Scheduling in Multi-hop Multi-band Cognitive Radio Networks Utilizing Potential Fields

Henri Hentilä

School of Science

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo August 7, 2017

Thesis supervisor:

Prof. Visa Koivunen

Thesis advisor:

D.Sc. (Tech.) Jan Oksanen

Aalto University
School of Science

AALTO UNIVERSITY
SCHOOL OF SCIENCE

ABSTRACT OF THE
MASTER'S THESIS

Author: Henri Hentilä

Title: Scheduling in Multi-hop Multi-band Cognitive Radio Networks Utilizing Potential Fields

Date: August 7, 2017 Language: English Number of pages: 8+67

Signal Processing and Acoustics

Professorship: Signal Processing

Supervisor: Prof. Visa Koivunen

Advisor: D.Sc. (Tech.) Jan Oksanen

The problem of scheduling transmissions in multi-hop multi-band cognitive radio (CR) networks is studied. The scheduling is subdivided into long- and short-term problems. The long-term problem concerns the routing of data packets. It is solved by means of a novel potential field based modeling technique that abstracts routes away from paths defined by sequences of specific network links and onto an underlying potential field. The short-term problem is formulated as a mixed integer nonlinear program (MINLP) that describes the instantaneous interference constraints that secondary users (SUs) have to satisfy if they wish to utilize the spectrum licensed to primary users (PUs). These interference constraints are defined in terms of required signal-to-interference-plus-noise ratios (SINRs) at receivers. A centralized greedy algorithm, as well as a distributed version of it, is proposed for solving the short-term problem. Simulations show that the proposed methods work well in both static and dynamic networks. Moreover, the methods contain adjustable parameters that allow one to e.g. improve the fairness of the scheduling at the cost of overall throughput, or to significantly improve the overall throughput at the cost of a non-zero but arbitrarily small probability of violating the interference constraints.

Keywords: cognitive radio, scheduling, routing, potential field

AALTO-UNIVERSITETET
HÖGSKOLAN FÖR TEKNIKVETENSKAPER

SAMMANDRAG AV
DIPLOMARBETET

Författare: Henri Hentilä

Titel: Skedulering i kognitiva månghopps- och mångbandsradionätverk utnyttjandes potentialfält

Datum: August 7, 2017 Språk: Engelska Sidantal: 8+67

Signalbehandling och akustik

Professur: Signalbehandling

Övervakare: Prof. Visa Koivunen

Handledare: TkD Jan Oksanen

Skedulering av sändningar i kognitiva månghopps- och mångbandsradionätverk undersöks. Skeduleringen delas upp i ett långsiktigt respektive ett kortsiktigt problem. Det långsiktiga problemet behandlar dirigeringen av datapaket. Problemet löses genom en ny potentialfältsbaserad modelleringsteknik där datarutterna beskrivs av ett underliggande potentialfält istället för sekvenser av specifika nätverkslänkar. Det kortsiktiga problemet utgörs av ett icke-linjärt optimeringsproblem som beskriver de interferensvillkor vilka de sekundära användarna måste uppfylla för att få tillgång till det spektrum som licensierats till primära användare. Interferensvillkoren är definierade i form av nödvändiga signal-till-interferens-plus-brusförhållanden hos mottagare. En centraliserad girig algoritm, samt en distribuerad version av den, föreslås som lösning på det kortsiktiga problemet. Simulationer visar att de föreslagna metoderna fungerar väl i både statiska och dynamiska nätverk. Dessutom innehåller metoderna justerbara parametrar med vilka man t.ex. kan förbättra rättvisheten i skeduleringen på bekostnad av den totala datahastigheten, eller märkbart förbättra den totala datahastigheten på bekostnad av en positiv men godtyckligt liten sannolikhet att interferensvillkoren inte alltid uppfylls.

Nyckelord: kognitiv radio, skedulering, routing, potentialfält

Contents

Symbols and Abbreviations

Symbols

i, j, k, l	indices commonly used in the subscript of symbols to refer to SUs
N	the total number of SUs in the network
m	index commonly used in the superscript of symbols to refer to a frequency band, which has a corresponding PU transmitter `TX` and PU receiver `RX` associated with it
M	the total number of frequency bands in the network
`TX`	special symbol that denotes a transmitter; when used in conjunction with m refers to the PU transmitter in band m
`RX`	special symbol that denotes a receiver; when used in conjunction with m refers to the PU receiver in band m
$\mathcal{N}$	the set of SUs in the network
$\mathcal{M}$	the set of licensed frequency bands
$\overline{\mathcal{N}}$	the set of candidate transmitters
$\underline{\mathcal{N}}$	the set of candidate receivers
$\mathcal{N}_i$	the set of transmission neighbors of SU i
$\mathcal{N}_i^m$	the set of transmission neighbors of SU i in band m
$\overline{\mathcal{N}}_i^m$	the set of candidate neighbors of SU i in band m
$\mathcal{N}_i^*$	the set of control neighbors of SU i
$\mathcal{N}_i^*(s)$	the set of s-hop control neighbors of SU i
$\mathcal{C}$	the set of candidate transmissions
$\mathcal{C}_i^m$	the set of candidate transmissions in band m with transmitter i
$\mathcal{S}$	the transmission set consisting of all SUs that have been scheduled to either transmit or receive
(i, j, m)	a link in the network, identified by its transmitter $i \in \mathcal{N}$, receiver $j \in \mathcal{N}$, and frequency band $m \in \mathcal{M}$
(i, j)	a link (i, j, m) for any band $m \in \mathcal{M}$
γ_{ij}^m	SINR of link (i, j, m)
$\gamma_{\mathtt{TX},\mathtt{RX}}^m$	SINR of the link between the PU transmitter and receiver in band m
S_{ij}^m	received power of signal sent from SU i to SU j in band m
I_j^m	received interference power at SU j in band m
N_j^m	received noise power at SU j in band m
IN_j^m	received interference-plus-noise power at SU j in band m
a_{ij}^m	scheduling decision for link (i, j, m)
b^m	state of the PU transmitter in band m
P_i^m	transmit power of SU i in band m
$P_{\mathtt{TX}}^m$	transmit power of the PU transmitter in band m
$\mathbf{x}_i$	coordinates of SU i
$\mathbf{x}_{\mathtt{TX}}^m$	coordinates of the PU transmitter in band m
$\mathbf{x}_{\mathtt{RX}}^m$	coordinates of the PU receiver in band m

d_{ij}	Euclidean distance between SUs i and j
$d^m_{\mathtt{TX},j}$	Euclidean distance between the PU transmitter in band m and SU j
$d^m_{i,\mathtt{RX}}$	Euclidean distance between SU i and the PU receiver in band m
$d^m_{\mathtt{TX},\mathtt{RX}}$	Euclidean distance between the PU transmitter and receiver in band m
g_{ij}	path loss between SUs i and j
$g^m_{\mathtt{TX},j}$	path loss between the PU transmitter in band m and SU j
$g^m_{i,\mathtt{RX}}$	path loss between SU i and the PU receiver in band m
$g^m_{\mathtt{TX},\mathtt{RX}}$	path loss between the PU transmitter and receiver in band m
σ_{SU}	noise power at SUs
σ_{PU}	noise power at PUs
α	environment-dependent path loss exponent
Γ	SINR threshold
z^m	probability that the PU transmitter in band m is active
Z	probability that all PUs are active
q_i	packet queue at SU i
$q_i[n]$	the n-th packet in queue q_i
$\lvert q_i \rvert$	the length of queue q_i
$\mathbf{src}(p)$	the source SU of packet p
$\mathbf{dst}(p)$	the destination SU of packet p
$\mathbf{dly}(p)$	the delay experienced by packet p so far
Φ_{ij}	potential at SU i w.r.t. destination j
$\underline{\Phi}_{ij}$	attractive (VFF) potential of destination j in the VFF
$\overline{\Phi}^m_{ij}$	repulsive (VFF) potential of the PU transmitter in band m
ν	parameter used to control the contribution of the attractive potential in the VFF
ξ	parameter used to control the contribution of the repulsive potential in the VFF
C^0_{ij}	the cost/delay (1 or ∞) of transmitting over link (i,j) in idle spectrum
$C^{1,m}_{ij}$	the cost/delay (1 or ∞) of transmitting over link (i,j,m) in occupied spectrum
D^0_{ij}	the expected delay between SUs i and j in idle spectrum
D^1_{ij}	the expected delay between SUs i and j in occupied spectrum
Q_{ij}	priority of link (i,j)
λ	parameter used to control the contribution of the priority in the overall link weight
$\mathtt{w}(\cdot,\cdot,\cdot)$	weight function $\mathtt{w} : \mathcal{N} \times \mathcal{N} \times \mathcal{M} \mapsto \mathbb{R}$
w^m_{ij}	weight of link (i,j,m)
$(\mathbf{a},\mathbf{P})$	solution to the short-term scheduling problem
$(\mathbf{a}_{\mathcal{S}},\mathbf{P}_{\mathcal{S}})$	partial solution to the short-term scheduling problem w.r.t. transmission set $\mathcal{S}$
$(\mathbf{a}_i,\mathbf{P}_i)$	local solution to the short-term scheduling problem at SU i
$\tilde{a}^m_{ij}$	help variable that indicates whether transmission over link (i,j,m) is successful or not

$P_{i,\mathrm{U}}^m$	upper bound of P_i^m
$P_{i,\mathrm{L}}^m$	lower bound of P_i^m
ϵ	parameter used when computing the transmit power upper bound $P_{i,\mathrm{U}}^m$
τ	parameter used to adjust the transmit power P_i^m between its upper and lower bounds
$\langle x \rangle_i$	worst case estimate of parameter x from the point of view of SU i

Abbreviations

DSA	dynamic spectrum access
CR	cognitive radio
PU	primary user
SU	secondary user
SINR	signal-to-interference-plus-noise ratio
CCI	co-channel interference
ACI	adjacent channel interference
MINLP	mixed integer nonlinear program
CCC	common control channel
GPF	greedy potential field
VFF	virtual force field
SSDOPF	single-session delay-optimal potential field

1 Introduction

Useful radio spectrum is a scarce resource that is subject to ever increasing data rate demands by emerging wireless systems. There are two main strategies for addressing this increasing demand. One is to exploit spectrum not already in use. This is the case with e.g. millimeter wave (mmWave) operating in the 110 to 300 GHz band, which is planned to be used in upcoming 5G mobile networks [1]. The other strategy, and the one of interest in this thesis, is to utilize the spectrum already in use more efficiently and in a flexible manner. As it turns out, the current static allocation of spectrum has lead to very inefficient usage of the available spectrum [2].

Cognitive radio [3] allows users to acquire knowledge of the spectrum state in their vicinity via spectrum sensing, and to adjust their transmission parameters in order to exploit this knowledge. This makes it possible to opportunistically access spectrum that has been statically licensed for exclusive use, allowing new unlicensed devices to seamlessly operate in already existing infrastructure. The *underutilized spectrum*, i.e. the licensed spectrum that can be exploited for unlicensed use, can be found either by sensing the activity of licensed users, or by adjusting one's transmission parameters, or, preferably, a combination of both.

From the point of view of a single user, the main problem in cognitive radio is to identify the underutilized spectrum. This could involve assessing when a licensed user is inactive, or determining a transmit power level that allows the unlicensed user to transmit when the licensed user is active. However, from the point of view of an entire *network* of cognitive users, the problem is equally about *how to exploit* the underutilized spectrum. Namely, the underutilized spectrum is a shared resource among the cognitive users, and hence not every user may be able to exploit the underutilized spectrum that they have identified. To ensure that the underutilized spectrum is exploited to its fullest, a *scheduling* problem needs to be solved.

Unlike traditional wireless networks, scheduling in a cognitive radio network is not as simple as finding a predefined schedule for the users to adhere to. This is due to the fact that the underutilized spectrum is a time-varying resource (in addition to being location- and frequency-varying) for which the future may be unknown. In particular, the licensed users are active at different times and hence the underutilized spectrum identified at a certain time instance may not be the same as the underutilized spectrum available at another time. For this reason, it is crucial that underutilized spectrum is exploited in a timely manner after being identified, and that the scheduling is not reliant on any particular portion of the spectrum being available (underutilized) at a given time. Consequently, we will prefer solutions that are of low enough complexity, and that can adapt to dynamic radio environments.

This thesis will examine the scheduling problem in cognitive radio networks, with particular emphasis on the problems discussed above. The main contributions are the following:

1. The scheduling problem in cognitive radio networks is formulated by subdividing it into long- and short-term problems. In the long-term problem, the focus is on routing data packets, whereas the short-term problem describes the issues instrinsic to the time-varying spectrum of cognitive radio networks.

2. A potential field modeling technique for routing packets in multi-hop multi-band cognitive radio networks is developed. This technique was originally introduced in [4], and later extended from single-band to multi-band networks in [5].

3. A centralized polynomial-time greedy algorithm that approximately solves the short-term scheduling problem is proposed. The algorithm was originally published in [5], and has received some slight modifications to be applicable to the problem as defined in this thesis.

4. A distributed version of the centralized greedy algorithm is developed. The technique presented is a generalized version of the one originally published in [6].

This thesis work has led to publications [5] and [6]. In [5], contributions 2 and 3, i.e. the potential field modeling technique for packet routing and the centralized greedy scheduling algorithms were proposed. In [6], the centralized scheduling method was modified into a corresponding distributed method, leading to contribution 4 above.

The rest of the thesis is organized as follows. In chapter 2, the necessary background material is discussed. This includes an introduction to the studied problem on a general level, a breakdown of the different components that define the problem, as well as a brief literature review. In chapter 3, the problem to be studied in this thesis is explained in detail. Here, the interference model to be used is defined, and the scheduling is broken down into long- and short-term problems. In chapters 4 and 5, the centralized and distributed solutions to the problem at hand are presented. A key component of both solutions is the potential field modeling that allows us to weight transmission links in the scheduling. The results achieved by said solutions are then demonstrated via extensive computer simulations in chapter 6. Finally, the thesis is concluded with a summary in chapter 7.

2 Background

In this chapter, we will look at existing work on the subject of scheduling in cognitive radio networks. We will start by defining what scheduling in cognitive radio networks is, by first introducing the concept of *cognitive radio* in section 2.1, and then discussing the problem of *scheduling* in such networks in section 2.2. The scheduling problem will then further be broken down into a number of different components in section 2.3. Finally, we conduct a review of existing solutions in section 2.4, and conclude the chapter by pointing out where this thesis fits into the big picture in section 2.5.

2.1 Cognitive Radio

The current static allocation of radio resources is unable to properly utilize the available spectrum [2]. In particular, the licensees that have exclusive rights to the spectrum are inactive for considerable portions of time, leaving *spectrum white spaces* during which the spectrum is idle and, yet, cannot be used by others. The need for *dynamic spectrum access* (DSA), where the spectrum resources can be dynamically reallocated, is thus warranted.

Different DSA techniques can be broadly categorized into the following three models [7]:

1. dynamic exclusive use model
2. open sharing model
3. hierarchical access model

The *dynamic exclusive use model* is similar to static allocation, but allows spectrum resource rights to be reallocated either through trading/selling or in a more open manner based on network statistics and user needs. However, unlike the *open sharing model* and *hierarchical access model*, the allocations are made on a large time-scale and thus suffer from the same spectrum white spaces as the static allocation. In the open sharing model, also known as the *commons model*, users share their resources as peers allowing the spectrum to be used by each user on a per-need basis. The hierarchical model is similar, but splits users by priority into two groups: the licensed *primary users* (PUs), and the *secondary users* (SUs) who are only allowed to use the spectrum not already in use by PUs.

Our interest in this thesis is the hierarchical access model. Specifically, we study *cognitive radio* (CR) as an implementation of this model. The term cognitive radio was introduced by Joseph Mitola III [3] to refer to a type of software-defined radio that can intelligently adapt its system parameters to cope with changing spectrum conditions. Letting the SUs be users equipped with CRs, and PUs the already existing licensed users under the old static allocation policy, CR allows us to design SU networks that can coexist with legacy systems already in place. In particular, CR allows the SUs to sense and identify the idle spectrum not used by PUs (*spectrum opportunity identification*), which they can then utilize for transmission (*spectrum opportunity exploitation*) in a manner that is transparent to the PUs (as defined by

the *regulatory policy*). Although an important component of CR, this thesis will not consider the problems relating to spectrum opportunity identification. Instead, our focus will be on the spectrum opportunity exploitation under a given regulatory policy.

More importantly, we will consider this problem from the point of view of an entire *CR network*. Here we use the term CR network to refer to a wireless network operating under the hierarchical access model, where the SUs are equipped with CR. Our aim is thus not only to design policies that allow SUs to efficiently utilize the spectrum white spaces left by the PUs, but also to ensure that the SUs coordinate their decisions with each other. We refer to this problem as *scheduling* in the CR network.

2.2 The Scheduling Problem

The problem we are studying is the following. Given a CR network consisting of both licensed PUs and unlicensed SUs, we want to schedule the transmissions of the SUs with the following two goals:

G1 Provide guarantees that PUs are not disturbed by the SUs' transmissions.

G2 Optimize some objective that characterizes how good the scheduling is from the SUs' point of view.

The wording above naturally lends itself to some interpretation, and in this chapter, the idea is to give an overview of different interpretations and solutions found in the field. Before that, however, a more detailed breakdown of the scheduling problem will be provided.

We will start by considering the problem in the context of general wireless networks. Fundamentally, scheduling in wireless networks is the act of coordinating concurrent transmissions, i.e. deciding who should be allowed to transmit using which resources. As mentioned in the introduction, the need for scheduling arises due to the communication channel - in this case the wireless spectrum - being shared by the network nodes. If the channel is not shared, there would be no need for scheduling as all nodes could transmit/receive data at any time without interfering with each other's transmissions.

Scheduling in wireless networks can be separated into two different types [8]: scheduling on the protocol level, and scheduling as a resource allocation problem. The former considers the individual interactions between network nodes, in particular the interactions between neighboring nodes who can hear each other. The latter, meanwhile, performs the scheduling on the network as a whole by allocating available resources, i.e. the available spectrum, among all nodes requesting it.

The biggest difference between the two approaches is the way they model the data traffic. When scheduling on the protocol level, the data traffic is modeled as individual packets with unknown and often bursty arrival rates. Thus, the goal could be to minimize the downtime between transmissions while avoiding conflicts when two neighboring nodes both have packets to transmit. When scheduling in the form

of resource allocation, the data traffic is instead modeled by given rate constraints, and the goal is to find the links in the network that can be active concurrently such that these rate constraints are met. Alternatively, no rate constraints are given and the goal could be to simply maximize the sum rate for the network in either an opportunistic or fair manner. Another way to interpret this, is that protocol level scheduling is performed on a small time-scale taking each individual instance of time into account, while resource allocation is performed on a large time-scale with less concern for momentary fluctuations in the system state.

Since the two approaches deal with different problems, the more interesting scheduling problems arise when both are considered simultaneously. That is, when both short- and long-term goals are taken into account. The objective in goal **G2** could be to maximize the sum data rate of the network, subject to the constraint that individual packet queues with bursty arrival rates do not grow uncontrollably. The key difference between traditional resource allocation and this type of hybrid scheduling is the way the demands or constraints on the scheduling vary over time. The varying nature of the constraints is one of the defining features of CR networks when compared to traditional wireless networks, making the hybrid scheduling approach of more interest for our purposes.

