\{ T_{\text{cost}}(t_{\text{cost}}), |\zeta(\mathbf{p}^{(t-1)}) - \zeta(\acute{\mathbf{p}}^{(t)})| \} \right\} \tag{D.10}$$

Thus, according to (D.6), the *cost annealing index* after *reannealing* is

$$t_{\text{cost}} \leftarrow \left(\frac{|\ln T_{\text{cost}}(0) - \ln T_{\text{cost}}(t_{\text{cost}})|}{c_{\text{cost}}} \right)^{N/Q_{\text{cost}}} \tag{D.11}$$

Appendix E

Details of C codes and MATLAB scripts

Here we provide an overview of the software that implements the spatial reuse MAC protocol, and produces the numerical results of spectrum sensing performance for the grid network described in Chapter 5. All the necessary C codes and MATLAB scripts are stored in the folder named "Programming_MAC", which can be obtained from the author by request. Table E.1 lists these C and MATLAB files with corresponding descriptions.

Directory "C" contains all C files for the implementation of the spatial reuse MAC protocol via the Greedy and ASA algorithms, which is used to calculate the scheduling result **p**. The GNU Scientific Library (GSL) is necessary for the compilation of the C files, which is available at <http://www.gnu.org>, and we used the version 1.16. The C files are compiled on a Fedora Linux environment using the GCC compiler version 4.8.3 – 20140911. "Greedy_MAC.c" under the subdirectory "Greedy" is the greedy algorithm with *k-distance colouring* initialization method. We have explained in detail how to set the network parameters via the code comments at the beginning of the C file, such as Rician fading factor, number of sensors, cooperation level and so on. Therefore, there is no necessary to list these parameters again. After setting up the corresponding parameters for the desired network environment, we need to access the current directory, and compile the C code by using the following GCC command:

```
gcc -o Greedy_MAC Greedy_MAC.c -lgsl -lgslcblas -lm -g
```

This command creates “Greedy_MAC” as an executable file, which can be run using the following command:

```
./ Greedy_MAC
```

The results are saved in the output file named “greedy_usr_out”, which contains the number of states generated, run time, final value of the cost function, and scheduling result vector.

Subdirectory “ASA” contains the files for the ASA algorithm. The C-language implementation of the ASA algorithm is based on the source code provided by A. L. Ingber, which is available at <http://www.ingber.com/#ASA> since 1993. We made some modifications as described in Section 4.3.2, and integrate the WSN setting into it. The network parameters can be changed in “asa_usr_asa.h”, which are very similar to those in “Greedy_MAC.c”, and also explained via code comments at the beginning of the file. The key ASA parameters and options for our system are summarized in Table 5.2, and others are kept to their default values. The change of ASA parameters, such as the maximum number of generated states, number of cost samples, initial parameter temperature and so on, can be made in the file “asa_opt” and “asa_usr_asa.h”. There are over 100 options provided by the original source code to be tuned, and a guide document named “ASA-README.pdf” is attached, which explains how to change the ASA parameters in detail, and can be downloaded from <http://www.ingber.com/#ASA>.

The “ASA-Makefile” is provided as a template for user’s own Makefile, which can be just copied to Makefile for quick use. To run the ASA algorithm using the GCC compiler after setting up the network and ASA parameters, type the following command:

```
gcc -o ASA_MAC asa_usr.c asa_usr_cst.c asa.c -lgsl -lgslcblas -lm -g
./ ASA_MAC
```

"ASA_MAC" is the created executable file, whose output is saved in "asa_usr_out" and "asa_out". "asa_usr_out" is a concise version, which only contains the exit indicator, final cost value, run time, and the scheduling result vector. "asa_out" contains much more detailed information, including the ASA parameters, initial state, intermediate process etc.

Directory "MATLAB" contains the MATLAB scripts calculating the network averaged probabilities of detection and false alarm after applying the spatial reuse MAC protocol. "f_DecisionFusion" is the function for the *k-out-of-n* decision fusion. The four inputs of "f_DecisionFusion" are the matrix containing $P_{d,j}^{(i)}$ or $P_{f,j}^{(i)}$ (defined in (3.16) and (3.17)), *adjacency matrix* $\mathbf{R}$, number of sensors N, and the fusion factor Ω. The output of "f_DecisionFusion" is the vector containing the probability of detection or false alarm of each sensor after decision fusion, i.e. $Q_{d,i}$ or $Q_{f,i}$ (defined in (3.18) and (3.19)). "MAC_ROC.m" is the main script, and we first set the matrix "mResMac" in "MAC_ROC.m" as the scheduling result obtained from the C files, whose first column contains the sensor IDs, and the second column contains the T-F slot IDs. Then, follow the comments in the scripts to set the network parameters, which must correspond to those used in the C files. Run "MAC_ROC.m", and the results Q_d and Q_f with different detection thresholds λ, will be stored in two column vectors, named "vResQd" and "vResQf".