Before moving on to CR networks and goal **G1**, we will briefly return to our initial problem formulation for scheduling: deciding who should be allowed to transmit using which resources. As has already been established, the assumption is that not everyone can transmit simultaneously using the same resources or the task would be trivial. In the case of wireless networks this assumption is a natural consequence of the transportation medium being considered, namely the wireless spectrum where any two concurrent transmissions interfere with each other to some degree. The problem then, is to decide when two or more transmissions interfere too much, i.e. deciding which subsets of transmissions can occur concurrently. This is by no means a simple task, as we will see later.

In CR networks, there is not only interference between concurrent SU transmissions, but also interference from PUs to SUs, and from SUs to PUs. In particular, it is the interference from SUs to PUs that is of interest in goal **G1**. There are two main strategies for controlling this interference (both of which will be explained in more detail later): forbidding SUs that would cause non-negligible interference at PU receivers from transmitting concurrently with the PUs, or lowering the SUs' transmit powers such that the interference caused by their transmissions is negligible at the PU receivers.

Note that the problem described above is very similar to that of finding sets of nodes that can transmit concurrently. The only notable difference is that in this case the PUs have priority. In that sense, the strategies mentioned above are applicable in both cases and scheduling in CR networks can be seen as a special case of scheduling in wireless networks where a subset of the transmitters is already given, namely the set of transmitting PUs. However, this is only true when considering a snapshot of the system state. In reality, the set of transmitting PUs is not constant, but varies over time, often sporadically. It is in fact the sporadic activity of the PUs that CR is designed to exploit.

In particular, since the SUs want to utilize the unused portions of the licensed spectrum, and these change as a function of time, it is crucial that SUs can adapt their transmission parameters on the fly. An important part of this is being able to accurately sense the spectrum to identify idle or underutilized parts of it. However, spectrum sensing is outside the scope of this work, and we will not discuss this aspect further. See [9], [10] for references on spectrum sensing. Our focus will instead be on exploiting the changing state of the spectrum to the fullest, which is best done via the hybrid scheduling that considers both networkwide spectrum allocation and individual interactions between neighboring nodes.

In the following subsections we will take a closer look at the problems involved in designing a scheduling algorithm for CR networks, and then present existing solutions, both new and old.

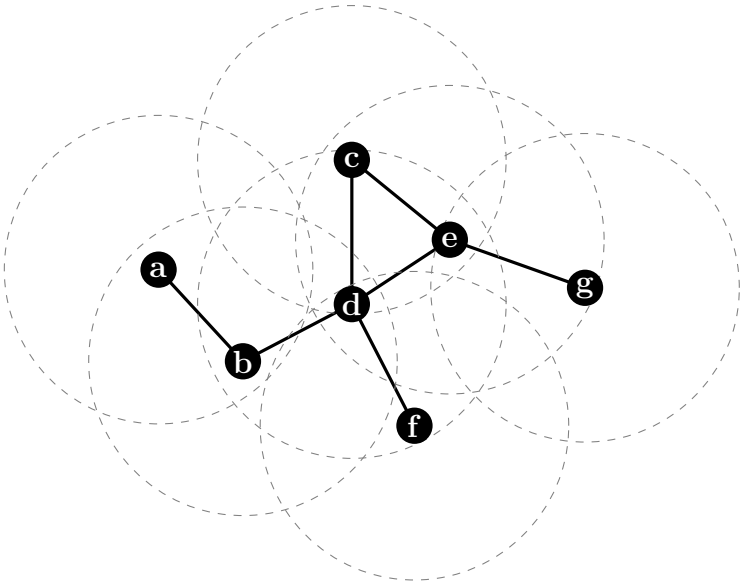
2.3 A Taxonomy of the Scheduling Problem

When designing a scheduling algorithm, the system for which the scheduling is to be done first has to be modeled. Designing this model requires a tradeoff between taking into account the intended real-world operational environment while keeping complexity down so as to make both designing and analyzing algorithms for it more tractable. Below are listed some of the main decisions that need to be made when designing the system model.

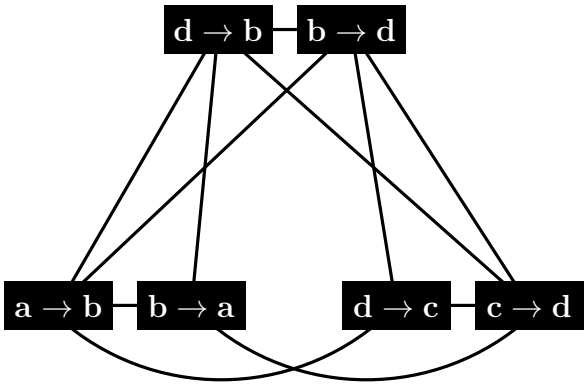
2.3.1 Interference Model

Among the most important decisions when modeling a wireless network is how to model the interference. Accurately modeling the interference at every point in space is often not feasible because of the variety of propagation environments, spectrum usage patterns, employed modulation and access techniques, shadowing and multipath effects as well as data transmission needs of the users in the vicinity, just to name a few factors. Therefore, one must resort to approximate models.

One of the more popular ways to model the interference is in the form of a *connection graph* $G_C = (V, E)$ where the vertices V represent network nodes, and the edges $E \subseteq V \times V$ the links between them. An example of this can be seen in Fig. 1a. If $\{u, v\} \in E$, we say that u and v are 1-hop neighbors and any transmission by u is heard by v, vice versa. Transmissions from a node's k-hop neighbors (where k denotes the distance between the nodes in the graph) for $k > 1$ are not heard by the node. In the network shown in Fig. 1a, we see that e.g. nodes b and e cannot hear each other's transmissions, as their distance in the graph is 2. Under this model, interference is simply seen as an unwanted transmission. That is, any transmissions from u's 1-hop neighbors that u is not the intended recipient of is seen as interference at node u. Since transmissions are only heard by 1-hop neighbors, this means that k-hop neighbors for $k > 1$ do not interfere with each other. Thus, any pair of nodes that are 3 or more hops away from each other could safely transmit concurrently, as the receivers of their transmissions would be at least 2 hops away from the other (unwanted) transmitter and hence outside the interference range of it. In Fig. 1a



(a) connection graph



(b) interference graph

Figure 1: Different implementations of the protocol model. The edges in the connection graph in (a) are based on a given transmit/interference distance shown as a dashed circle around each node. The interference graph in (b) is constructed from the connection graph in (a) restricted to nodes a, b, c and d. For instance, we can see from (b) that the (directed) links (a, b) and (d, c) cannot be active simultaneously. Consulting (a), we conclude that this is due to the fact that d is a 1-hop neighbor of b, and thus d cannot transmit if b is trying to listen to a.

we see that this is the case with e.g. nodes a and c, and they can therefore safely transmit at the same time. The connection graph model is popular not only for its simplicity, but also due to the fact that graphs are well studied and have numerous analytical results and algorithms available in the literature. Examples of this type of

interference model include [11], [12].

Another possibility, also utilizing graph theory, is to model the interference via an *interference graph* [1] $G_I = (V_I, E_I)$. Here, the vertices V_I represent transmission links, i.e. transmitter-receiver pairs, and two vertices $u, v \in V_I$ have en edge between them if and only if the links they represent cannot be active simultaneously due to interference. An example of this is shown in Fig. 1b. The advantage of this type of graph compared to the connection graph is that receivers and transmitters do not have to be considered separately, making it much simpler to decide when two links can be active at the same time. In particular, two links $u, v \in V_I$ can be active at the same time if and only if $\{u, v\} \notin E_I$. Thus, finding a set of transmission links that can be active simultaneously is equivalent to finding an independent set $V_I' \subseteq V_I$ in G_I. For example, in Fig. 1b the links (a, b) and (c, d) form an independent set, and can therefore be active at the same time. Since the goal is often to utilize the available spectrum to its fullest, finding the maximum independent set in G_I is of particular interest. Unfortunately, finding maximum independent sets in arbitrary graphs is a known **NP**-complete problem, and thus most algorithms settle for a maximal independent set. See [13], [14], [15] for examples of this type of interference model.

In what is referred to as the *protocol model* in [16], interference is a binary relation that includes any pair of nodes that are within a predefined *interference distance* of each other. Although the relation is based on the distances between the nodes in [16], we note that this is simply a special case of the above mentioned graph theoretic approaches where the existence of edges between nodes would be based on the distances between them. Henceforth, we will use the term *protocol model* to refer to any type of interference model where the interference is a binary relation between pairs of nodes (connection graph) or between pairs of transmitter-receiver links (interference graph). An example of a connection graph and (part of) the corresponding interference graph is shown in Fig. 1.

There are two drawbacks with the protocol model. Firstly, interference is not a binary quantity, but continuous and inversely proportional to the distance it travels among other things. Secondly, it fails to take into account the cumulative nature of interference. In combination with the first point, this means that even though two unwanted transmitters may not individually add enough interference to disturb a transmission, their cumulative interference might.

To account for these shortcomings, [16] presents an alternative model, referred to as the *physical model*. Here, the interference is no longer modeled in isolation between pairs of nodes, but instead each unwanted transmission i adds its own interference I_i to a cumulative total interference $I = \sum_i I_i$. The quality of links are then determined

[1] The term interference graph is sometimes used to refer to a similar concept as the connection graph that is exclusively made to model the interference. That is, interference and transmission ranges are not assumed to be equal and thus connection and interference links constitute their own separate graphs. However, since we can apply the exact same reasoning and algorithms to this type of interference graph as we could to the connection graph, it does not require any further analysis and we will not make any distinction between the two.

using the signal-to-interference-plus-noise ratio (SINR)

$$\text{SINR} = \frac{S}{I + N} \tag{1}$$

where S is the power of the signal the receiver is listening to and N a noise term that represents the background noise. It is easy to see that the more interference, the lower the SINR will be. The sets of transmissions that can occur concurrently can then be decided as follows: given an SINR threshold Γ, a set of transmissions can occur concurrently if the SINR of each transmission is above the threshold. Note that the interference power I at each receiver will depend on the entire set of chosen transmitters, and thus we can no longer model the links as edges in a graph as in the protocol model. Examples of these types of physical models can be found in e.g. [17], [18], [19].

2.3.2 Spectrum Bands

So far we have referred to the wireless spectrum as a shared channel or resource between the network nodes. This is to emphasize that a transmission in the channel will affect all other transmissions taking place concurrently. However, transmissions that use different enough frequencies will only add a small or even negligible amount of interference at each other's receivers. In such a case we say that the transmissions use orthogonal frequency bands. In particular, we note that the full frequency range available can be split into disjoint sets of orthogonal frequency bands that can be used more or less concurrently without worrying about interference between the bands. This is the case in OFDM modulation used in LTE systems and WiFi, for example. Usually, there is still some overlap and leakage between the bands, and if we want an accurate model of the interference we should consider not only *co-channel interference* (CCI), i.e. interference from transmissions in the same band, but also *adjacent channel interference* (ACI). In practice, however, most works ignore the ACI as its contribution to the overall interference is small. For a more in-depth explanation of the ACI and its role in the overall interference, see e.g. [20].

If we assume only the existence of CCI, each frequency band can be considered its own set of resources that does not affect other bands. However, letting k be the number of channels available, this is not the same as performing k separate scheduling tasks. This is due to the fact that, although the resources of each band can be treated separately, the nodes for which the scheduling is performed are shared. Thus, scheduling in a multi-band environment introduces new challenges not present in the single-band environment. In particular, one must decide whether a transmission that cannot occur concurrently with other transmissions should be moved to a different time slot or a different frequency band. Fig. 2 demonstrates this problem.

2.3.3 Network Connectivity

Another aspect that has a significant impact on what type of scheduling will be needed is the number of hops between the network nodes. In a single-, or 1-hop network, all nodes can communicate directly with each other, while in a multi-hop network the

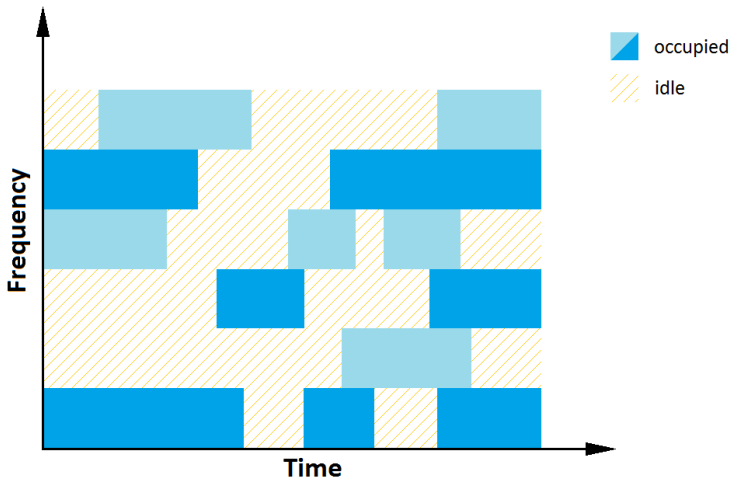


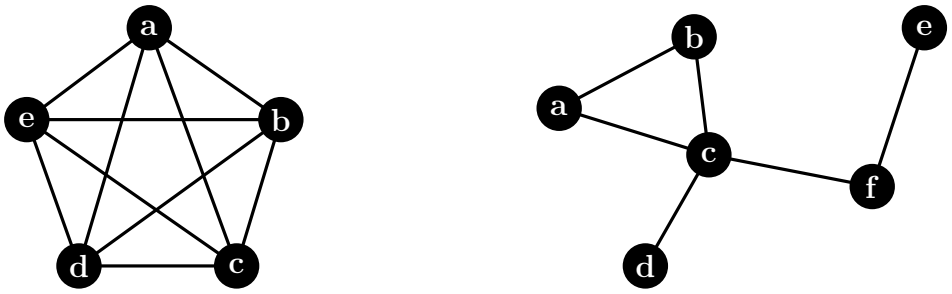
Figure 2: In a multi-band spectrum environment, spectrum opportunities vary in both time and frequency domains.

communication between two nodes may require intermediate nodes acting as relays. This is most easily demonstrated via the connection graphs introduced earlier, as in Fig. 3. In a 1-hop network, the connection graph $G_C = (V, E)$ is fully connected, i.e. $E = W$ where W is the set $(V \times V) \backslash \{(v, v) \in V \times V\}$. In a multi-hop network on the other hand, the edge set E is a proper subset of W.

Often, scheduling in a multi-hop network is not very different from scheduling in a 1-hop network, as long as the data links, i.e. the transmitter and receiver of each data transmission, are already given to us [12], [11], [14]. Therefore, we will make a further distinction between multi-hop networks where the data links are given, and multi-hop networks where instead of data links, we are given source-destination pairs for a set of data streams. Since the given source-destination pairs are generally not 1-hop neighbors, the latter case introduces the problem of finding routes for the data in addition to scheduling the transmissions. Examples of such network models can be found in [13], [21], [22].

2.3.4 Scheduling Mode

Having decided on a network model, the next problem is designing the actual scheduling algorithm. For this, one of the more important decisions is *how* and/or *where* the scheduling is to be performed. Namely, one should decide if the scheduling is to be performed in a *centralized* manner by a single central entity [13], [23], [21], or *distributively* by each individual network node [14], [22], [12]. The key problem here is that the network parameters needed to make the scheduling decisions are initially spread out among the nodes of the network. Thus, we must decide whether it is worth the communication overhead of transferring this information to a central scheduling entity, or if we are better off letting individual nodes make scheduling decisions based on incomplete, local information of the network.



(a) 1-hop network (b) multi-hop network

Figure 3: Different kinds of networks as depicted via their connection graphs. In the 1-hop network in (a), all nodes have a direct link to every other node. By comparison, the nodes in (b) are only directly connected to a subset of the other nodes. For instance, if d wants to communicate with e, it must do so via c and f.

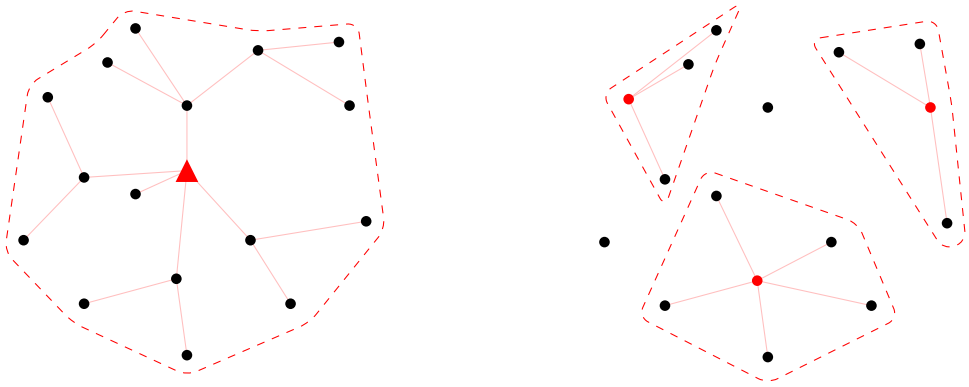
Clearly, more information can only improve the scheduling, and therefore the centralized approach will always make better or at least as good decisions as the distributed one. In particular, the distributed approach may lead to conflicts in the scheduling decisions, which is completely avoided in centralized scheduling. On the other hand, getting the information to the central decisionmaker may prove challenging and lead to delays and overhead in exchanging information among the nodes, especially in a multi-hop network as shown in Fig. 4. The decision then often comes down to what type of time-scale the scheduling decisions are made for. For long-term decisions it may be better to perform the more costly centralized scheduling, while short-term decisions may require a distributed approach.

2.3.5 Spectrum Utilization Technique

When dealing with CR networks specifically, we need to consider how SUs should utilize the leftover licensed spectrum. There are two main approaches to this, shown in Fig. 5, often referred to as *spectrum overlay* and *spectrum underlay*, respectively.

In spectrum overlay, SUs are free to utilize any frequency bands that are sensed to be idle. Because the state of each frequency band will shift between idle and occupied based on the current PU activity, this means that SUs must be able to quickly change channels at any time. In some studies, this technique is called *spectrum interweave*, and spectrum overlay instead refers to a collaborative approach where SUs help relay PU transmissions they interfere with [24]. Scheduling algorithms based on the spectrum overlay approach can be found in e.g. [25], [14], [26].

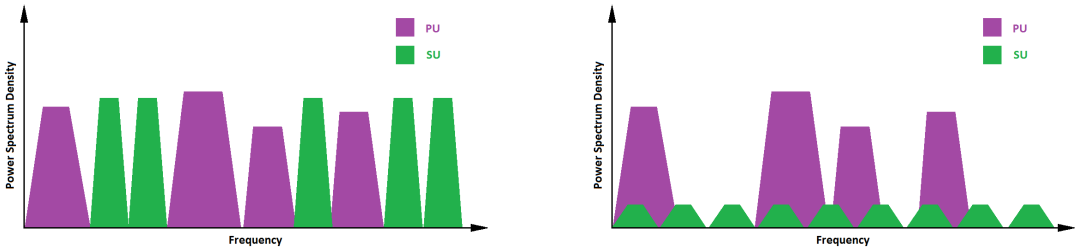
In spectrum underlay, meanwhile, SUs may use any licensed channel at any time. However, in order to guarantee that PUs do not experience too much interference, they must decrease their transmit powers. This allows SUs to transmit in licensed bands without having to worry about a changing spectrum availability, but at the cost of reduced transmission rates compared to spectrum overlay. While this kind of pure spectrum underlay makes sense if we cannot accurately sense the spectrum state or doing so is too costly, most scheduling algorithms that use spectrum underlay



(a) centralized scheduling

(b) distributed scheduling

Figure 4: Centralized (a) versus distributed (b) scheduling. The decisionmakers are marked in red, and the portion of the network which they have received information about is indicated by the corresponding dashed red line. The red lines between nodes show the paths in the underlying connection graph used to transfer this information. Although the centralized decisionmaker in (a) has perfect global knowledge of the network, this information must be collected from up to 3 hops away. Compare this to (b), where the decisionmakers only have incomplete local information available, but can receive this information directly from their 1-hop neighbors.