Table E.1 C and MATLAB files with corresponding descriptions.

File Name	Location	Description
asa.c(.h)	Programming_MAC/C/ASA	The ASA optimization algorithm including generating procedure, acceptance procedure, annealing procedure, etc.
asa_usr.c(.h)	Programming_MAC/C/ASA	The main procedure to solve the optimization problem in the grid network using the ASA algorithm.
asa_usr_cst.c	Programming_MAC/C/ASA	The cost function for our optimization problem.
asa_usr_asa.h	Programming_MAC/C/ASA	The header file containing the macro definitions for ASA and network setting parameters.
asa_opt	Programming_MAC/C/ASA	The file containing the ASA Options.
Greedy_MAC.c	Programming_MAC/C/Greedy	The main procedure to solve the optimization problem in the grid network using the greedy algorithm.
MAC_ROC.m	Programming_MAC/MATLAB	Calculating the network probabilities of detection and false alarm using the scheduling result obtained for the C files.
f_DecisionFusion.m	Programming_MAC/MATLAB	The function for the *k-out-of-n* decision fusion rule.

References

[1] A. Khattab, D. Perkins, and M. A. Bayoumi, *Cognitive Radio Networks: From Theory to Practice.* New York, NY: Springer, 2013.

[2] K. Letaief and W. Zhang, "Cooperative communications for cognitive radio networks," *Proceedings of the IEEE*, vol. 97, no. 5, pp. 878–893, 2009.

[3] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: a survey," *Computer Networks*, vol. 50, no. 13, pp. 2127–2159, Sep. 2006.

[4] S. Haykin, "Cognitive radio: brain-empowered wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 201–220, Feb. 2005.

[5] Z. Lei and S. J. Shellhammer, "IEEE 802.22: The first cognitive radio wireless regional area network standard," *IEEE Commun. Mag.*, vol. 47, no. 1, pp. 130–138, 2009.

[6] C. Cordeiro, K. Challapali, D. Birru, and N. Sai Shankar, "IEEE 802.22: the first worldwide wireless standard based on cognitive radios," in *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on.* IEEE, 2005, pp. 328–337.

[7] F. Abinader Jr, E. P. Almeida, F. S. Chaves, A. M. Cavalcante, R. D. Vieira, R. C. Paiva, A. M. Sobrinho, S. Choudhury, E. Tuomaala, K. Doppler *et al.*, "Enabling the coexistence of LTE and Wi-Fi in unlicensed bands," *Communications Magazine, IEEE*, vol. 52, no. 11, pp. 54–61, 2014.

[8] H. Zhang, X. Chu, W. Guo, and S. Wang, "Coexistence of Wi-Fi and heterogeneous small cell networks sharing unlicensed spectrum," *Communications Magazine, IEEE*, vol. 53, no. 3, pp. 158–164, 2015.

[9] A. Sahai, N. Hoven, and R. Tandra, "Some fundamental limits on cognitive radio," in *Allerton Conference on Communication, Control, and Computing.* Monticello, Illinois, 2004, pp. 1662–1671.

[10] D. Cabric, S. M. Mishra, and R. W. Brodersen, “Implementation issues in spectrum sensing for cognitive radios,” in *Signals, systems and computers, 2004. Conference record of the thirty-eighth Asilomar conference on*, vol. 1. IEEE, 2004, pp. 772–776.

[11] W. A. Gardner and C. M. Spooner, “Signal interception: performance advantages of cyclic-feature detectors,” *Communications, IEEE Transactions on*, vol. 40, no. 1, pp. 149–159, 1992.

[12] Z. Tian and G. B. Giannakis, “A wavelet approach to wideband spectrum sensing for cognitive radios,” in *Cognitive Radio Oriented Wireless Networks and Communications, 2006. 1st International Conference on.* IEEE, 2006, pp. 1–5.

[13] Y. Zeng and Y.-C. Liang, “Covariance based signal detections for cognitive radio,” in *New Frontiers in Dynamic Spectrum Access Networks, 2007. DySPAN 2007. 2nd IEEE International Symposium on.* IEEE, 2007, pp. 202–207.