(a) spectrum overlay

(b) spectrum underlay

Figure 5: Different techniques for utilizing the leftover licensed spectrum. In spectrum overlay, SUs may transmit at full power in any frequency band sensed to be idle. In spectrum underlay, SUs can use any frequency band for their transmissions, but must lower their transmit power.

combine it with spectrum overlay. That is, letting SUs transmit with full power in any idle bands, and with the adjusted lower power in occupied bands. Examples of this include [27], [28], [19].

2.3.6 Objective

Finally, in order to measure how good a scheduling algorithm is, we need some objective to measure it against. In other words, we need to define what the objective function we wish to optimize in goal **G2** is.

This objective is often to maximize some variation of the sum data rate achieved by the scheduled transmissions [15], [29], [17]. Naturally, if we only consider the sum data rate of all links, this could lead to a very uneven distribution of data rates among the transmitters, with some transmitters getting significantly higher rates than others. Thus, some works try to achieve fairness in the data rates, e.g. through proportional fairness [27] or by giving the links different weights [30].

Sometimes we are not interested in maximizing the rates, but simply meeting certain rate requirements [13], [21], [12]. On top of meeting these requirements, there is also often a minimization task involved in meeting them. In [13], the goal is to minimize the number of time slots needed to achieve the required rates, while [21] aims to use as little spectrum as possible for the task.

A performance criterion that is closely related to the data rate or throughput, and is of particular interest when studying video and speech signals, is latency. This is more easily understood when modeling the data traffic as packets. In this case the latency is defined as the delay experienced by packets as they travel from source to destination. Maximizing data rates then roughly corresponds to minimizing the delays of packets. In [5], the average delay of packets is used to measure the throughput achieved by the scheduling, while the maximum delay experienced by any packet is used to measure fairness. Meanwhile, [23] considers both maximization of data rates and minimization of delays.

Models with data in the form of packets often include queues for storing the packets. This introduces yet another type of criterion: the queue sizes. Of particular interest is keeping the queue sizes below a threshold [31], [22]. The are two main reasons for this. Firstly, shorter queue sizes means packets spend less time waiting in queues and thus their delays will be shorter. Secondly, if the queue sizes grow uncontrollably, we eventually end up with a congestion in the network data traffic. Not to mention that the storage capacity, and thus maximum queue size, is in practice always limited.

In some networks, e.g. sensor networks, we may not be interested in throughput, but instead only be concerned with the energy efficiency of the network [18]. In such networks, the scheduling tends to favor low transmission powers and as a consequence shorter transmit distances.

2.4 Brief Review of Scheduling Algorithms

In the following, a brief review of a number of proposed scheduling algorithms will be provided. Note that the first two examples, [12] and [15], are designed for general wireless networks, and as such do not consider goal **G1**.

In [12], the network is modeled via the connection graph introduced earlier. There is no separate modeling of the data traffic, but instead, the scheduling aims to find a TDMA schedule for the network nodes such that the spectrum is utilized to its fullest. The links are assumed to all have the same capacity, and thus the objective reduces to finding the largest possible subset of network nodes that can transmit concurrently in each time slot. This problem is shown to be **NP**-complete. The scheduling is performed in a distributed manner by having 1-hop neighbors exchange

information about themselves and their other neighbors. In particular, it is shown that under the interference model used, nodes can make non-conflicting decisions by having information from at most 4 hops away. Thus, the scheduling requires at most 4 iterations of information exchange between neighbors.

The method used in [15] also utilizes a connection graph to model the network. Unlike [12], the scheduling is not only concerned with assigning transmitters to different time slots, but also the routing of data. These two problems are together formulated as a mixed integer linear program (MILP). In addition to an objective function that aims to maximize a utility based on data rates, the MILP consists of rate, interference and routing constraints. The interference constraints ensure the scheduling is conflict-free, while the rate and routing constraints guarantee that data travels from source to destination at a high enough rate. Two problems with the MILP are identified, namely that it is **NP**-hard and that the routing constraints only model the data flow in the long term, while failing to take the short-term state of each flow into account. The first problem is solved by relaxing the MILP to a linear program (LP). The second problem, meanwhile, is solved by constructing a short-term link scheduler that finds individual link assignments for each time slot. Specifically, the connection graph is transformed into a *multi-graph* where edges are duplicated by the number of time slots that the long-term scheduler assigned to the corresponding links. The short-term scheduler then performs the link scheduling by finding *edge colorings* in the multi-graph, where each color corresponds to a set of links that can be active concurrently. Because edge coloring too is an **NP**-hard problem, the short-term link scheduler is based on a greedy approach that only finds 2-approximate solutions to the problem.

Another network model based on the protocol model can be found in [13]. Here, nodes have separate transmission and interference distances much like in [16], which are used to construct an interference graph for the network. This interference graph then determines a set of interference constraints, which together with routing and rate constraints are used to formulate the scheduling as a mixed integer nonlinear program (MINLP). Like MILPs, MINLPs are **NP**-hard. To accomodate this, [13] decomposes the MINLP into a number of smaller problems. One of these problems is that of finding maximal independent sets in the interference graph. Another problem involves finding a transmit distance for the base station (BS), which acts as the source of all data streams in the model. In particular, it turns out that despite the BS being capable of performing 1-hop transmissions to all nodes in the network, the overall throughput can be increased by manually setting a maximum transmit distance, outside of which transmissions will have to be performed in multiple hops. The authors in [13] refer to these types of transmissions as *hybrid mode* transmissions, which consist of one *infrastructure* transmission from the BS to a network node, and one or more *ad hoc* transmissions from one network node to another.

In [30], the scheduling is again formulated as a MINLP involving interference and rate constraints. No routing constraints are needed as the network in this case is modeled as a set of predetermined data links. Unlike the previous examples, no concurrent SU transmissions using the same licensed band are allowed. Thus, even though the interference is here modeled using the physical model, the resulting

non-binary interference constraints are only between SUs and PUs. Interference constraints between SUs are instead implicitly binary, and correspond to fully connected interference graphs, one for each frequency band. Centralized algorithms, in addition to distributed implementations of them, are presented for optimally solving the MINLP. In the distributed implementation, the SUs and PUs cooperate in what resembles an auction with PUs as sellers and SUs as buyers of the licensed spectrum. The final transmission rights in each frequency band are determined via a virtual clock method, where the SU whose virtual clock expires first is allowed to transmit.

Network models based on interference graphs are again studied in [14]. As before, the scheduler tries to find independent sets in the interference graph to form the sets of transmitters that are to transmit in each time slot. Instead of rate constraints, the scheduler in this case is interested in queues of two different kinds. The first is the *virtual collision queue* associated with each PU, which lets SUs keep track of the number of collisions they have had with the PUs. The second is the *data queue* associated with each SU that holds the data packets that are to be transmitted. The goal is to guarantee the asymptotic convergence of data queues while keeping the virtual collision queues below a certain threshold. A distributed algorithm is presented that achieves both. This algorithm consists of a protocol where in each time slot transmitters broadcast some information about themselves to their 1-hop neighbors. The information includes nondeterministic contention variables which indicate whether the transmitter wants to contend for a certain band or not. After the broadcast messages, both the transmitter and receiver of each link can independently determine whether the link should be active in that time slot or not based on messages received from their neighbors.

Data queues are also considered in [22]. The scheduling in this case is based on the back-pressure algorithm first introduced in [31]. In the back-pressure algorithm, each transmission link is assigned a *link backlog* which is a function of the difference in queues sizes between the link's transmitter and receiver. The idea is then to prioritize links with higher backlogs in the scheduling, which ensures that no individual queue grows uncontrollably. The original back-pressure algorithm guarantees this sort of queue stability, but requires global knowledge of the network and thus must be performed centrally. Since this may not be a feasible option in a CR network, [22] develops a suboptimal distributed version of it. This distributed algorithm adapts CSMA/CA to coordinate the decisions made by each SU, and consists of four separate phases: flow selection (choosing receiver of a transmission), random backoff period, transmission, and forwarding decision.

Scheduling based on the physical interference model is considered in [19]. Once again the problem is formulated as a MINLP that includes routing, rate and interference constraints. In this case the interference constraints are not binary, but based on the SINR induced by all concurrently transmitted signals in the same band. The unused licensed spectrum is utilized via spectrum underlay. Thus, in addition to deciding how to allocate the spectrum among the SUs, the scheduling includes finding approriate transmit power levels for the chosen transmitters. A centralized algorithm that guarantees $(1-\epsilon)$-approximate solutions to the MINLP is proposed. Note that the MINLP only models a fixed state of the system. Hence, using the

proposed algorithm in practice would require that changes in the PU states happen less frequently than the time it takes to run the algorithm.

In [27], a stochastic variant of the physical model is used. In particular, it is assumed that the received signal S and interference I power levels in (1) are random and cannot be measured by their instantaneous values. Instead, they must be estimated based on their mean values measured over time. This adds uncertainty to the scheduling decisions made. Specifically, it cannot be guaranteed that PUs remain undisturbed by the SUs' transmissions, or that SU transmissions do not interfere with each other. The interference constraints in this case only require that the instantaneous interference constraints must be met on the average. These constraints correspond to explicit bounds on the violation probabilities of the instantaneous constraints.

The techniques reviewed so far have all used conventional half-duplex nodes. That is, they must commit to a single frequency band in which they either transmit or receive. In [29], the nodes are instead equipped with MIMO (multiple-input and multiple-output) antennas. This allows them to not only utilize multiple frequency bands for a single transmission, but also dedicate a portion of their antennas to *interference cancellation*. In interference cancellation, an antenna is used to listen to an unwanted signal, which allows the receiving node to decode the transmission and reduce the interfering power. The underlying interference model in this case is the connection graph. Thus, an SU can successfully receive a signal from one of its neighbors in a given frequency band if one of two conditions are met. Either none of its other neighbors are transmitting in the band. Alternatively, for each unwanted transmitting neighbor, it has a dedicated interference cancelling antenna that cancels the interference caused by that neighbor. This leads to a tradeoff between using as many antennas as possible for reception to increase the received signal rate, and dedicating enough antennas towards interference cancellation.

2.5 Motivation for This Thesis

We have now studied the problem of scheduling in CR networks in some detail, including existing solutions to it. The chapter will be concluded by discussing where this thesis fits into the big picture. In particular, we will specify the components discussed in section 2.3 that comprise the particular problem studied in this thesis. Additionally, we will try to identify some key areas not covered by existing solutions, that the methods presented later in this thesis aim to improve upon.

In this thesis, the type of CR scheduling problem we are interested in is the following:

1. **Interference model:** The *physical interference model* will be used to model interferences. This provides a more accurate model of the interference than the protocol model, but also makes it more challenging to design algorithms for it.

2. **Spectrum bands:** We will consider *multi-band* networks. Scheduling decisions will therefore have to consider which frequency band scheduled transmissions should use.

3. **Network connectivity:** The network is assumed to be *multi-hop*, with *no predetermined datalinks*. Thus, scheduling decisions will also need to consider the routing of data. This means that for each scheduled transmission we will need to decide the receiver of the transmission in addition to the aforementioned frequency band.

4. **Scheduling mode:** We will design both *centralized* and *distributed* scheduling methods. The centralized method will mostly be presented as an abstract blueprint which the distributed method will provide a more realistic implementation of.

5. **Spectrum utilization technique:** The licensed spectrum will be accessed by SUs via a *combination of spectrum over- and underlay*. Specifically, SUs are allowed to use any licensed band regardless of PU activity, but will adjust their transmit powers based on the PU activity to ensure that they do not interfere with the PU transmissions.

6. **Objective:** We will look at the latency or *delay* of data packets. Our main interest will be the minimization of the mean delay of packets, corresponding to maximizing the overall throughput of the network. However, we will also try to minimize the maximum delays experienced by packets in order to ensure fairness in the scheduling.

To the best of the author's knowledge, there is no study that sufficiently covers the particular type of problem described above, that is, problems that fall into all six of the categories specified. Some works like [19] study problems that fall into the first five categories, but with an objective of maximizing the sum data rate of the network. This alone justifies the existence of the thesis. Furthermore, most related research focuses on designing solutions that optimally solve the problem, but do so at the cost of a high computational complexity. Because the available spectrum in CR networks varies as a function of time, we believe it is crucial to instead design algorithms of low complexity that can utilize the licensed spectrum among SUs in a timely manner before the spectrum state changes. The scheduling methods presented in this thesis, in particular the distributed one, address this by shifting focus from the optimality of the solution to ensuring low running times. Finally, in most works the routing in CR networks ignores the fact that certain portions of the spectrum may be less likely to be available than others. To address this, we will utilize a novel potential field modeling technique that uses any information we can gather about the PUs and their activities to design routes that better navigate via the spectrum holes left by the PUs, i.e. where there is more likely to be available spectrum.

3 Scheduling Framework

In this chapter we will formally define the problem to be studied in this thesis. We start off by specifying the different components that comprise our system model in section 3.1. The interference model, which plays a central role in determining when and where communication can take place, is then introduced in section 3.2. Finally, in section 3.3, the scheduling problem is defined as two separate tasks: a long-term problem of assigning weights to links, and a short-term problem of finding a minimum weight set of transmission links that can be active simultaneously under the model described in sections 3.1 and 3.2.

3.1 System Model

We consider a multi-hop CR network with SUs $\mathcal{N} = \{1, \ldots, i, \ldots, N\}$ attempting to opportunistically utilize a set of licensed frequency bands $\mathcal{M} = \{1, \ldots, m, \ldots, M\}$ for their own transmissions. In each licensed band $m \in \mathcal{M}$, there is a stationary PU transmitter and a corresponding PU receiver listening to its transmissions. The location $\mathbf{x}_{\mathtt{TX}}^m$ of the PU transmitter, as well as the transmit power $P_{\mathtt{TX}}^m$ used by it, is assumed to be known by the SUs. In addition, it is assumed that the SUs have through observations determined the probability z^m that the PU transmitter transmits at any given time instance. For the PU receiver, the transmit power and probability are clearly 0. However, its location $\mathbf{x}_{\mathtt{RX}}^m$ is not known to the SUs.

Each SU is equipped with a single half-duplex transceiver that can transmit or receive in one frequency band at a time. The transceivers operate in a synchronized time-slotted manner, with each time slot further divided into *sensing*, *control* and *transmission* minislots. In the sensing minislot, SUs sense the spectrum to identify the state $b^m \in \{0, 1\}$ of each PU transmitter, where $b^m = 1$ indicates that the PU transmitter in m is transmitting and $b^m = 0$ that it is idle. They then use the control minislot to exchange any information necessary to make the scheduling decisions over a *common control channel* (CCC) that is outside the licensed spectrum. In the transmission minislot, SUs that have been scheduled for transmission during the control minislot transmit data.

The data to be transmitted is modeled as packets p, each of the same constant size and with a given source and destination SU, $\mathbf{src}(p)$ and $\mathbf{dst}(p)$, respectively. When a packet arrives at its source, the scheduler will attempt to schedule transmissions that transport the packet to its destination with as little delay $\mathbf{dly}(p)$ as possible. While en route, the packet is stored in a first-in-first-out (FIFO) queue q_i at the SU i currently in possession of the packet. The nth packet in queue q_i is denoted $q_i[n]$ (where $n = 1, 2, \ldots$), and its current length $|q_i|$. There is no upper limit on $|q_i|$, and packets remain in the queue until they are transmitted. The capacity of all links in the network is assumed to be either C or 0 bits per second, and the length of a transmission minislot equal to the time needed to transmit one packet over links of capacity C. Thus, when SU i is scheduled to transmit, it will transmit packet $q_i[1]$, and, if the transmission is successful, the queue is updated as $q_i[n] \leftarrow q_i[n+1]$ for $n = 1, 2, \ldots, |q_i| - 1$.

The scheduling decisions are made for each time slot in separate, and consist of two parts, $\mathbf{a}$ and $\mathbf{P}$. The decision vector $\mathbf{a} \in \{0,1\}^{|\mathcal{N}|\times|\mathcal{N}|\times|\mathcal{M}|}$ contains the decision variables $a_{ij}^m \in \{0,1\}$ that indicate the transmitter, receiver and frequency band of each scheduled transmission. In particular, $a_{ij}^m = 1$ if SU i is to transmit to SU j using frequency band m, and $a_{ij}^m = 0$ otherwise. The power vector $\mathbf{P} \in \mathbb{R}^{|\mathcal{N}|\times|\mathcal{M}|}$ contains the transmit powers $P_i^m \geq 0$ to be used by each SU i in frequency band m.

For a data packet p, the source and destination may be any pair of distinct SUs in the network, i.e. $\mathbf{src}(p) \in \mathcal{N}$ and $\mathbf{dst}(p) \in \mathcal{N} \setminus \{\mathbf{src}(p)\}$. Because the network is multi-hop, this means that $\mathbf{src}(p)$ and $\mathbf{dst}(p)$ are generally not 1-hop neighbors. Thus, in addition to determining who should be allowed to transmit and using which frequency band, the scheduler must find paths for the packets from their sources to their destinations. As the scheduling is to be performed one time slot at a time, and the spectrum state may vary from one time slot to another, we will not attempt to construct end-to-end paths for the packets. Instead, the packets will be routed opportunistically by assigning weights to the links, with the specific receiver at each step depending on the spectrum state.

3.2 Interference Model

In this thesis, we will consider the physical interference model discussed in the previous chapter. The links in the network are represented by triples $(i,k,m) \in \mathcal{N} \times \mathcal{N} \times \mathcal{M}$ that each have an SINR

$$\gamma_{ik}^m = \frac{S_{ik}^m}{N_k^m + I_k^m} \tag{2}$$

associated with them. Before properly defining the signal S_{ik}^m, noise N_k^m and interference I_k^m terms, we will discuss the role of the SINR in determining the existence, or availability, of links in the network.

The capacity of a link is assumed to be either 0 or C bits per second, where C is a constant value that corresponds to a bitrate at which the transmission of a packet will take exactly the duration of the transmission minislot of a time slot. The capacity of a link (i,k,m) is C if and only if, for a predefined threshold value Γ (that depends on C) we have $\gamma_{ik}^m \geq \Gamma$, in which case we say that the link is *available*. A link that is not available has a capacity of 0 bits per second and thus can not be used for transmission. For a transmitter i, the set of available links (i,k,m) for a fixed m determine i's *m-neighborhood*

$$\mathcal{N}_i^m = \{k \in \mathcal{N} \mid k \neq i \text{ and } \gamma_{ik}^m \geq \Gamma\}. \tag{3}$$

The union of SU i's m-neighbors, $\bigcup_{m=1}^{M} \mathcal{N}_i^m$, then defines the *transmission neighbors* $\mathcal{N}_i \subseteq \mathcal{N}$ of i, consisting of all SUs k that i can transmit to.

Now that we know how the SINR is used to determine the availability of links in the network, we will next discuss the underlying physical model in more detail. Transmitted signals experience *path loss* by a factor g, $0 < g < 1$, before reaching a receiver. That is, the received power of a signal transmitted with power P is gP. We assume a simplified model where the path loss g depends only on the distance between the transmitter and receiver. If we let $d(\mathbf{x},\mathbf{y})$ denote the distance between

points $\mathbf{x}$ and $\mathbf{y}$, the path loss between a transmitter located at $\mathbf{x}_{\mathtt{TX}}$ and a receiver located at $\mathbf{x}_{\mathtt{RX}}$ is assumed to be

$$g(\mathbf{x}_{\mathtt{TX}}, \mathbf{x}_{\mathtt{RX}}) = (1 + d(\mathbf{x}_{\mathtt{TX}}, \mathbf{x}_{\mathtt{RX}}))^{-\alpha}, \tag{4}$$

where α is a known environment-dependent path loss exponent, typically in the range 2 to 5. To emphasize the different roles and amount of available information of SUs and PUs, we will use slightly different notation for the path loss factor g (and the corresponding distance d) depending on whether the transmitter and/or receiver is a PU:

- Between SUs i and j, the distance is denoted d_{ij}, and the corresponding path loss g_{ij}.
- Between the PU transmitter in band m and an SU i, the distance is denoted $d^m_{\mathtt{TX},i}$, and the corresponding path loss $g^m_{\mathtt{TX},i}$.
- Between an SU i and the PU receiver in band m, the distance is denoted $d^m_{i,\mathtt{RX}}$, and the corresponding path loss $g^m_{i,\mathtt{RX}}$.
- Between the PU transmitter and receiver in band m, the distance is denoted $d^m_{\mathtt{TX},\mathtt{RX}}$, and the corresponding path loss $g^m_{\mathtt{TX},\mathtt{RX}}$.

The received signal power at SU k in frequency band m for a signal originating from SU i is now

$$S^m_{ik} = g_{ik} P^m_i. \tag{5}$$

Meanwhile, the interference received by SU k in frequency band m consists of all unwanted signals in the band. Here we separate the interference J^m_k caused by the PU transmitter in band m, and all interference signals I^m_{kj} originating from other SUs j

$$I^m_k = J^m_k + \sum_{j=1}^{N} I^m_{kj}. \tag{6}$$

Using the assigned decision variables a^m_{ij} for SUs and observed variables b^m describing the state of PUs we then have

$$J^m_k = b^m g^m_{\mathtt{TX},k} P^m_{\mathtt{TX}}, \quad I^m_{kj} = \sum_{\substack{l=1 \\ l \neq k}}^{N} a^m_{jl} g_{jk} P^m_j. \tag{7}$$

For the noise power, we assume a constant value σ_{SU} that does not depend on the location or frequency band:

$$N^m_k = \sigma_{\mathrm{SU}}. \tag{8}$$

Putting it all together, (2) can now be written as

$$\gamma^m_{ik} = \frac{g_{ik} P^m_i}{\sigma_{\mathrm{SU}} + b^m g^m_{\mathtt{TX},k} P^m_{\mathtt{TX}} + \sum_{j=1}^{N} \sum_{\substack{l=1 \\ l \neq k}}^{N} a^m_{jl} g_{jk} P^m_j}. \tag{9}$$

In addition to the SINR at SUs, we are also concerned with the SINR at the PU receivers. In this case, since there is only one PU transmitter-receiver pair per frequency band, all possible interference will come from the SUs. Letting $\gamma^m_{\mathtt{TX,RX}}$ denote the SINR at the PU receiver in band m, we will then have

$$\gamma^m_{\mathtt{TX,RX}} = \frac{g^m_{\mathtt{TX,RX}} P^m_{\mathtt{TX}}}{\sigma_{\mathrm{PU}} + \sum_{j=1}^{N} \sum_{l=1}^{N} a^m_{jl} g^m_{j,\mathtt{RX}} P^m_j}, \tag{10}$$

where σ_{PU} is the noise as measured by the PUs.

3.3 Formulation of the Problem

The goal of the scheduling is to minimize the end-to-end delays of packets, that is, the number of time slots that it takes for packets to reach the destination from the source. However, finding long-term solutions may not be practical as the activity of the PUs, and hence spectrum state, in any future time slot is not known to us in advance. In particular, we do not know the specific value of b^m for a future time slot until it is observed at the beginning of the time slot. For this reason, the scheduling will be split into two separate tasks: the long-term problem of minimizing the end-to-end delays, and the short-term problem of finding the specific transmission links in each time slot that allow us to meet the long-term goal.

For the long-term problem, there are two aspects to consider. First, since the source and destination of a packet are generally not 1-hop neighbors, we need to find appropriate routes for the packets. Second, the amount of available spectrum is limited, and thus we must decide how to allocate the spectrum in a manner that jointly minimizes the end-to-end delays of *all* packets, possibly taking fairness into account. Since the specific transmission links used in each time slot are ultimately decided by our short-term scheduler, the task of the long-term scheduler is to assign weights to links to aid the short-term scheduler make decisions in accordance with the aspects discussed above. Specifically, we want to assign weights w^m_{ij} to each link (i, j, m) that satisfy the following properties:

1. $w^m_{ij} < 0$, if transmitting packet $q_i[1]$ over link (i, j, m) is expected to *reduce* the end-to-end delay of $q_i[1]$ compared to not transmitting at all.

2. $w^m_{ij} > 0$, if transmitting packet $q_i[1]$ over link (i, j, m) is expected to *increase* the end-to-end delay of $q_i[1]$ compared to not transmitting at all.

3. $w^m_{ij} = 0$, if $|q_i| = 0$ or transmitting $q_i[1]$ over link (i, j, m) is not expected to affect the end-to-end delay of $q_i[1]$ compared to not transmitting at all.

4. If $w^m_{ij} < 0$ and $w^{m'}_{i'j'} < 0$, we have $w^m_{ij} < w^{m'}_{i'j'}$ if transmitting packet $q_i[1]$ over link (i, j, m) is expected to reduce the end-to-end delay of $q_i[1]$ more than transmitting packet $q_{i'}[1]$ over link (i', j', m') is expected to reduce the end-to-end delay of $q_{i'}[1]$.

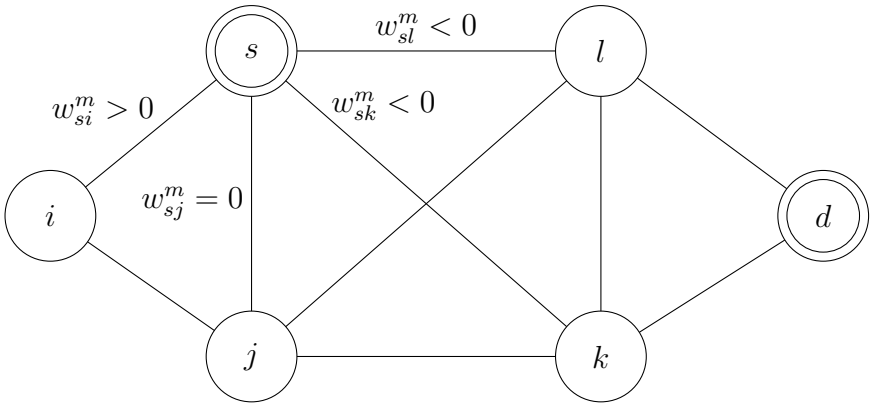


Figure 6: Toy example of a network and the weights associated with outgoing links from node s when using the protocol model to determine link availability. Node d is the destination of packet $q_s[1]$. For simplicity, we assume a single frequency band m.

The first three properties address the routing of packets, and we will refer to these as the *routing properties* of the weights w_{ij}^m. The fourth property, meanwhile, referred to as the *ranking property*, ranks the weights according to how much transmitting over the corresponding link affects the end-to-end delay of the packet to be transmitted over the link.

As an example, consider the network shown in Fig. 6, where the protocol model is used to determine the availability of links. Each packet has a shortest distance to its destination, and a transmission can either reduce this distance by 1 (links (s, k, m) and (s, l, m) for packet $q_s[1]$), increase the distance by 1 (link (s, i, m) for packet $q_s[1]$), or keep the shortest distance as is (link (s, j, m) for packet $q_s[1]$). A natural interpretation of the routing properties would then be to assign weights as shown in the figure. In this case a transmission can at most reduce the distance by 1, so it is not immediately clear how the ranking property would be applied. One possibility is to consider the packet queues at receiving nodes. In particular, we might have $|q_l| = 0$ and $|q_k| > 0$ and thus expect an added queueing delay when transmitting to k. We would then want the assigned weights to satisfy $w_{sl}^m < w_{sk}^m$.

Note that the weight w_{ij}^m depends on the state of q_i, and hence must be decided for each time slot separately. This is what we wanted to avoid in the long-term scheduling. Therefore, the long-term problem is not to decide the specific value of each w_{ij}^m, but to design a *weight function* $\mathtt{w} : \mathcal{N} \times \mathcal{N} \times \mathcal{M} \mapsto \mathbb{R}$ that the short-term scheduler can use to decide the link weights in each time slot.

Now, assuming we have defined the weight function, the short-term problem in any given time slot is to find a set of transmission links that can be activated simultaneously and whose sum of weights is as small as possible. More formally, we want to determine the values of the decision vector $\mathbf{a}$ and power vector $\mathbf{P}$ based on the observed PU states b^m and the weights w_{ij}^m computed according to the weight

function such that the following problem is solved

$$\min_{\mathbf{a},\mathbf{P}} \quad \sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij}^{m} w_{ij}^{m} \tag{11}$$

$$\text{s.t.} \quad \gamma_{ij}^{m} \geq a_{ij}^{m}\Gamma, \qquad \forall\, i \in \mathcal{N}, j \in \mathcal{N}, m \in \mathcal{M} \tag{12}$$

$$\gamma_{\mathtt{TX,RX}}^{m} \geq b^{m}\Gamma, \qquad \forall\, m \in \mathcal{M} \tag{13}$$

$$P_{i}^{m} \leq P_{\max}, \qquad \forall\, i \in \mathcal{N}, m \in \mathcal{M} \tag{14}$$

$$P_{i}^{m} \geq 0, \qquad \forall\, i \in \mathcal{N}, m \in \mathcal{M} \tag{15}$$

$$\sum_{m=1}^{M}\sum_{i=1}^{N} a_{ij}^{m} + \sum_{m=1}^{M}\sum_{k=1}^{N} a_{jk}^{m} \leq 1, \qquad \forall\, j \in \mathcal{N} \tag{16}$$

$$\text{and} \quad a_{ij}^{m} \in \{0,1\}, \qquad \forall\, i \in \mathcal{N}, j \in \mathcal{N}, m \in \mathcal{M}. \tag{17}$$

Constraints (12) and (13) are interference constraints, with (12) ensuring that all scheduled SU transmission links have a sufficiently high SINR, and (13) doing the same for PU transmissions. Meanwhile, constraints (14) and (15) are physical contraints for the transmission power, with the constant $P_{\max}$ found in the former determined by the physical properties of the transceivers. Constraints (16) are due to the half-duplex nature of the transceivers. That is, a transceiver cannot transmit and receive at the same time, and doing either can only be done using one of the M frequency bands. There are no routing constraints, as they are already implicitly contained in the weights w_{ij}^{m}.

Note that we have abbreviated the left hand side in constraints (12) and (13). Replacing γ_{ij}^{m} and $\gamma_{\mathtt{TX,RX}}^{m}$ by their full expressions as defined by (9) and (10), we can see that both constraints contain products of the optimization variables P_{i}^{m} and a_{ij}^{m}, and are thus nonlinear. Combined with (17) constraining the variables a_{ij}^{m} to (binary) integers, this means that the problem we are dealing with is a mixed integer nonlinear program, or MINLP. As has already been discussed, MINLPs are **NP**-hard. Because the solution only applies to one time slot, we may thus not have enough time to find the optimal solution before it becomes obsolete. For this reason, we will not require optimality of our solution, and instead focus on designing *low-complexity* algorithms that approximately solve the problem. No explicit approximation bounds will be established. Instead, the performances of the proposed algorithms are measured by the end-to-end delays of packets achieved by applying the algorithms over a large number of consecutive time slots.

Furthermore, recall that we do not know the locations of PU receivers. Thus, both $g_{\mathtt{TX,RX}}^{m}$ and $g_{j,\mathtt{RX}}^{m}$ are unknown to us, and satisfying constraints (13) with certainty given the information we have would require taking a conservative approach and assuming the worst case. However, as we will see later, this leads to unnecessarily strict bounds on the transmission power of SUs. We will therefore relax constraint (13). We do this by making it a *soft* constraint, meaning that we do not require a solution to satisfy (13) to be considered feasible, but would still like to avoid violating it. Similar to the objective function, we will not set explicit violation bounds but instead measure the violation probability achieved by our algorithms.

To summarize, the scheduling problem considered in this thesis consists of two parts. First, we want to design a weight function that assigns weights w_{ij}^m to links in the network, such that solving the MINLP defined by (11)-(17) in every time slot leads to small end-to-end delays for packets in the long term. Second, we want to design a low-complexity algorithm that approximately solves the (relaxed) MINLP.

4 Centralized Scheduling

In this chapter a centralized scheduling approach is considered. The method developed as a part of this thesis work and published in [5] is described in detail. The long-term scheduling problem is solved in section 4.1 by utilizing a novel potential field modeling technique that was originally introduced in [4]. The short-term scheduling problem, meanwhile, is solved in section 4.2 by means of a polynomial-time greedy algorithm. This algorithm iteratively updates a partial solution by fixing an increasing subset of the optimization variables, until the objective can no longer be improved without modifying the already fixed subset of optimization variables.

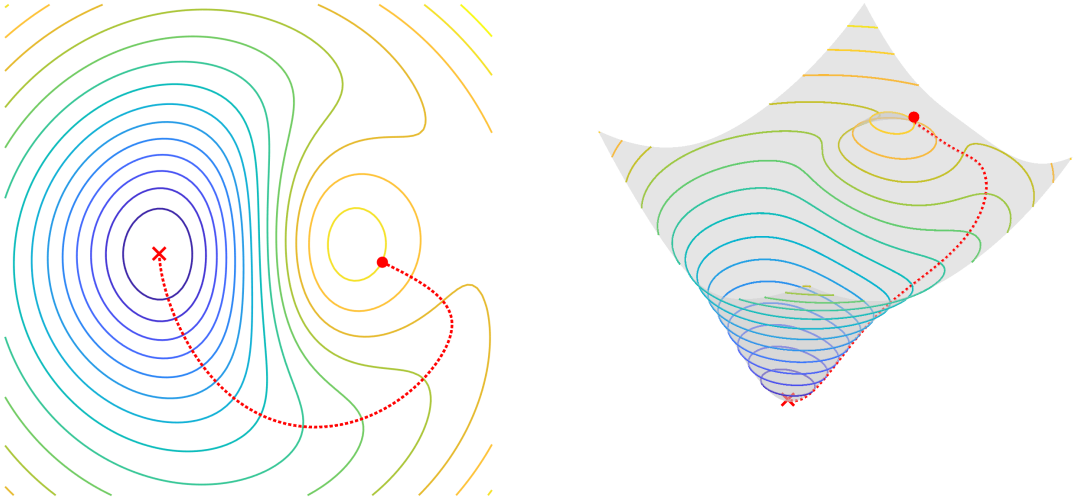
4.1 Solving the Long-Term Problem

In our earlier discussion of the weight function, we briefly examined the case of assigning weights to network links whose availability was based on the protocol model. Under this model, the expected delay of packets needed to determine the weights was related to the length of the shortest path in the underlying connection graph. Although there is no direct equivalent of shortest paths in the physical interference model studied in this thesis, we can extend this line of thinking by instead considering *physical shortest paths*. Rather than being link based, such paths are based on the geometry and physical properties of the interferences involved. In particular, we construct potential fields that model the physical aspects, and let the negative gradient of these fields define the physical shortest paths. We then design a weight function that utilizes not only information about the physical shortest paths, but also the packet queues involved.

4.1.1 Routing Using Potential Field Modeling

The neighborhoods $\mathcal{N}_i$, i.e. the available links in the network, depend on the observed PU state vector $\mathbf{b} = (b^1, \ldots, b^M)^{\mathrm{T}}$ as well as the choices of the decision vector $\mathbf{a}$ and power vector $\mathbf{P}$ in each time slot. Hence, it does not make sense to determine end-to-end paths for packets to travel along as there is no way of knowing beforehand what paths will be available. Instead, we will design *potential fields* that abstract the path selection away from sequences of individual network nodes and onto an underlying force field that simply directs the packets towards their destinations while trying to avoid areas with bad spectrum opportunities. The specific relay node at each individual step can then be chosen opportunistically based on the current spectrum availability.

Each SU $j \in \mathcal{N}$ can act as the destination of a packet, and has its own potential field associated with it. Our goal is to design these potential fields in such a way that a packet following their negative gradient will reach the destination with minimum delay. The intuitive interpretation of the potential field is that it pushes packets from higher to lower potentials, much like the gravitational potential field of a hill would push a ball downwards, as shown in Fig. 7. In this case, the global minimum of the field where the packet eventually ends up would be the destination SU j, and any



(a) (b)

Figure 7: The gravitational potential field of a hill enclosed in a valley, with (a) showing the valley directly from above and (b) from an angle. The solid colored curves show the elevation, and consequently the potential inside the valley, with high (potential) indicated by yellow and low (potential) by blue. The dashed red line shows the path defined by the negative gradient of the field, i.e. the path along which the red ball will roll down to the bottom of the valley if we disregard its momentum and if there are no external forces.

areas we want to avoid along the way would be modeled as hilltops corresponding to local maxima. For example, we may expect the spectrum opportunities near PUs to be more scarce than further away from them, and thus model the PU's locations as local maxima in the field.

Although the gravitational potential field of the valley shown in Fig. 7 provides an intuitive example for how the potential fields designed in this thesis function, we should note that there is one fairly significant difference between the two. Namely, in the potential fields designed in this thesis the packets can only reside in distinct locations (the locations of SUs), and as a result the actual routes taken by packets will not strictly follow the negative gradient of the field as suggested earlier. In fact, as we will see later, it is not even in our best interest for the packets to strictly follow the negative gradient of the potential field. Therefore, we will not require that the potential field we design is differentiable, or that it is defined outside of the locations of SUs. Letting Φ_{ij} denote the potential at SU i w.r.t. destination SU j, it will thus suffice to compute the value Φ_{ij} for all $i \in \mathcal{N}$ to determine the potential field associated with destination j.

When designing the potential fields, we would like the potential at the destination to be zero, i.e. $\Phi_{jj} = 0$. Because the destination is the global minimum, we should then also have $\Phi_{ij} > 0$ for all $i \neq j$. Finally, if the delay from an SU i to destination j is expected to be lower than the delay from an SU i' to j, we want the potential field to satisfy $\Phi_{ij} < \Phi_{i'j}$. In the following, three different types of potential fields

are discussed. The latter two are extended versions (from single- to multi-band) of potential fields originally introduced in [4]. We will inspect visual representations of these potential fields in chapter 6.

Greedy Potential Field: Our first, naive, attempt at a potential field is the *greedy potential field* (GPF). In the GPF, packets are routed directly towards the location of the destination SU, disregarding any possible sources of interference on the way. Specifically, the potential at SU i w.r.t. destination SU j is simply the Euclidean distance d_{ij} between i and j:

$$\Phi_{ij} = d_{ij}. \tag{18}$$

It is easy to very that $\Phi_{jj} = 0$, and $\Phi_{ij} > 0$ for all $i \neq j$ as we wanted. Additionally, we will have $d_{ij} > d_{i'j} \implies \Phi_{ij} > \Phi_{i'j}$, meaning that the further away from the destination an SU is, the longer we would expect the packet delay between this SU and the destination to be, which seems sensible.

Virtual Force Field: Completely ignoring the interference as in GPF may lead to packets getting stuck in high-interference areas where spectrum opportunities are scarce and thus lead to unacceptably high delays for these packets. However, if we knew what areas are more likely to contain interference, we could model this as repulsive forces that increase the potential when approaching them. In particular, we note that the PU transmitters are stationary and have priority for their transmissions. Hence, whenever a PU is transmitting, this leads to unavoidable interference to SUs in the area surrounding the PU.