[14] R. Tandra and A. Sahai, “SNR walls for signal detection,” *Selected Topics in Signal Processing, IEEE Journal of*, vol. 2, no. 1, pp. 4–17, 2008.

[15] T. Yucek and H. Arslan, “A survey of spectrum sensing algorithms for cognitive radio applications,” *Communication Surveys and Tutorials*, vol. 11, no. 1, pp. 116–130, 2009.

[16] D. Duan, L. Yang, and J. C. Principe, “Cooperative diversity of spectrum sensing for cognitive radio systems,” *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 3218–3227, Jun. 2010.

[17] W. Zhang and K. B. Letaief, “Cooperative spectrum sensing with transmit and relay diversity in cognitive radio networks,” *IEEE Trans. Wireless Commun.*, vol. 7, pp. 4761–4766, Dec. 2008.

[18] S. Atapattu, C. Tellambura, and H. Jiang, “Energy detection based cooperative spectrum sensing in cognitive radio networks,” *IEEE Trans. Wireless Commun.*, vol. 10, no. 4, pp. 1232–1241, Apr. 2011.

[19] S. M. Mishra, A. Sahai, and R. Brodersen, “Cooperative sensing among cognitive radios,” in *Proc. IEEE ICC’06*, Jun. 2006, pp. 1658–1664.

[20] Z. Han, R. Fan, and H. Jiang, “Replacement of spectrum sensing in cognitive radio,” *IEEE Trans. Wireless Commun.*, vol. 8, no. 6, pp. 2819–2826, Jun. 2009.

[21] S. N. Shankar, C. Cordeiro, and K. Challapali, “Spectrum agile radios: utilization and sensing architectures,” in *Proc. IEEE DySPAN 2005*, Nov. 2005, pp. 160–169.

[22] A. Sahai, S. M. Mishra, R. Tandra, and N. Hoven, "Sensing for communication: the case of cognitive radio," in *Allerton Conf. on Commun., Control, and Computing,*, 2006, pp. 1035–1037.

[23] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks: A survey," *Physical communication*, vol. 4, no. 1, pp. 40–62, 2011.

[24] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung, "MAC essentials for wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 12, no. 2, pp. 222–248, 2010.

[25] I. Demirkol, C. Ersoy, and F. Alagoz, "MAC protocols for wireless sensor networks: a survey," *IEEE Commun. Mag.*, vol. 06, pp. 115–121, Apr. 2006.

[26] K. I. Kredo and P. Mohapatra, "Medium access control in wireless sensor networks," *Computer Networks*, vol. 51, no. 4, pp. 961–994, Mar. 2007.

[27] S. Chaudhari, J. Lunden, V. Koivunen, and H. V. Poor, "Cooperative sensing with imperfect reporting channels: Hard decisions or soft decisions?" *IEEE Trans. Signal Process.*, vol. 60, no. 1, pp. 18–28, Jan. 2012.

[28] A. Goldsmith, *Wireless Communications.* New York, NY: Cambridge University Press, 2005.

[29] M. K. Simon and M.-S. Alouini, *Digital Communication over Fading Channels*, 2nd ed. Hoboken, NJ: John Wiley & Sons, 2005.

[30] A. Iyer, C. Rosenberg, and A. Karnik, "What is the right model for wireless channel interference," *IEEE Trans. Wireless Commun.*, vol. 8, no. 5, pp. 2662–2671, May 2009.

[31] A. Ghasemi and E. S. Sousa, "Collaborative spectrum sensing for opportunistic access in fading environments." in *New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005. 2005 First IEEE International Symposium on.*, Baltimore, MD, USA, Nov. 2005, pp. 131–136.

[32] F. F. Digham, M.-S. Alouini, and M. K. Simon, "On the energy detection of unknown signals over fading channels." in *Communications, 2003. ICC '03. IEEE International Conference on*, vol. 5, May 2003, pp. 3575 – 3579.

[33] P. K. Varshney, *Distributed Detection and Data Fusion*, C. Burrus, Ed. New York, NY: Springer, 1997.

[34] Y. Hong, "On computing the distribution function for the Poisson binomial distribution," *Computational Statistics and Data Analysis Data Analysis*, vol. 59, pp. 41–51, 2013.

[35] M. Fernandez and S. Williams, "Closed-form expression for the Poisson-Binomial probability density function," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 46, no. 2, pp. 803–817, 2010.

[36] I. Cidon and M. Sidi, "Distributed assignment algorithm for multihop packet radio networks," *IEEE Trans. Comput.*, vol. 38, no. 10, pp. 1353–1361, Oct. 1989.