Inspired by this observation, we define the *virtual force field* (VFF) to be a potential field that contains two types of potentials: the *attractive* potential $\underline{\Phi}_{ij}$ of the destination j, and the *repulsive* potential $\overline{\Phi}^m_{ij}$ of the PU transmitter in each frequency band m. The attractive potential is similar to the GPF potential, but uses a user-defined scaling factor ν to adjust the contribution of the attractive potential in the overall potential:

$$\underline{\Phi}_{ij} = \frac{1}{2}\nu(d_{ij})^\alpha. \tag{19}$$

Meanwhile, the repulsive potential of the PU transmitter in band m is defined as

$$\overline{\Phi}^m_{ij} = \begin{cases} \frac{\xi(d_{ij})^\alpha z^m}{2}\left(\frac{1}{d^m_{\mathrm{TX},i}} - \frac{1}{d^m_{\mathrm{B}}+d_{ij}}\right)^\alpha, & \text{if } d^m_{\mathrm{TX},i} \leq d^m_{\mathrm{B}} + d_{ij} \\ 0, & \text{otherwise.} \end{cases} \tag{20}$$

where ξ is a user-defined scaling factor (corresponding to ν in the attractive potential), and d^m_{B} a virtual boundary distance that corresponds to the distance from the PU transmitter in band m at which no secondary communication can take place if the PU transmitter is transmitting (see [4] for details).

The potential value at SU i w.r.t. destination SU j is now

$$\Phi_{ij} = \underline{\Phi}_{ij} + \min_{m\in\mathcal{M}} \overline{\Phi}^m_{ij}. \tag{21}$$

Here we take the minimum of all repulsive potentials to reflect the fact that the frequency bands are orthogonal: an SU i that is close to one or more PU transmitters may still have plenty of spectrum opportunities in bands not occupied by the PU transmitters close to it.

Single-Session Delay-Optimal Potential Field: Instead of trying to model the interference explicitly via repulsive forces, we could do so implicitly by considering how the interference affects the availability of links. In particular, since we know the locations $\mathbf{x}_{\mathtt{TX}}^m$ of PU transmitters as well as their transmit powers $P_{\mathtt{TX}}^m$, we know for any possible PU state vector $\mathbf{b}$ what links are available if there are no SUs transmitting. Furthermore, we know the probability z^m that $b^m = 1$ for each $m \in \mathcal{M}$ and therefore also the probability of each $\mathbf{b}$. Thus, we can determine the probability that a link (i, j, m) is available if there are no ongoing SU sessions.

Now, let us consider a simplified scenario where only one packet is to be transported from source to destination (a *single-session* scenario). That is, there will be no interference from other SUs and we only have interference from PUs to worry about. Then, by the discussion above, the network can be modeled as a connection graph similar to the protocol model, but with each link having a certain (known) probability of availability associated with it. These probabilities give us an expected delay of transmitting over each link. Let p_{ij}^m be the probability that link (i, j, m) is available. Either the link is available (probability p_{ij}^m) and we can transmit over it (1 time slot), or it is occupied (probability $(1 - p_{ij}^m)$) and we need to wait (for an expected $1/p_{ij}^m$ time slots) for it to become available before transmitting (1 time slot), giving an expected delay of

$$p_{ij}^m \times 1 + (1 - p_{ij}^m) \times (\frac{1}{p_{ij}^m} + 1) \tag{22}$$

time slots to transmit over the link. Extending this reasoning to paths in the connection graph, we can then determine the *minimum expected delay* between any pair of SUs in the network. This is indeed what we will do in the *single-session delay-optimal potential field* (SSDOPF), where the potential value Φ_{ij} denotes the minimum expected delay between nodes i and j when no other SU sessions are active. However, we will not directly use Eq. (22).

Notice that in the single-session scenario considered, there is no reason to favor a link (i, j, m) over another link (i, j, m') if they are both available. Thus, if we ignore the frequency band m of links, $(i, j, m) \mapsto (i, j)$, the PU states $\mathbf{b}_1 = (b^1, \ldots, b^{m-1}, 0, b^{m+1}, \ldots, b^M)^{\mathrm{T}}$ and $\mathbf{b}_2 = (b^1, \ldots, b^{m'-1}, 0, b^{m'+1}, \ldots, b^M)^{\mathrm{T}}$ where $m \neq m'$ are indistinguishable from each other. In particular, if link $(i, j, m) \mapsto (i, j)$ is available given state $\mathbf{b}_1$, then so is link $(i, j, m') \mapsto (i, j)$ given state $\mathbf{b}_2$, and vice versa. For this reason, we only need to separate between the following two spectrum states when determining link availability: if $b^m = 0$ for *some* band m, the spectrum is *idle*; if $b^m = 1$ for *all* bands m, the spectrum is *occupied*. In the former case, only the transmitter and receiver of links are considered, i.e. $(i, j, m) \mapsto (i, j)$. Assuming that the activities of different PUs are independent, the probability of the spectrum

being occupied is

$$Z = \prod_{m \in \mathcal{M}} z^m, \tag{23}$$

and the probability that it is idle $1 - Z$.

Now let C_{ij}^0 denote the delay of transmitting over link (i, j) in idle spectrum, and $C_{ij}^{1,m}$ the delay of transmitting over link (i, j, m) in occupied spectrum. This delay is 1 if the link is available, otherwise it is ∞. The potential values are then computed iteratively as follows. Initially, $\Phi_{ij} = \infty$ for all $i \neq j$ and $\Phi_{jj} = 0$. Then, we repeat for each $i \neq j$ until convergence:

1. Compute the minimum expected delay in idle spectrum:

$$D_{ij}^0 = \min_{k \in \mathcal{N}} C_{ik}^0 + \Phi_{kj}. \tag{24}$$

2. Compute the minimum expected delay in occupied spectrum:

$$D_{ij}^1 = \min \left\{ \underbrace{\min_{k \in \mathcal{N}} \min_{m \in \mathcal{M}} C_{ik}^{1,m} + \Phi_{kj}}_{\text{Transmit at occupied band}}; \quad \underbrace{\frac{1}{1 - Z} + D_{ij}^0}_{\text{Wait for an idle band}} \right\}. \tag{25}$$

3. Update the potential value, i.e. the minimum expected delay, as:

$$\Phi_{ij} = (1 - Z) D_{ij}^0 + Z D_{ij}^1. \tag{26}$$

The algorithm is run until no more potential values change over the course of an iteration. Note that unlike the GPF and VFF, the SSDOPF is not differentiable. However, as mentioned earlier this will not be a problem as we will not be using the gradient of the potential fields.

4.1.2 The Weight Function

Recall that the purpose of the weight function is to assign weights to network links that satisfy the routing and ranking properties defined in the previous chapter. If the weights satisfy the routing properties, we ensure that the short-term scheduler will only schedule links that reduce the delays of packets (compared to not transmitting). If the weights also satisfy the ranking property, the short-term scheduler will prioritize links according to how much transmitting over them reduces the delay of the packets transmitted. In our earlier example using the protocol model, it was possible to find weights that satisfied these properties by considering the shortest paths involved. Now that we have designed potential fields that define corresponding shortest paths under the physical interference model, we can use these to construct the weight function.

The shortest paths in this case are determined by the negative gradient of the potential fields. However, as the packets only have distinct locations that they can reside in along the path (the locations of the SUs), we cannot directly use the gradient

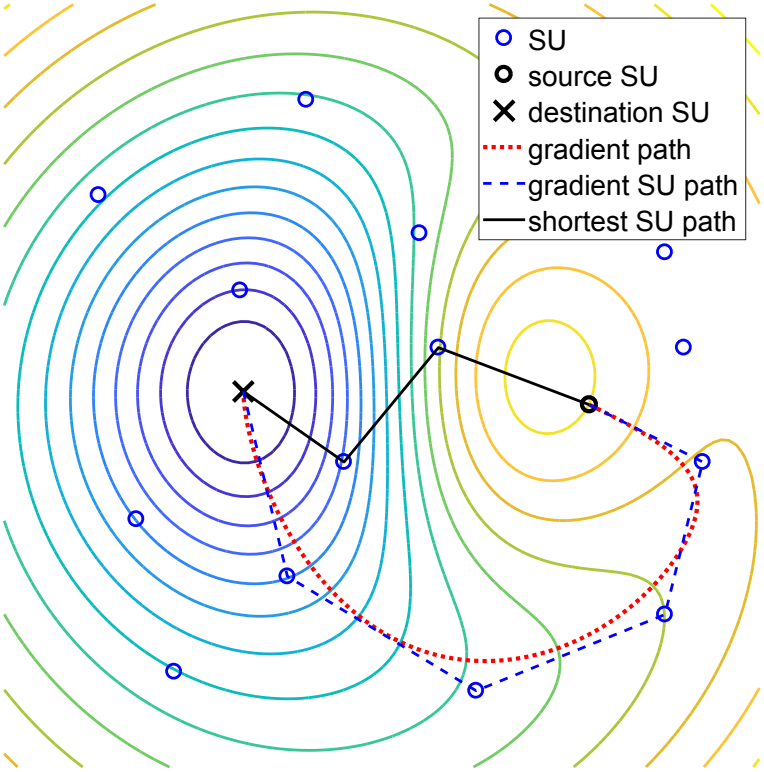


Figure 8: Example potential field used for packet routing in a CR network. Note that this potential field is the same as the gravitational potential field of the valley in Fig. 7, and not based on any of the proposed potential fields. The goal is to route a packet from the source (black circle) to the destination (black cross). The red dotted line shows the path from source to destination as defined by the negative gradient of the field, and the blue dashed line the corresponding path for the packet in the CR network. The black solid line shows an alternative path where each SU simply forwards the packet to the neighbor with the lowest potential. Although the first link along the black path goes directly over a local (and in this case also global) maximum and thus is unlikely to be available, there is no reason not to use this path *if* the link happens to be available, as the packet would end up in lower potential than it would after the first transmission along the gradient path. Additionally, the total path length is then reduced considerably compared to the gradient path.

of the potential field to decide their routes. Furthermore, it may be the case that a transmission allows us to take a shortcut through the potential field due to favorable spectrum conditions. For instance, say the negative gradient of a potential field would route a packet from i to j via k, because the link (i, j, m) is not likely to be available for any $m \in \mathcal{M}$. But in this case, the link (i, j, m) *is* available, so we can use this link to get the packet directly to j in one hop instead of two. An example of this is shown in Fig. 8. Therefore, it is not even in our best interest to strictly follow the direction of the negative gradient of the potential field. Instead, we want to find transmissions in each hop that *reduce* the potential as much as possible,

which from the point of view of an SU i simply means that it should transmit to the transmission neighbor $j \in \mathcal{N}_i$ with the lowest potential $\Phi_{j,\mathbf{dst}(q_i[1])}$. In particular, we note that link weights w_{ij}^m that are proportional to the difference in potential between the transmitter i and receiver j, i.e. $w_{ij}^m \propto \Phi_{j,\mathbf{dst}(q_i[1])} - \Phi_{i,\mathbf{dst}(q_i[1])}$, satisfy both the routing and ranking properties.

However, the potential fields do not take SU packet queues into account. Since packets can only be transmitted when they are at the head of a queue, we want to avoid transmitting packets to receivers with long queues. Additionally, we want to ensure no packet is experiencing an unreasonably large delay, and would therefore like to prioritize packets that have already delayed a lot. With these observations in mind, we define a *priority* Q_{ij} associated with the transmitter and receiver of each link according to:

$$Q_{ij} = \begin{cases} \left(\frac{\sum_{n=1}^{|q_i|} \mathbf{dly}(q_i[n])}{1+|q_j|} \right)^{\lambda}, & \text{if } j \neq \mathbf{dst}(q_i[1]), \\ \left(\sum_{n=1}^{|q_i|} \mathbf{dly}(q_i[n]) \right)^{\lambda}, & \text{otherwise,} \end{cases} \tag{27}$$

where $\mathbf{dly}(q_i[n])$ is the delay (number of time slots) experienced by packet $q_i[n]$ so far, and $\lambda \geq 0$ a user defined constant. The numerator increases the priority when the packets at the transmitter have already experienced large delays, while the denominator decreases the priority the longer the packet queue at the receiver is (unless the receiver is the destination, in which case this queue is ignored).

The weight function $\mathtt{w} : \mathcal{N} \times \mathcal{N} \times \mathcal{M} \mapsto \mathbb{R}$ is now defined as

$$\mathtt{w}(i,j,m) = \begin{cases} Q_{ij} \frac{\Phi_{j,\mathbf{dst}(q_i[1])} - \Phi_{i,\mathbf{dst}(q_i[1])}}{\Phi_{i,\mathbf{dst}(q_i[1])}}, & \text{if } |q_i| > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{28}$$

The denominator $\Phi_{i,\mathbf{dst}(q_i[1])}$ ensures that packets do not get stuck near the destination where the absolute difference $|\Phi_{\mathbf{dst}(q_i[1]),\mathbf{dst}(q_i[1])} - \Phi_{i,\mathbf{dst}(q_i[1])}| = |0 - \Phi_{i,\mathbf{dst}(q_i[1])}| = \Phi_{i,\mathbf{dst}(q_i[1])}$ may be low. Note that this weight function does not depend on the frequency band m of the link.

4.2 Solving the Short-Term Problem

In order to solve the MINLP defined by (11)-(17), one must decide transmitter, receiver, frequency band and transmit power of each transmission that is to take place. Instead of trying to decide all these parameters at the same time, we will design a greedy algorithm that iteratively updates a partial solution by fixing an increasing subset of the parameters. In particular, the algorithm will greedily schedule one new transmission at a time in order of increasing weight. After each scheduled transmission, the remaining transmissions that have not yet been scheduled must be updated to take the new transmission into account. The algorithm terminates when no more transmissions can be found that have a negative weight or scheduled without violating one of the constraints.

4.2.1 Partial Solution

Recall that our goal is to find a solution $(\mathbf{a}, \mathbf{P})$ to the MINLP defined by (11)-(17). We will attempt to find this solution in a greedy manner, by constructing a *transmission set* $\mathcal{S} \subseteq \mathcal{N}$, and a corresponding *partial solution* $(\mathbf{a}_\mathcal{S}, \mathbf{P}_\mathcal{S})$ one transmission (transmitter-receiver-frequency-power tuple) at a time. In particular, for each transmitter or receiver $j \in \mathcal{S}$, the values of parameters a_{ij}^m, a_{jk}^m and P_j^m in the partial solution $(\mathbf{a}_\mathcal{S}, \mathbf{P}_\mathcal{S})$ have been fixed for all $i, k \in \mathcal{N}$ and $m \in \mathcal{M}$.

The transmission set $\mathcal{S}$ consists of the SUs $i \in \mathcal{N}$ that have been scheduled to either transmit or receive. It is initially empty and then updated every time a new scheduled transmitter-receiver pair is decided. Given transmission set $\mathcal{S}$, the next transmitter to be scheduled is chosen from the set of *candidate transmitters*

$$\overline{\mathcal{N}} = \{i \in \mathcal{N} \mid i \notin \mathcal{S}, |q_i| > 0\}. \tag{29}$$

That is, $\overline{\mathcal{N}}$ is the set of SUs not yet scheduled for either transmission or reception and that have nonempty packet queues. Its receiver, meanwhile, is chosen from the set of *candidate receivers*

$$\underline{\mathcal{N}} = \mathcal{N} \setminus \mathcal{S}, \tag{30}$$

which consists of all SUs not yet added to $\mathcal{S}$. In each iteration, the scheduler will pick a link $(i, j, m) \in \overline{\mathcal{N}} \times \underline{\mathcal{N}} \times \mathcal{M}$ and schedule it to transmit ($a_{ij}^m = 1$) with power $P_i^m > 0$, after which the transmission set $\mathcal{S}$ is updated to include i and j. In the upcoming sections, we will explain how the link (i, j, m) and its transmit power P_i^m are chosen.

For the initial empty transmission set $\mathcal{S} = \emptyset$, we have $(\mathbf{a}_\mathcal{S}, \mathbf{P}_\mathcal{S}) = (\mathbf{0}, \mathbf{0})$. That is, initially all optimization variables are set to 0. It is easy to verify that this constitutes a feasible solution. After each update made to $(\mathbf{a}_\mathcal{S}, \mathbf{P}_\mathcal{S})$, we want to maintain this feasibility. Therefore, if we set $a_{jk}^m = 1$ and $P_j^m > 0$, then by the half-duplex constraint (16) we must have

$$a_{jk}^{m'} = 0 \qquad \forall\ m' \in \mathcal{M} \setminus \{m\} \tag{31}$$
$$a_{ij}^m = 0 \qquad \forall\ i \in \mathcal{N}, m \in \mathcal{M} \tag{32}$$
$$a_{jk'}^m = 0 \qquad \forall\ k' \in \mathcal{N} \setminus \{k\}, m \in \mathcal{M} \tag{33}$$
$$a_{j'k}^m = 0 \qquad \forall\ j' \in \mathcal{N} \setminus \{j\}, m \in \mathcal{M} \tag{34}$$
$$a_{kl}^m = 0 \qquad \forall\ l \in \mathcal{N}, m \in \mathcal{M} \tag{35}$$

Note that we already do this implicitly by adding j and k to $\mathcal{S}$, since the variables associated with them will then no longer be modified.

4.2.2 Power Control: Upper Bound

Given a partial solution $(\mathbf{a}_\mathcal{S}, \mathbf{P}_\mathcal{S})$, assume we want to add $i \in \overline{\mathcal{N}}$ as the next transmitter. Because the transmission of i would add interference at unintended receivers (either SUs that have been scheduled to receive or the PU receivers), we must ensure this interference does not lower any of the unintended receivers' SINR

below the threshold Γ. In particular, we must ensure that constraints (12) and (13) remain satisfied if we set $a_{ij}^m = 1$ and $P_i^m > 0$ for some $j \in \mathcal{N}$ and $m \in \mathcal{M}$. Since all other parameters involved are fixed (or 0), it is possible to find an upper bound $P_{i,\mathrm{U}}^m$ on the transmission power P_i^m that guarantees the constraints remain satisfied. In the following, we will discuss how to determine this upper bound for a given $i \in \overline{\mathcal{N}}$ and $m \in \mathcal{M}$.

First, note that according to constraint (14) we must have

$$P_{i,\mathrm{U}}^m \leq P_{\max}. \tag{36}$$

Now let $u \in \mathcal{S}$ be a transmitter that has been scheduled to transmit in band m and k its receiver, i.e. $a_{uk}^m = 1$. Then we can see that the corresponding constraint (12) is guaranteed to remain satisfied if i were to transmit, when the upper bound $P_{i,\mathrm{U}}^m$ satisfies

$$P_{i,\mathrm{U}}^m \leq \frac{\frac{g_{uk}P_u^m}{\Gamma} - \sigma_{\mathrm{SU}} - b^m g_{\mathtt{TX},k}^m P_{\mathtt{TX}}^m - \sum_{j=1}^{N} \sum_{\substack{l=1 \\ l\neq k}}^{N} a_{jl}^m g_{jk} P_j^m}{g_{ik}}. \tag{37}$$

Here we have simply rearranged the terms of the constraint (12) associated with transmission over link (u, k, m) and replaced the transmit power P_i^m by its upper bound $P_{i,\mathrm{U}}^m$. Repeating this for all $u \in \mathcal{S}$ that have been scheduled to transmit in band m, we then know $P_i^m \leq P_{i,\mathrm{U}}^m$ satisfies all constraints (12). Naturally, if there are no $u \in \mathcal{S}$ scheduled to transmit in band m, this step can be skipped.

For constraint (13), we assume $b^m = 1$. If this is not the case, there is no PU transmission in band m that i may interfere with, and hence no need to further modify $P_{i,\mathrm{U}}^m$. Since we do not know the location of the PU receiver, the path losses $g_{i,\mathtt{RX}}^m$ and $g_{\mathtt{TX},\mathtt{RX}}^m$ are unknown to us. Thus, if we want to guarantee that (13) remains satisfied, our only option is to take a conservative approach and assume the worst case. Depending on the distance $d_{\mathtt{TX},i}^m$, there are two possible worst cases to consider. In particular, note that since the PU transmitters have a fixed transmit power $P_{\mathtt{TX}}^m$, they have a corresponding *coverage distance* d_{C}^m inside of which the receiver of their transmission must be for the transmission to be successful. Specifically, d_{C}^m is the distance from the PU transmitter in band m at which its transmissions have an SINR equal to Γ when there is no interference:

$$\frac{(1 + d_{\mathrm{C}}^m)^{-\alpha} P_{\mathtt{TX}}^m}{\sigma_{\mathrm{PU}}} = \Gamma \quad \Longleftrightarrow \quad d_{\mathrm{C}}^m = \left(\frac{P_{\mathtt{TX}}^m}{\sigma_{\mathrm{PU}}\Gamma}\right)^{1/\alpha} - 1. \tag{38}$$

If SU i is within the coverage region, i.e. $d_{\mathtt{TX},i}^m \leq d_{\mathrm{C}}^m$, the PU receiver can be arbitrarily close to i. As it is unlikely that i can transmit in such a case without lowering the PU receiver's SINR below the threshold, we will simply set $P_{i,\mathrm{U}}^m = 0$, effectively forbidding i from transmitting in band m.

If, on the other hand, i is outside the coverage region of the PU transmitter, the worst case is for the PU receiver to be on the border of the coverage region and as close to i as possible. This means that the receiver would be located where the border of the coverage region and a line between the PU transmitter and i intersect. However, at the border of the coverage region the SINR is equal to the threshold.

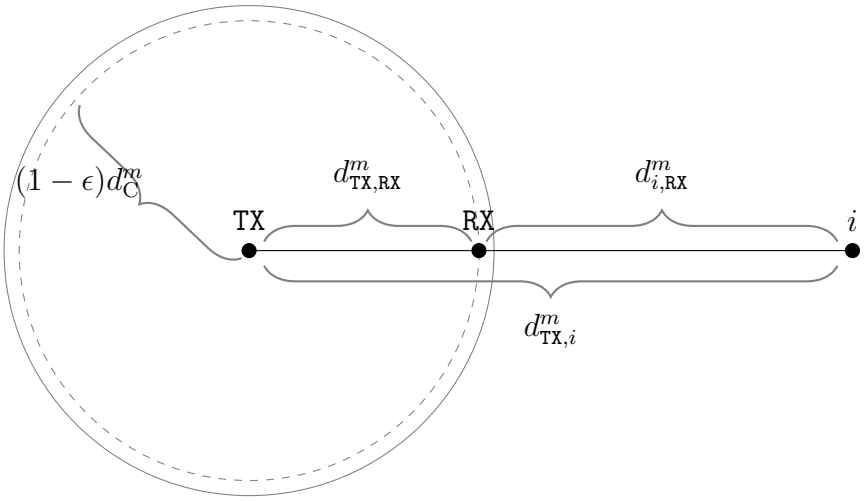


Figure 9: Relaxed worst case scenario for the PU receiver location from the point of view of an SU i that is outside the coverage region (circled area) of the PU transmitter.

Thus, *any* amount of interference, no matter how small, would reduce the SINR of the receiver below the threshold in this theoretical worst case scenario. We would therefore have $P^m_{i,\mathrm{U}} = 0$ in this case as well. This is not very practical, as i could be arbitrarily far away from the PU receiver and still not be allowed to transmit in band m if the PU transmitter is transmitting. For this reason, we will relax the worst case scenario somewhat, and assume that the PU receiver is at distance at most $(1-\epsilon)d^m_{\mathrm{C}}$ from the PU transmitter, where $0 \leq \epsilon < 1$. This relaxed worst case is depicted in Fig. 9.

Now let $\langle x \rangle_i$ denote the worst case estimation of parameter x from the point of view of SU i. As can be seen in Fig. 9, we have

$$\langle d^m_{i,\mathtt{RX}} \rangle_i = d^m_{\mathtt{TX},i} - (1-\epsilon)d^m_{\mathrm{C}}, \tag{39}$$

$$\langle d^m_{\mathtt{TX},\mathtt{RX}} \rangle_i = (1-\epsilon)d^m_{\mathrm{C}}, \tag{40}$$

which further gives us

$$\langle g^m_{i,\mathtt{RX}} \rangle_i = (1 + \langle d^m_{i,\mathtt{RX}} \rangle_i)^{-\alpha}, \tag{41}$$

$$\langle g^m_{\mathtt{TX},\mathtt{RX}} \rangle_i = (1 + \langle d^m_{\mathtt{TX},\mathtt{RX}} \rangle_i)^{-\alpha}. \tag{42}$$

Rearranging the terms of constraint (13) and replacing the unknown path losses by their worst case estimates as defined in (41) and (41), as well as the transmit power P^m_i by its upper bound $P^m_{i,\mathrm{U}}$, we see that the upper bound must satisfy

$$P^m_{i,\mathrm{U}} \leq \frac{\frac{\langle g^m_{\mathtt{TX},\mathtt{RX}} \rangle_i P^m_{\mathtt{TX}}}{\Gamma} - \sigma_{\mathrm{PU}} - \sum_{j=1}^{N} \sum_{l=1}^{N} a^m_{jl} \langle g^m_{j,\mathtt{RX}} \rangle_j P^m_j}{\langle g^m_{i,\mathtt{RX}} \rangle_i}. \tag{43}$$

Note that if $\epsilon > 0$ and the PU receiver is in fact at distance $d^m_{\mathtt{TX},\mathtt{RX}} > (1-\epsilon)d^m_{\mathrm{C}}$ from the transmitter, this upper bound no longer guarantees that constraint (13) remains

satisfied. However, as our simulation examples will demonstrate this tradeoff may well be worth it if violation of constraint (13) is not strictly prohibited.

4.2.3 Candidate Transmissions

Based on the computed upper bounds $P^m_{i,\mathrm{U}}$, we get a *candidate neighborhood* $\overline{\mathcal{N}}^m_i$ for each $i \in \overline{\mathcal{N}}$ and $m \in \mathcal{M}$ according to

$$\overline{\mathcal{N}}^m_i = \{k \in \mathcal{N}^m_i \mid P^m_i = P^m_{i,\mathrm{U}}\} \cap \underline{\mathcal{N}}. \tag{44}$$

These neighborhoods determine all the possible links $(i, j, m) \in \overline{\mathcal{N}} \times \underline{\mathcal{N}} \times \mathcal{M}$ that can be scheduled for transmission next without violating any constraints. The weight of link (i, j, m) is

$$w^m_{ij} = \mathtt{w}(i, j, m) \tag{45}$$

where $\mathtt{w}$ is the weight function provided by the long-term scheduler. Since the objective function (11) can only decrease by scheduling links (i, j, m) with negative weights $w^m_{ij} < 0$, we can disregard links (i, j, m) with $w^m_{ij} \geq 0$. Thus, we get the set of *candidate transmissions* $\mathcal{C}$ from which to choose the next transmission as

$$\mathcal{C} = \{(i, j, m) \mid i \in \overline{\mathcal{N}},\ j \in \overline{\mathcal{N}}^m_i \text{ and } w^m_{ij} < 0\}. \tag{46}$$

Although more sophisticated heuristics could be used to decide which candidate transmission $c \in \mathcal{C}$ to choose for transmission next, our algorithm greedily chooses the candidate transmission (i, j, m) with the lowest weight w^m_{ij}.

4.2.4 Power Control: Lower Bound and Adjusted Power

Let $(i, j, m) \in \mathcal{C}$ be the candidate transmission with the lowest weight. Then we will have $a^m_{ij} = 1$. What remains is to decide the transmit power P^m_i. From before, we know that it must satisfy $P^m_i \leq P^m_{i,\mathrm{U}}$. However, since the SINR at the receiver j must be high enough, there is also a lower bound $P^m_{i,\mathrm{L}}$ on the transmit power, i.e. $P^m_i \geq P^m_{i,\mathrm{L}}$. In particular, $P^m_{i,\mathrm{L}}$ is the transmit power with which the SINR at the receiver j is equal to the threshold Γ. Rearranging the terms in constraint (12), we see that this lower bound is

$$P^m_{i,\mathrm{L}} = \frac{\Gamma\left(\sigma_{\mathrm{SU}} + b^m g^m_{\mathtt{TX},k} P^m_{\mathtt{TX}} + \sum_{j=1}^{N} \sum_{\substack{l=1 \\ \neq k}}^{N} a_{jl} g_{jk} P^m_j\right)}{g_{ik}}. \tag{47}$$

Due to the way the link (i, j, m) was chosen, we know that $P^m_{i,\mathrm{L}} \leq P^m_{i,\mathrm{U}}$. Thus, we have $P^m_{i,\mathrm{L}} \leq P^m_i \leq P^m_{i,\mathrm{U}}$. This can be expressed via parameter $\tau \in [0, 1]$ as

$$P^m_i = (1 - \tau) P^m_{i,\mathrm{L}} + \tau P^m_{i,\mathrm{U}}. \tag{48}$$

We could try to determine an individual τ for each scheduled transmission that allows us to further optimize the solution. However, we will use a much simpler approach with a constant predetermined τ for all transmissions.

Algorithm 1 Centralized Scheduling Algorithm

procedure GREEDYSCHEDULING
  $\mathbf{a} \leftarrow \mathbf{0} \in \{0,1\}^{|\mathcal{N}|\times|\mathcal{N}|\times|\mathcal{M}|}$
  $\mathbf{P} \leftarrow \mathbf{0} \in \mathbb{R}^{|\mathcal{N}|\times|\mathcal{M}|}$
  $\overline{\mathcal{N}} \leftarrow \emptyset$
  for $i \in \mathcal{N}$ **do**
    if $|q_i| > 0$ **then**
      $\overline{\mathcal{N}} \leftarrow \overline{\mathcal{N}} \cup \{i\}$
  $\underline{\mathcal{N}} \leftarrow \mathcal{N}$
  while $\overline{\mathcal{N}} \neq \emptyset$ **do**
    $\mathcal{C} \leftarrow \emptyset$
    for $i \in \overline{\mathcal{N}}$ **do**
      for $m \in \mathcal{M}$ **do**
        compute $P^m_{i,\mathrm{U}}$ according to (36)-(43)
        compute $\overline{\mathcal{N}}^m_i$ according to (44)
        for $j \in \overline{\mathcal{N}}^m_i$ **do**
          compute w^m_{ij} according to (45)
          if $w^m_{ij} < 0$ **then**
            $\mathcal{C} \leftarrow \mathcal{C} \cup \{(i,j,m)\}$
    if $\mathcal{C} = \emptyset$ **then**
      exit **while** loop
    $(i,j,m) \leftarrow \arg\min_{(i,j,m)\in\mathcal{C}} w^m_{ij}$
    $a^m_{ij} \leftarrow 1$
    compute $P^m_{i,\mathrm{L}}$ according to (47)
    $P^m_i \leftarrow (1-\tau)P^m_{i,\mathrm{L}} + \tau P^m_{i,\mathrm{U}}$
    $\overline{\mathcal{N}} \leftarrow \overline{\mathcal{N}} \setminus \{i,j\}$
    $\underline{\mathcal{N}} \leftarrow \underline{\mathcal{N}} \setminus \{i,j\}$
  return $(\mathbf{a}, \mathbf{P})$

4.2.5 Scheduling Algorithm

The scheduling algorithm developed in this thesis work, referred to as GREEDYSCHEDULING, is presented as Algorithm 1. In short, the algorithm operates iteratively by picking the current lowest weight transmission link until either there are no more candidate transmitters left unscheduled ($\overline{\mathcal{N}} = \emptyset$), or no more candidate transmissions can be found ($\mathcal{C} = \emptyset$). After each transmission choice, the remaining candidate transmissions have to be updated to reflect the new spectrum conditions induced by the newly added transmission. This can be compared to finding a maximal independent set in a graph, where each vertex added to the independent set prevents other vertices from being added to it.

Although GREEDYSCHEDULING differs somewhat from the three scheduling algorithms presented in [5] – *frequency divided scheduling* (FDS), *frequency shared scheduling* (FSS), and *transmitter-indifferent scheduling* (TIS) – we note that it roughly corresponds to the FSS algorithm when $\tau > 0$ (with equivalence for $\tau = 1$).

Furthermore, when $\tau = 0$, it will make the exact same assignments to the decision variables a_{ij}^m as the FDS algorithm. Because the transmit powers P_i^m are only chosen to make the assignment $\mathbf{a}$ feasible, the resulting end-to-end delays of packets depend entirely on the assignments $\mathbf{a}$, and hence the results given by GreedyScheduling for $\tau = 0$ are the same as when using the FDS algorithm.

The run time of Algorithm 1 is $\mathcal{O}(N^3)$, i.e. polynomial w.r.t. the size N of the network. However, note that Algorithm 1 is intended to be as easily readable as possible, and does not include some obvious optimizations that would make it harder to read. For instance, after setting $a_{ij}^m = 1$, we only need to recompute the upper bounds $P_{j,\mathrm{U}}^m$ for $j \in \overline{\mathcal{N}}$ in the same band m. For all other $m' \neq m$, $P_{j,\mathrm{U}}^m$ for $j \in \overline{\mathcal{N}}$ remain as they were. Nevertheless, the run time of GreedyScheduling remains polynomial w.r.t. the size of the network in either case.

The solution produced by GreedyScheduling is a feasible solution to the short-term problem. That is, to the problem defined by (11)-(17) where the constraint (13) has been relaxed to a soft constraint that does not affect the feasibility of the solution. We will not analyze any possible approximation bounds associated with this solution. Instead, we will measure its performance numerically, the results of which will be demonstrated later, in chapter 6. Note that we have not discussed *how* the information required to run the centralized algorithm is collected, or how its decisions are made known to the SUs. Hence, it is not clear how the algorithm would be implemented in practice in the type of multi-hop CR networks being studied. We address this in the next chapter by instead considering a distributed version of the algorithm.

5 Distributed Scheduling

The scheduling method discussed in the previous chapter requires knowledge of the spectrum and packet queue state at *every* SU. In the multi-hop network considered in this thesis, acquiring such knowledge is not only challenging, but may be infeasible if the network is large enough. Instead of trying to acquire complete *global* information of the network, we might therefore be interested in performing the scheduling with only incomplete *local* information. In this chapter, we will present a distributed version of the previously discussed scheduling method, as proposed in [6]. Here, every SU acts as their own scheduler and must make scheduling decisions based only on information about their local neighborhoods. To make the connection between this distributed scheduling method and its previously discussed centralized counterpart more obvious, we will consider a generalized version of the one proposed in [6], where the scheduling protocol may be repeated $S \geq 1$ times in each time slot ($S = 1$ in [6]).

Because it is not immediately obvious what network parameters are known by which SUs, and consequently what information SUs need to share with their neighbors, we start this chapter by discussing local information and the sharing of it in section 5.1. We then proceed to present the solutions to the long- and short-term problems in sections 5.2 and 5.3, respectively. In this case, the solution to the long-term problem is the same as it was in the centralized scheduling, and we therefore only discuss the long-term problem briefly by highlighting some of the issues introduced by the new ad hoc setting. The solution to the short-term problem is similar to the one in the centralized scheduling, but it is now distributed among the SUs as subproblems that they must solve with the help of their neighbors.

5.1 Local Information and Control Neighborhoods

The scheduling is performed individually by each SU, and we therefore need to consider what an SU knows about the network state at any given time. We will group the information available at an SU into three categories: local information, semi-local information, and global information. Local information is any information that is only known by a local cluster of SUs within the network, and global information known to the entire network, with semi-local being somewhere in between. Because the SUs may communicate and thus propagate any information that they have about the network, we notice that the locality of a certain parameter is directly related to how frequently its value changes. In particular, the more frequently the value of a parameter changes, the less time it has to propagate before the propagated value is outdated, thus making the parameter more local.

We will define the three information categories as follows. Local information consists of parameters whose values may change in every time slot, sometimes even within a time slot. This includes parameters like the decision variables a_{ij}^m and packet queues q_i. Semi-local information, meanwhile, consists of parameters that may change between time slots, but is more likely to change much less frequently than that. That is, if t_c is the expected time between changes in the parameter's value and t_s the duration of a time slot, we have $t_c >> t_s$. This includes parameters like

Table 1: The locality of parameters that determine the network state. The last column lists the approximate time-scale for which a particular value of a parameter is valid, where t_s is the length of a time slot and T the time it takes for a message to propagate throughout the network. Semi-local parameters are needed when solving the long-term problem, and local parameters in the short-term problem. Note that the table only includes a selection of all network parameters and is thus not comprehensive.

parameter	**description**	**locality**	**time-scale**
$\mathbf{x}_{\mathtt{TX}}^m$	coords. of $\mathrm{PU}_{\mathtt{TX}}$ in band m	global	$>> T$
$\mathbf{x}_i$	coords. of SU i	semi-local	$>> t_s$
a_{ij}^m	decision var. of link (i, j, m)	local	$\leq t_s$
b^m	sensed state of PU m	local	t_s
P_i^m	tr. power of SU i in band m	local	$\leq t_s$
$P_{\mathtt{TX}}^m$	tr. power of $\mathrm{PU}_{\mathtt{TX}}$ in band m	global	$>> T$
q_i	packet queue at SU i	local	t_s
z^m	tr. prob. of $\mathrm{PU}_{\mathtt{TX}}$ in band m	global	$>> T$
Φ_{ij}	potential value of i w.r.t. j	semi-local	$>> t_s$
σ_{SU}	noise power at SUs	global	$>> T$
σ_{PU}	noise power at PUs	global	$>> T$
Γ	SINR threshold	global	$>> T$

the coordinates $\mathbf{x}_i$ of SUs. Of course, if SU i is moving, the value $\mathbf{x}_i$ will change on a continuous (sub-t_s) time-scale. However, we assume that the movement in such a case is slow enough that the difference in coordinates between two consecutive time slots can effectively be ignored. Finally, global information includes all parameters that change so infrequently that they can be considered constant from the point of view of the scheduling. By the assumptions made about our network, this includes parameters like the coordinates $\mathbf{x}_{\mathtt{TX}}^m$ of PU transmitters and their transmit probabilities z^m.