[37] Y. Sagduyu and A. Ephremides, "The problem of medium access control in wireless sensor networks," *IEEE Wireless Commun. Mag.*, vol. 11 (6), pp. 44–53, Dec. 2004.

[38] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi, *Complexity and approximation: Combinatorial optimization problems and their approximability properties.* Springer Science & Business Media, 2012.

[39] W. K. Lai and G. G. Coghill, "Channel assignment through evolutionary optimization," *Vehicular Technology, IEEE Transactions on*, vol. 45, no. 1, pp. 91–96, 1996.

[40] V. Maniezzo and A. Carbonaro, "An ants heuristic for the frequency assignment problem," *Future Generation Computer Systems*, vol. 16, no. 8, pp. 927–935, 2000.

[41] S. O. Krumke, M. V. Marathe, and S. Ravi, "Models and approximation algorithms for channel assignment in radio networks," *Wireless networks*, vol. 7, no. 6, pp. 575–584, 2001.

[42] T. ElBatt and A. Ephremides, "Joint scheduling and power control for wireless ad hoc networks," *Wireless communications, IEEE Transactions on*, vol. 3, no. 1, pp. 74–85, 2004.

[43] M. Alicherry, R. Bhatia, and L. E. Li, "Joint channel assignment and routing for throughput optimization in multi-radio wireless mesh networks," in *Proceedings of the 11th annual international conference on Mobile computing and networking.* ACM, 2005, pp. 58–72.

[44] A. P. Subramanian, H. Gupta, S. R. Das, and J. Cao, "Minimum interference channel assignment in multiradio wireless mesh networks," *Mobile Computing, IEEE Transactions on*, vol. 7, no. 12, pp. 1459–1473, 2008.

[45] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity.* Courier Corporation, 1998.

[46] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Computing Surveys (CSUR)*, vol. 35, no. 3, pp. 268–308, 2003.

[47] A. Colorni, M. Dorigo, F. Maffioli, V. Maniezzo, G. Righini, and M. Trubian, “Heuristics from nature for hard combinatorial optimization problems,” *International Transactions in Operational Research*, vol. 3, no. 1, pp. 1–21, 1996.

[48] D. Brelaz, “New methods to color the vertices of a graph,” *Communications of the ACM*, vol. 22, no. 4, pp. 251–256, 1979.

[49] G. J. Chaitin, M. A. Auslander, A. K. Chandra, J. Cocke, M. E. Hopkins, and P. W. Markstein, “Register allocation via coloring,” *Computer languages*, vol. 6, no. 1, pp. 47–57, 1981.

[50] C. Avanthay, A. Hertz, and N. Zufferey, “A variable neighborhood search for graph coloring,” *European Journal of Operational Research*, vol. 151, no. 2, pp. 379–388, 2003.

[51] L. Ingber, A. Petraglia, M. R. Petraglia, M. A. S. Machado *et al.*, “Adaptive simulated annealing,” in *Stochastic global optimization and its applications with fuzzy adaptive simulated annealing.* New York, NY: Springer, 2012, pp. 33–62.

[52] D. Bertsimas, J. Tsitsiklis *et al.*, “Simulated annealing,” *Statistical science*, vol. 8, no. 1, pp. 10–15, 1993.

[53] H. Szu and R. Hartley, “Fast simulated annealing,” *Physics letters A*, vol. 122, no. 3, pp. 157–162, 1987.

[54] D. S. Baum, J. Hansen, and J. Salo, “An interim channel model for beyond-3G systems: extending the 3GPP spatial channel model (SCM),” in *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st.*, vol. 5, May 2005, pp. 3132–3136.

[55] J. Unnikrishnan and V. V. Veeravalli, “Cooperative sensing for primary detection in cognitive radio,” *Selected Topics in Signal Processing, IEEE Journal of*, vol. 2, no. 1, pp. 18–27, 2008.

[56] G. Fertin, E. Godard, and A. Raspaud, “Acyclic and k-distance coloring of the grid,” *Information Processing Letters*, vol. 87, no. 1, pp. 51–58, 2003.

[57] J. Shen, T. Jiang, S. Liu, and Z. Zhang, “Maximum channel throughput via cooperative spectrum sensing in cognitive radio networks,” *Wireless Communications, IEEE Transactions on*, vol. 8, no. 10, pp. 5166–5175, 2009.

[58] W. Kuo and M. J. Zuo, *Optimal reliability modeling: principles and applications.* John Wiley & Sons, 2003.