We list the most relevant network parameters and which information category they belong to in Table 1. Note that this list does not include parameters that in our model can be determined by other parameters already listed in the table. For example, we do not include the SINR γ_{ij}^m in this table as it can be determined by the parameters $\mathbf{a}$, $\mathbf{P}$, b^m, P_{TX}^m, σ_{SU} and Γ, as well as the coordinates $\mathbf{x}_{\mathtt{TX}}^m$ and $\mathbf{x}_i$ of all SUs $i \in \mathcal{N}$, all of which are listed in the table. In such cases, the locality of the parameter is determined by the most local parameter needed to compute its value. Because we need e.g. the local parameters $\mathbf{a}$ when determining γ_{ij}^m, the computed

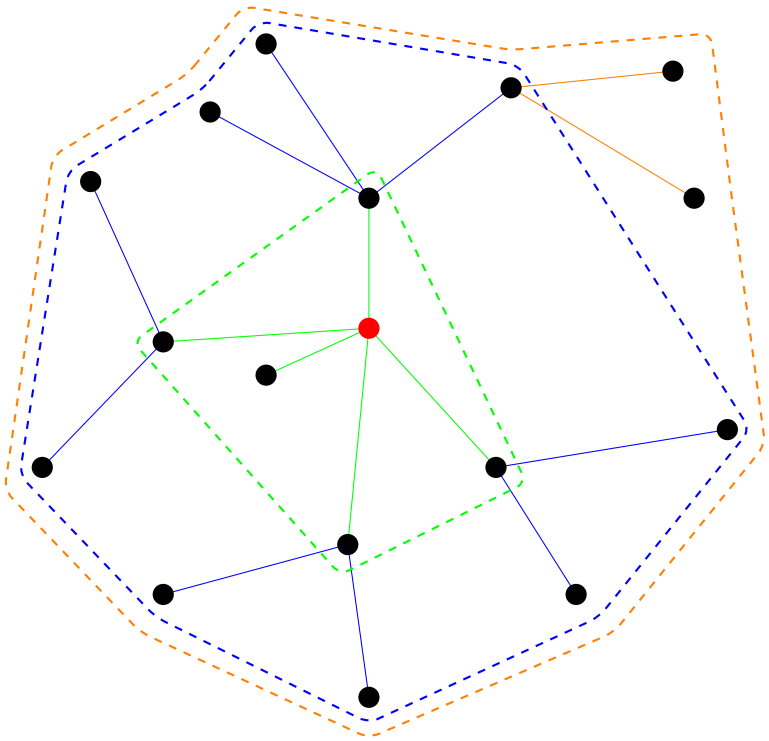


Figure 10: The s-hop control neighborhoods (for $s = 1$, $s = 2$ and $s = 3$) of an SU (marked in red). Green shows the 1-hop control neighbors, blue the 2-hop control neighbors and orange the 3-hop control neighbors.

SINR γ_{ij}^m is local.

As discussed earlier, the exchange of information happens over a CCC. This CCC is outside the licensed spectrum and therefore does not suffer from PU interference. Furthermore, we assume that the control messages sent by the SUs over this channel only add negligible interference that in practice can be ignored. That is, the CCC is an interference-free channel. We denote the neighborhood of SU i in the CCC the *control neighborhood* $\mathcal{N}_i^*$. Note that since there is no interference in the CCC, the control neighborhood contains the transmission neighborhood $\mathcal{N}_i = \bigcup_{m \in \mathcal{M}} \mathcal{N}_i^m$, with equality if there is an idle channel m (i.e. a channel m with no ongoing transmissions, either SU or PU). Furthermore, the links induced by the control neighborhoods are available at all times and can thus be depicted in the form of a connection graph.

During the distributed scheduling process, SUs can only directly communicate with their control neighbors. However, because a control neighbor $j \in \mathcal{N}_i^*$ in turn communicates with its own control neighbors $k \in \mathcal{N}_j^*$, SU i can receive information about SU k two hops away via its one-hop neighbor j. For this reason, we will also talk about s-hop control neighborhoods $\mathcal{N}_i^*(s)$, $s \geq 0$. We have $\mathcal{N}_i^*(0) = \{i\}$ and for $s \geq 1$:

$$\mathcal{N}_i^*(s) = \mathcal{N}_i^*(s-1) \cup \{k \mid k \in \mathcal{N}_j^*(s-1) \text{ for some } j \in \mathcal{N}_i^*\}. \tag{49}$$

An example is shown in Fig. 10. Later on, we will use a parameter S to determine the number of rounds in our scheduling protocol, which will allow SUs to make

decisions based on information about their S-hop control neighborhoods $\mathcal{N}_i^*(S)$.

5.2 Solving the Long-Term Problem

For the long-term problem, we will define the weight function using the same potential field model as in the centralized case. However, as the potential field value Φ_{ij} depends on the distance d_{ij} defined by the semi-local parameters $\mathbf{x}_i$ and $\mathbf{x}_j$, we must device a strategy that allows SUs to accurately estimate the correct value. If the SUs are stationary, the parameters $\mathbf{x}_i$ and $\mathbf{x}_j$ become global and the problem is solved. Therefore, in the discussion below, we consider the case when SUs are mobile. Note that this discussion is only intended to highlight some of the challenges associated with computing the potential fields in an ad hoc setting, and hence is kept relatively high-level and not taken into consideration in the simulations later. In the simulations, semi-local parameters will be treated as global.

Each SU will keep a record of the last known coordinates of other SUs in the network. When an SU moves (more than some threshold distance), it will broadcast its updated coordinates to its control neighbors. The control neighbors in turn forward the updated coordinates to their own control neighbors, and so on, propagating the information throughout the network. Due to the way the information is propagated, we note that the further the distance (number of hops) between two nodes in the network, the less likely it is that they will have up-to-date coordinates of each other.

We do not need to determine the potential values Φ_{ij} for every $i, j, \in \mathcal{N}$. Indeed, since we only need to know weights of outgoing links from SUs i with nonempty packet queues, it suffices to determine the potentials of such SUs as well as their transmission neighbors $j \in \mathcal{N}_i$. Specifically, if $|q_i| > 0$, we only need to know the potential values $\Phi_{i,\mathbf{dst}(q_i[1])}$ and $\Phi_{j,\mathbf{dst}(q_i[1])}$ for each $j \in \mathcal{N}_i$. In the case of GPF and VFF, the semi-local parameters needed to determine these potentials are the coordinates $\mathbf{x}_i$ of the SU i, the coordinates $\mathbf{x}_j$ of its transmission neighbors $j \in \mathcal{N}_i$ as well as the coordinates $\mathbf{x}_{\mathbf{dst}(q_i[1])}$ of the destination. The first is known to the SU and the second can be found within the 1-hop control neighborhood $\mathcal{N}_i^*(1)$ and is therefore easy to get up-to-date values of. This is not necessarily the case for the coordinates $\mathbf{x}_{\mathbf{dst}(q_i[1])}$ of the destination which may be any number of hops away from i. However, the further away the destination is from the transmitter, the less its movements will impact the distances needed to compute the potential values. Therefore, unless the SUs are highly mobile, the potential values when using GPF and VFF are likely good estimates of the true values.

When using the SSDOPF, however, the coordinates of neighbors and destination are not enough to compute the needed potentials at transmitters. In particular, recall from eqs. (24) and (25) that we need to know the potential values of all transmission neighbors $\mathcal{N}_i$ when computing the potential value of SU i. This recursive definition of the potential field means that computing a single value of the potential field requires cooperation between all the SUs in the network. One possible strategy for computing the SSDOPF potentials is then the following. In addition to storing the last known coordinates of other SUs, an SU i will now also keep track of the potentials Φ_{jk} for all $j, k \in \mathcal{N}$. Every time i finds a neighbor $j \in \mathcal{N}_i$ that would reduce its potential

Φ_{ik} for some $k \in \mathcal{N}$, i updates Φ_{ik} to reflect this and then broadcasts the updated Φ_{ik} to its control neighbors. This in turn may prompt one or more of the control neighbors $j \in \mathcal{N}_i^*$ to update their potential Φ_{jk} which is then further broadcast to the control neighbors of j. Alternatively, SU i can use its own local estimates of each $\mathbf{x}_j$ to compute the entire SSDOPF potential field by itself according to the centralized algorithm described in section 4.1.1.

Unlike GPF and VFF, the SSDOPF is very sensitive to any changes in the network topology. For instance, if SU j moves, it suffices to update the potentials Φ_{ij} and Φ_{jk} for all $i, k \in \mathcal{N}$ in GPF and VFF. In the SSDOPF, however, we may have to recompute *all* potential values Φ_{ik} for $i, k \in \mathcal{N}$. That is, in the former case only $2N$ potential values are impacted by the change, while in the latter up to N^2 potential values may be affected. Therefore, using the SSDOPF to compute link weights only makes sense if the network is relatively static. If t_c is the mean time between changes in the network topology and T the time it takes for messages to travel from one end of the network to the other, we should have $t_c >> T$ for the use of SSDOPF to be a feasible choice.

5.3 Solving the Short-Term Problem

For the short-term problem, it is no longer possible to fully coordinate the assignments of a_{ij}^m and P_i^m due to the distributed nature of the scheduling. To account for this, we will start by reformulating the problem to allow for conflicts in the scheduling. We will then present a solution where SUs cooperate with their control neighbors to assign values to subsets of the optimization variables a_{ij}^m and P_i^m.

5.3.1 Reformulation of the Problem

Although violating constraint (12) in the MINLP defined by (11)-(17) for one link (i, j, m) makes the corresponding solution infeasible, it does not invalidate it from a physical point of view. Indeed, the physical interpretation in such a case is simply that the transmission on link (i, j, m) fails, which does not affect the success of other transmissions. Naturally, the assignment a_{ij}^m is then wasted and we add unnecessary interference to other transmissions, but as long as constraints (12) for these transmissions are still satisfied, our solution remains physically valid. For this reason, we will relax the problem. First, we introduce new variables $\tilde{a}_{ij}^m \in \{0, 1\}$ for all $(i, j, m) \in \mathcal{N} \times \mathcal{N} \times \mathcal{M}$, that indicate whether transmission over link (i, j, m) is successful ($\tilde{a}_{ij}^m = 1$) given the current assignment of $\mathbf{a}$ and $\mathbf{P}$. Then, the problem

defined by (11)-(17) is reformulated as

$$\min_{\mathbf{a},\tilde{\mathbf{a}},\mathbf{P}} \quad \sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij}^m \tilde{a}_{ij}^m w_{ij}^m \tag{50}$$

$$\text{s.t.} \quad (13)-(17),$$

$$\gamma_{ij}^m \geq a_{ij}^m \tilde{a}_{ij}^m \Gamma, \quad \forall\, i \in \mathcal{N}, j \in \mathcal{N}, m \in \mathcal{M} \tag{51}$$

$$\tilde{a}_{ij}^m \leq a_{ij}^m, \quad \forall\, i \in \mathcal{N}, j \in \mathcal{N}, m \in \mathcal{M} \tag{52}$$

$$\text{and} \quad \tilde{a}_{ij}^m \in \{0,1\}, \quad \forall\, i \in \mathcal{N}, j \in \mathcal{N}, m \in \mathcal{M}, \tag{53}$$

where as before we abbreviate the left-hand side of constraint (51) (which contains e.g. the optimization variables P_i^m) using Eq. (9). The problem remains a MINLP, and hence is still **NP**-hard. The purpose of the relaxation is simply to increase the feasibility region to allow for conflicting assignments as discussed above.

Note that, although the new parameters $\tilde{a}_{ij}^m$ are listed as optimization variables, their values are directly determined by $\mathbf{a}$ and $\mathbf{P}$. In particular, given a certain $\mathbf{a}$ and $\mathbf{P}$, the value of the objective function (50) cannot be improved compared to the corresponding value in the old objective function (11), regardless of how we choose $\tilde{\mathbf{a}}$. Hence, an optimal solution in the new problem is also an optimal solution in the old problem, and the role of the new variables $\tilde{a}_{ij}^m$ is exclusively to ensure feasibility: when a constraint (12) in the old problem is violated by our solution, we set the corresponding $\tilde{a}_{ij}^m$ to 0, guaranteeing that the new constraint (51) remains satisfied. In practice, we will not be assigning any values to $\tilde{\mathbf{a}}$, and instead simply assume that they are given the appropriate values that retain the feasibility of our solution $(\mathbf{a}, \mathbf{P})$.

The constraint (52) is not strictly speaking necessary as the case $\tilde{a}_{ij}^m > a_{ij}^m$ does not affect the value of the objective function or the feasibility of the solution. However, we include it to emphasize that $\tilde{a}_{ij}^m$ represents whether or not transmission over link (i, j, m) is successful, where $\tilde{a}_{ij}^m = 1$ (the transmission is successful) implies that the link has been scheduled for transmission and therefore $a_{ij}^m = 1$.

The end result of the relaxation is that violation of constraint (12) in the original problem is transformed into a penalty in the objective function. That is, we allow for assignments of $\mathbf{a}$ and $\mathbf{P}$ that lead to conflicts between the SUs, but would like to avoid these in order to minimize the objective. Because the problem remains **NP**-hard, we will still not require optimality of our solution.

5.3.2 Distributed Assignment of Optimization Variables

In the distributed scheduling, every SU $i \in \mathcal{N}$ is responsible for assigning the values of a subset of rows in $\mathbf{a}$ and $\mathbf{P}$. Specifically, the variables a_{ij}^m and P_i^m for all $j \in \mathcal{N}$ and $m \in \mathcal{M}$ are assigned by SU i. In the approach proposed in [6], SUs make these assignment through collaboration with their 1-hop control neighbors. However, in this thesis we will consider a more general case where SUs can coordinate their decisions with their S-hop control neighborhood for any $S \geq 1$.

In particular, S is a predetermined constant that defines how many times we repeat a slightly modified version of the distributed scheduling protocol defined in [6], to be presented in detail later. Each repetition, or *round* s as we will henceforth

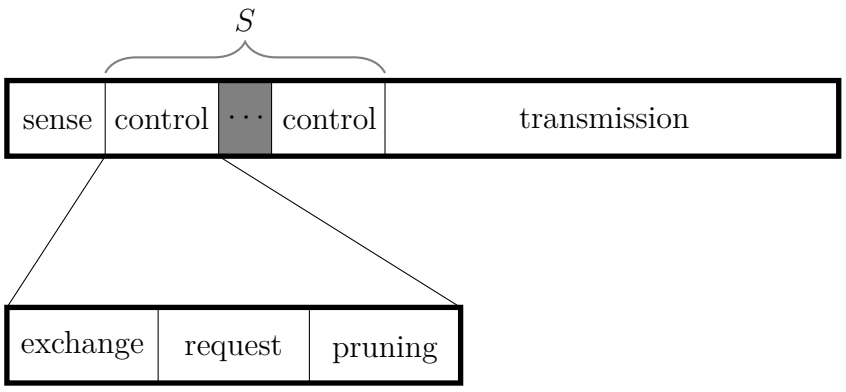


Figure 11: The division of a time slot into sensing, control and transmission minislots. The control minislot is further divided into three phases that are repeated S times, where $S \geq 1$ is a predetermined constant.

refer to them as, of the distributed scheduling protocol roughly corresponds to one iteration of rows 10-26 in Algorithm 1 describing the centralized scheduling method GreedyScheduling. Recall that in the latter, one such iteration consists of the following steps:

1. Given a partial solution $(\mathbf{a}_{\mathcal{S}}, \mathbf{P}_{\mathcal{S}})$, determine the set of candidate transmissions $\mathcal{C} \subseteq \overline{\mathcal{N}} \times \underline{\mathcal{N}} \times \mathcal{M}$ from which to choose the next transmission to add to the solution.

2. Choose the candidate transmission $(i, j, m) \in \mathcal{C}$ with the lowest weight w_{ij}^m, and add it to the partial solution.

One round of the distributed scheduling is functionally similar, but with two main differences. Firstly, because the distributed scheduling is performed by the SUs using local information, the partial solution in step 1 is substituted by a *local* solution $(\mathbf{a}_i, \mathbf{P}_i)$ specific to each SU i. Furthermore, in step 2, it will not be possible to determine the candidate transmission with the globally minimum weight. Instead, only local minima can be found and thus several, possibly conflicting, candidate transmissions may be chosen in each round.

5.3.3 Overview of the Scheduling Protocol

As discussed in chapter 3, time slots are divided into sensing, control and transmission minislots, with the control minislot reserved for the scheduling. In the distributed scheduling protocol developed in this thesis, the control minislot is further divided into $3S$ phases: S *exchange* phases, S *request* phases, and S *pruning* phases. Each exchange phase is followed by a request phase, and the request phase by a pruning phase, forming one round s of the scheduling protocol, for a total of S rounds. This is depicted in Fig. 11. It is important to note that S is a predetermined constant, otherwise we cannot control the length of a time slot.

The basic structure of the scheduling protocol is the following:

1. In the *exchange phase*, SUs exchange local variables with their control neighbors. During the first round, this will only include variables directly associated with the SUs themselves, such as the interferences measured by them in the sensing minislot. However, by having SUs forward variables received from their control neighbors, an SU i will after the s-th exchange phase have access to local variables from its entire s-hop control neighborhood $\mathcal{N}_i^*(s)$.

2. In the *request phase*, SUs i that want to transmit use the information received in previous exchange phases to nondeterministically choose a minimal weight transmission link to transmit over. They then send a transmission request for the chosen link to their control neighbors.

3. Finally, the SUs prune the set of requested transmissions in the *pruning phase* (called the decision phase in [6]), with the goal of only leaving non-conflicting transmission links to be scheduled for transmission.

In the following subsections, we will discuss each phase in more detail. We then conclude the chapter with a more detailed version of the above overview of the protocol.

5.3.4 The Exchange Phase

During the exchange phase, SUs exchange the values of local variables with their control neighbors. The local variables are of three different types: variables whose values are sensed during the sensing minislot, e.g. the PU state variable b^m; variables whose values are known to certain SUs but not to others, e.g. the packet queues q_i and any values associated with them; and variables whose values are determined by the SUs, e.g. the variable a_{ij}^m whose value is determined by SU i. In addition to local variables, SUs may also use this opportunity to share any changes they have discovered in semi-local variables. However, we will not consider how the correctness of semi-local variables affects the results during our simulations. Instead, they are treated as global variables. Similarly, it is possible that the values of sensed local variables change after they have been sensed, but we will not account for this and simply assume that these values remain the same throughout the time slot during which they were sensed.

In our model, the local variable b^m can in principle be sensed by any SU in the network, suggesting that we may not even need to share it with neighbors. This is due to the fact that no SUs are transmitting in the sensing minislot. Hence, any interference sensed in band m is known to come from the PU transmitter in that band, meaning $b^m = 1$. Furthermore, if an SU i knows b^m, then according to Eq. (9) it can determine the received interference-plus-noise power $IN_k^m = I_k^m + N_k^m$ at the time of sensing at any other SU $k \neq i$ in the network using only global and semi-local variables in addition to b^m:

$$IN_k^m = I_k^m + N_k^m = b^m g_{\mathtt{TX},k}^m P_{\mathtt{TX}}^m + \sigma_{\mathrm{SU}}. \tag{54}$$

Here we used the fact that $a_{ij}^m = 0$ for all i, j, m during the sensing whereby Eq. (6) and (7) gives us $I_k^m = b^m g_{\mathtt{TX},k}^m P_{\mathtt{TX}}^m + \sum_{l=1,l\neq k}^N a_{jl}^m g_{jk} P_j^m = b^m g_{\mathtt{TX},k}^m P_{\mathtt{TX}}^m$.

However, this assumes that our interference model is correct and can perfectly estimate any interferences in the network. Because this is unlikely to be the case in practice, SU k will send the *sensed* value of IN_k^m to its control neighbors $\mathcal{N}_k^*$. Note that this value implicitly contains b^m ($IN_k^m > \sigma_{SU} \implies b^m = 1$), allowing SUs that are not able to sense the interference from the PU transmitter to still find out the correct value of b^m if their neighbor can sense it. Also note that because the sensed IN_k^m only contains interference from PUs, SUs must estimate the true interference-plus-noise power (that is, the interference-plus-noise power during the transmission minislot) by manually adding any known SU interference to it. The sensed value IN_k^m will only be sent in the first exchange phase. In subsequent rounds this value is only forwarded on a per-need basis, as will be discussed below.

SUs will also exchange information regarding their packet queues. Because this information is only needed when computing the weights of links, it suffices for SU i to know the value $|q_j|$ for any $j \in \mathcal{N}_i$. Hence, in the first exchange phase, SU i will, in addition to sending the values IN_i^m for all $m \in \mathcal{M}$, also send the value $|q_i|$. There is no need to forward this value in later exchange phases.

For the assigned local variables a_{ij}^m and P_i^m, let $(\mathbf{a}(s), \mathbf{P}(s))$ be the global solution after s rounds. That is, the solution where elements $a_{ij}^m(s)$ and $P_i^m(s)$ have the value assigned to them by SU i in round s. This global solution is not known to SUs. Instead, each SU i will have their own local version, denoted $(\mathbf{a}_i(s), \mathbf{P}_i(s))$. As in the centralized case, we start out with $(\mathbf{a}(0), \mathbf{P}(0)) = (\mathbf{a}_i(0), \mathbf{P}_i(0)) = (\mathbf{0}, \mathbf{0})$. After each pruning phase, SU i will update the rows in $\mathbf{a}_i$ and $\mathbf{P}_i$ that it is responsible for in the global solution. Additionally, in the exchange phase of each round $s > 1$, SU i will send $(\mathbf{a}_i(s-1), \mathbf{P}_i(s-1))$, updated with the assignments it made in the previous round's pruning phase, to its control neighbors $\mathcal{N}_i^*$. Upon receving $(\mathbf{a}_j(s-1), \mathbf{P}_j(s-1))$ from one of its control neighbors $j \in \mathcal{N}_i^*$, SU i will update its own local solution to reflect its most recent knowledge of the elements in the global solution. Specifically, the elements in $\mathbf{a}_i(s)$ will be

$$[\mathbf{a}_i(s)]_{jk}^m = \begin{cases} a_{jk}^m(s-1), & \text{if } j = i, \\ a_{jk}^m(s-t), & \text{if } j \in \mathcal{N}_i^*(s) \text{ and } j \in \mathcal{N}_i^*(t) \setminus \mathcal{N}_i^*(t-1), \\ 0, & \text{if } j \notin \mathcal{N}_i^*(s), \end{cases} \tag{55}$$

and the elements in $\mathbf{P}_i(s)$

$$[\mathbf{P}_i(s)]_j^m = \begin{cases} P_j^m(s-1), & \text{if } j = i, \\ P_j^m(s-t), & \text{if } j \in \mathcal{N}_i^*(s) \text{ and } j \in \mathcal{N}_i^*(t) \setminus \mathcal{N}_i^*(t-1), \\ 0, & \text{if } j \notin \mathcal{N}_i^*(s). \end{cases} \tag{56}$$

That is, after the s-th exchange phase, SUs know the values $a_{jk}^m(s-1)$ and $P_j^m(s-1)$ assigned in the previous round for 1-hop control neighbors and themselves. For t-hop control neighbors, $2 \leq t \leq s$ (that are not also $(t-1)$-hop control neighbors), SUs know the values $a_{jk}^m(s-t)$ and $P_j^m(s-t)$ assigned to the global variables t rounds ago. An example of this can be seen in Fig. 12.

At this point, we should note that sending $(|\mathcal{N}| \times |\mathcal{N}| \times |\mathcal{M}|)$-size vectors over the CCC may not be feasible given our assumption that the messages are short enough

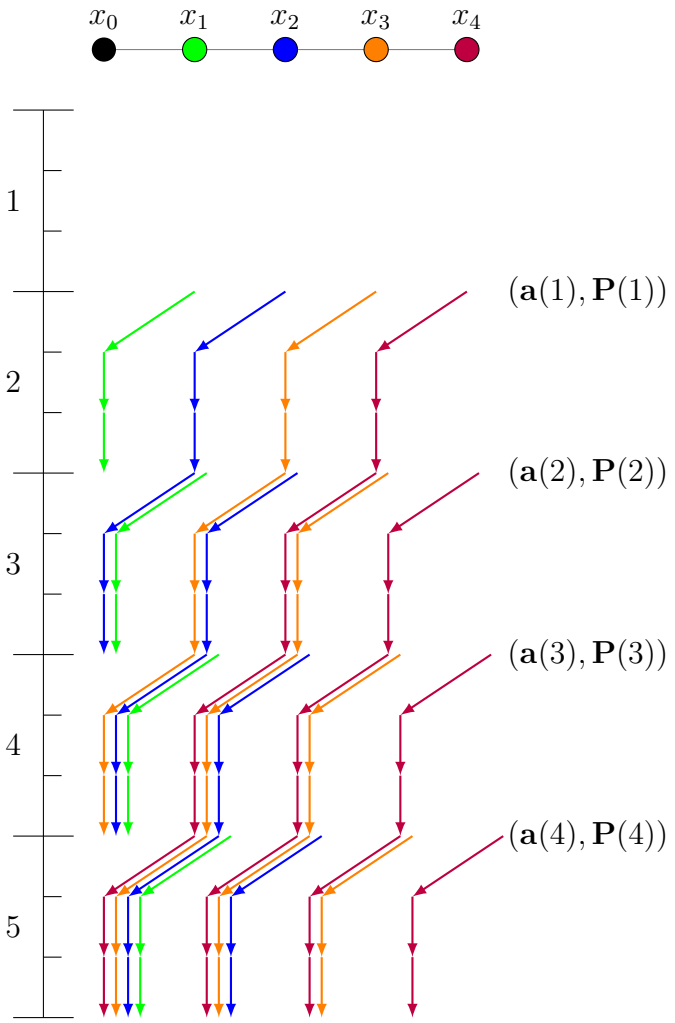


Figure 12: The flow of global optimization variables $(\mathbf{a}, \mathbf{P})$ in a size 5 path network over the course of 5 rounds. Time is depicted along the y-axis with each round indicated by its number on the left. The information flow is shown from the point of view of SU x_0, and information flow that is not relevant to x_0 has been omitted from the picture to make it more readable. In the s-th exchange phase, x_0 receives from its t-hop control neighbor x_t the assignments made by x_t in round $s-t$. For example, in the fifth exchange phase, x_0 receives the assignments made by x_4 in round 1.

not to cause any interference. Instead of sending the entire vectors $\mathbf{a}_i(s)$ and $\mathbf{P}_i(s)$ in every exchange phase, it suffices to send the rows that have changed and the updated values. In order to know which value is the most recent for each row, we also include the round s during which the assignment was made. That is, if SU i decides in the sth pruning phase to set $a_{ij}^m(s) = a \neq a_{ij}^m(s-1)$ and $P_i^m(s) = P \neq P_i^m(s-1)$, it will send the tuple $(i, j, m, a, P, s, IN_j^m)$ to its control neighbors, and the neighbors will in turn update their local solutions accordingly and forward the tuple to their control neighbors in the next round. We include the value IN_j^m in the message since it is known to i after the first exchange phase ($j \in \mathcal{N}_i^m \implies j \in \mathcal{N}_i^*$) and will help

Table 2: The information sent by an SU i to its control neighbors in the sth exchange phase.

	message includes	**note**
$s = 1$	IN_i^m $\lvert q_i \rvert$	for all $m \in \mathcal{M}$
$s > 1$	$(j, k, m, [\mathbf{a}_i(s)]_{jk}^m, [\mathbf{P}_i(s)]_j^m, t, IN_k^m)$	for $(j, k, m) : [\mathbf{a}_i(s)]_{jk}^m \neq [\mathbf{a}_i(s-1)]_{jk}^m$

other SUs better estimate the interferences involved in the assignment made by i.

Table 2 summarizes what information SUs send to their control neighbors in each exchange phase.

5.3.5 The Request Phase

In the request phase, SUs that want to transmit will use the information acquired in previous exchange phases to make requests to their control neighbors for some minimal weight transmission link. From the point of view of the network, this corresponds to SUs suggesting a preliminary assignment of $(\mathbf{a}(s), \mathbf{P}(s))$ in round s, which will then be pruned of any known conflicting assignments in the upcoming pruning phase. In each round s, we define the set of candidate transmitters $\overline{\mathcal{N}}$ and candidate receivers $\underline{\mathcal{N}}$ as before, with the transmission set $\mathcal{S}$ determined by the current assignment $(\mathbf{a}(s-1), \mathbf{P}(s-1))$. That is, $a_{ij}^m(s-1) = 1 \implies \{i, j\} \in \mathcal{S}$. Note that although SUs do not have access to the full transmission set, it will be enough for SU i to know (a) whether it is a candidate transmitter, and (b) which of its control neighbors are candidate receivers, both of which can be determined from the portion of the transmission set known to i.

Now let $i \in \overline{\mathcal{N}}$ be a candidate transmitter. Based on the information received in previous exchange phases, i will determine its candidate neighborhoods $\overline{\mathcal{N}}_i^m$ for each $m \in \mathcal{M}$. In order to do so, it must first compute an upper bound of its transmit power according to the method described in section 4.2.2. Here we note that due to the information available at SU i, Eq. (37) is modified into

$$P_{i,\mathrm{U}}^m \leq \frac{\frac{g_{uk}[\mathbf{P}_i(s)]_u^m}{\Gamma} - IN_k^m - \sum_{j=1}^{N} \sum_{\substack{l=1 \\ l \neq k}}^{N} [\mathbf{a}_i(s)]_{jl}^m g_{jk} [\mathbf{P}_i(s)]_j^m}{g_{ik}}, \tag{57}$$

and Eq. (43) into

$$P_{i,\mathrm{U}}^m \leq \frac{\frac{\langle g_{\mathtt{TX},\mathtt{RX}}^m \rangle_i P_{\mathtt{TX}}^m}{\Gamma} - \sigma_{\mathrm{PU}} - \sum_{j=1}^{N} \sum_{l=1}^{N} [\mathbf{a}_i(s)]_{jl}^m \langle g_{j,\mathtt{RX}}^m \rangle_j [\mathbf{P}_i(s)]_j^m}{\langle g_{i,\mathtt{RX}}^m \rangle_i}. \tag{58}$$

Having computed $P_{i,\mathrm{U}}^m$, the candidate neighborhood $\overline{\mathcal{N}}_i^m$ then consists of all control neighbors $k \in \mathcal{N}_i^*$ that are candidate receivers ($k \in \underline{\mathcal{N}}$) and whose SINR γ_{ik}^m is

estimated to be high enough when setting $P_i^m = P_{i,\mathrm{U}}^m$, i.e.

$$\hat{\gamma}_{ik}^m = \frac{g_{ik} P_{i,\mathrm{U}}^m}{I N_k^m + \sum_{j=1}^{N} \sum_{\substack{l=1 \\ l \neq k}}^{N} [\mathbf{a}_i(s)]_{jl}^m g_{jk} [\mathbf{P}_i(s)]_j^m} \geq \Gamma. \tag{59}$$

Next, i would like to pick a frequency band $m \in \mathcal{M}$ and receiver $j \in \overline{\mathcal{N}}_i^m$ for its transmission. As in the centralized case, it will utilize the weight function to determine the appropriate m and j. However, it will not make this decision deterministically by choosing e.g. the candidate transmission $(i, j, m) \in \mathcal{C}$ with the smallest weight w_{ij}^m. To understand why, recall that the weight function $\mathtt{w} : \mathcal{N} \times \mathcal{N} \times \mathcal{M} \mapsto \mathbb{R}$ that we are using does not depend on the frequency band m and thus we may have several bands $m' \in \mathcal{M}$ that contain the minimum weight transmission (i, j, m'). Furthermore, all candidate transmitters $i' \in \overline{\mathcal{N}}$ will be making these decisions simultaneously and independently of each other. Hence, a deterministic approach may lead to all candidate transmitters choosing their candidate transmissions from only a small subset of the available bands. Not only would this risk overusing the better bands leading to conflicts (the candidate transmissions were chosen based on the current known interference levels and with the assumption that no other new transmissions were added), but we would also end up with some bands not being used at all despite containing spectrum opportunities.

Therefore, i will choose the frequency band m of the transmission in a nondeterministic manner. In particular, let $\mathcal{C}_i^m$ be the set of candidate transmissions whose transmitter is i and frequency band is m, i.e.

$$\mathcal{C}_i^m = \{(i, j, m) \mid j \in \overline{\mathcal{N}}_i^m, w_{ij}^m < 0\} \tag{60}$$

Then i will choose the frequency band m of its transmission randomly according to the distribution $\mu_i : \mathcal{M} \mapsto \mathbb{R}$, where

$$\mu_i(m) = \frac{|\mathcal{C}_i^m|}{\sum_{m'=1}^{M} |\mathcal{C}_i^{m'}|}. \tag{61}$$

This way we still favor bands with good spectrum conditions, while also not completely neglecting less favorable frequency bands.

After the frequency band m has been chosen, i will pick the receiver j corresponding to the candidate transmission $(i, j, m) \in \mathcal{C}_i^m$ with the smallest weight w_{ij}^m, and send a transmission request for this link to all of its control neighbors $k \in \mathcal{N}_i^*$. The transmission request includes the link (i, j, m) as well as the weight w_{ij}^m.

5.3.6 The Pruning Phase

After the request phase, each candidate transmitter will have sent a request for one candidate transmission. Because SUs pick candidate transmissions with themselves as transmitter, this means we will have requests for $|\overline{\mathcal{N}}|$ different candidate transmissions,

each of which may conflict with one another. In the pruning phase, the goal is to prune the set of requested candidate transmissions such that any known conflicts are removed. The term *known* is key here, as there is no guarantee that SUs will be able to solve all conflicts.

In particular, an SU i has only received transmission requests from its 1-hop control neighbors, and hence can only discover conflicts between these. SU i's control neighbors are all within 2 hops of each other. Therefore, we can only guarantee conflict resolution within 2-hop control neighborhoods.

Let $\texttt{req}(i,j,m)$ denote a transmission request for link (i,j,m). We say that requests $\texttt{req}(i,j,m)$ and $\texttt{req}(i',j',m')$ conflict with each other if one (or more) of the following is true:

1. $i = j'$.
2. $j = i'$.
3. $j = j'$.
4. $m = m'$.

The first three conditions are due to the half-duplex nature of SUs. The fourth condition describes any transmissions that would interfere with each other due to using the same frequency band. Note that we could utilize our interference model based on SINR to possibly relax this fourth condition somewhat to only include transmissions in the same band that interfere with each other *too much*. However, as we only discover conflicts within 2-hop neighborhoods during the pruning phase, it is a fair assumption that any discovered conflicts would lead to failed transmissions regardless of the SINRs involved. Furthermore, computing SINRs accurately would require knowledge of the global assignment $(\mathbf{a}(s), \mathbf{P}(s))$.

SUs will use the included weights in the request messages to resolve any discovered conflicts. A transmission request $\texttt{req}(i,j,m)$ is *accepted* by SU $k \in \mathcal{N}_i^*$ if there are no conflicting requests, or w_{ij}^m is the minimum weight among all conflicting requests received by k. Otherwise the request is *rejected*. The request is accepted by sending the constraint $a_{ij}^m \geq 0$ to i, and declined by sending the constraint $a_{ij}^m \leq 0$.

After the pruning phase, each candidate transmitter will have received accept or reject messages from all of their control neighbors, and will make their assignments $a_{ij}^m(s)$ based on these. The candidate transmitters will set $a_{ij}^m(s)$ for their chosen candidate transmission (i,j,m) to the highest possible value. It is easy to see that only requests $\texttt{req}(i,j,m)$ that were accepted by *all* control neighbors lead to the assignment $a_{ij}^m(s) = 1$, and requests with at least one rejection to the assignment $a_{ij}^m(s) = 0$.

If $a_{ik}^m(s) = 1$ for some k, SU i will also need to determine $P_i^m(s)$. This is done using the same power control mechanism as in the centralized scheduling by computing a lower bound $P_{i,\mathrm{L}}^m$ in addition to the earlier upper bound $P_{i,\mathrm{U}}^m$ and then setting P_i^m according to (48) for some fixed τ. Similarly to the upper bound $P_{i,\mathrm{U}}^m$, we note that

Eq. (47) for computing $P_{i,\mathrm{L}}^m$ is now

$$P_{i,\mathrm{L}}^m = \frac{\Gamma\left(IN_k^m + \sum_{j=1}^{N}\sum_{\substack{l=1\\ \neq k}}^{N}[\mathbf{a}_i(s)]_{jl}^m g_{jk}[\mathbf{P}_i(s)]_j^m\right)}{g_{ik}}. \tag{62}$$

5.3.7 Summary of the Scheduling Protocol

The control minislot phases that constitute the distributed scheduling protocol have now been described in detail. Below we summarize the protocol for a better overview:

1. **Exchange phase**. SUs broadcast local variables to their control neighbors, and update their own knowledge based on messages received from other SUs. After s rounds, SU i will have local information from its s-hop control neighborhood $\mathcal{N}_i^*(s)$. The local information received from a t-hop control neighbor, $1 \leq t \leq s$, is from t rounds ago.

2. **Request phase**. Candidate transmitters $i \in \overline{\mathcal{N}}$ compute their transmit power upper bounds $P_{i,\mathrm{U}}^m$ and the corresponding candidate neighborhoods $\overline{\mathcal{N}}_i^m$ based on their current knowledge of the network and scheduling state. For each candidate neighborhood $\overline{\mathcal{N}}_i^m$ they determine the set of candidate transmissions $\mathcal{C}_i^m$. Next, they nondeterministically decide on a frequency band m according to the resulting channel preference distribution μ_i. Finally, they determine the minimum weight candidate transmission $(i,j,m) \in \mathcal{C}_i^m$ in the chosen band m, and broadcast a transmission request $\texttt{req}(i,j,m)$ to their control neighbors. The transmission request includes the chosen candidate transmission (i,j,m) and its weight w_{ij}^m.

3. **Pruning phase**. SUs prune the set of chosen candidate transmissions by either accepting or rejecting any received transmission requests. A request $\texttt{req}(i,j,m)$ is accepted if and only if there are no conflicting requests or w_{ij}^m is the minimum among all weights of conflicting requests. Any candidate transmitters that have had their transmission request $\texttt{req}(i,j,m)$ accepted by *all* their control neighbors set $a_{ij}^m(s) = 1$, and otherwise $a_{ij}^m(s) = 0$. If setting $a_{ij}^m(s) = 1$, they also compute the corresponding transmit power P_i^m.

After the Sth round, SUs will use the transmission minislot to transmit according to the assignment $(\mathbf{a}(S), \mathbf{P}(S))$. Note that only transmissions $(i,j,m,P_i^m) \in \mathcal{N} \times \mathcal{N} \times \mathcal{M} \times \mathbb{R}$ with an SINR γ_{ij}^m above or equal to the threshold Γ are successful. A transmission $(i',j',m',P_{i'}^{m'})$ is guaranteed to not conflict with transmission (i,j,m,P_i^m) if either $i' \in \mathcal{N}_i^*(2)$, or $i' \in \mathcal{N}_i^*(s)$ for some $2 < s \leq S$ and the assignments of these transmissions were made at least s rounds apart in the scheduling.

6 Experimental Results

In this chapter, the performance of the proposed centralized and distributed scheduling methods are studied in simulation. We start off in section 6.1 by taking a closer look at the potential fields. In particular, we inspect the different potential field models visually to get a better understanding of the packet routes defined by them and how the different potential field models differ from each other.

In the second part of this chapter, section 6.2, we test the performances of the proposed scheduling methods in terms of the end-to-end delays of packets achieved by them. In addition to comparing the methods to each other, we also study the impact of different parameter values for each model. This allows us to better understand the roles of different components in the scheduling. For example, we compare the performances of different potential fields in simulations, giving us quantitative results in addition to our earlier analytical ones.

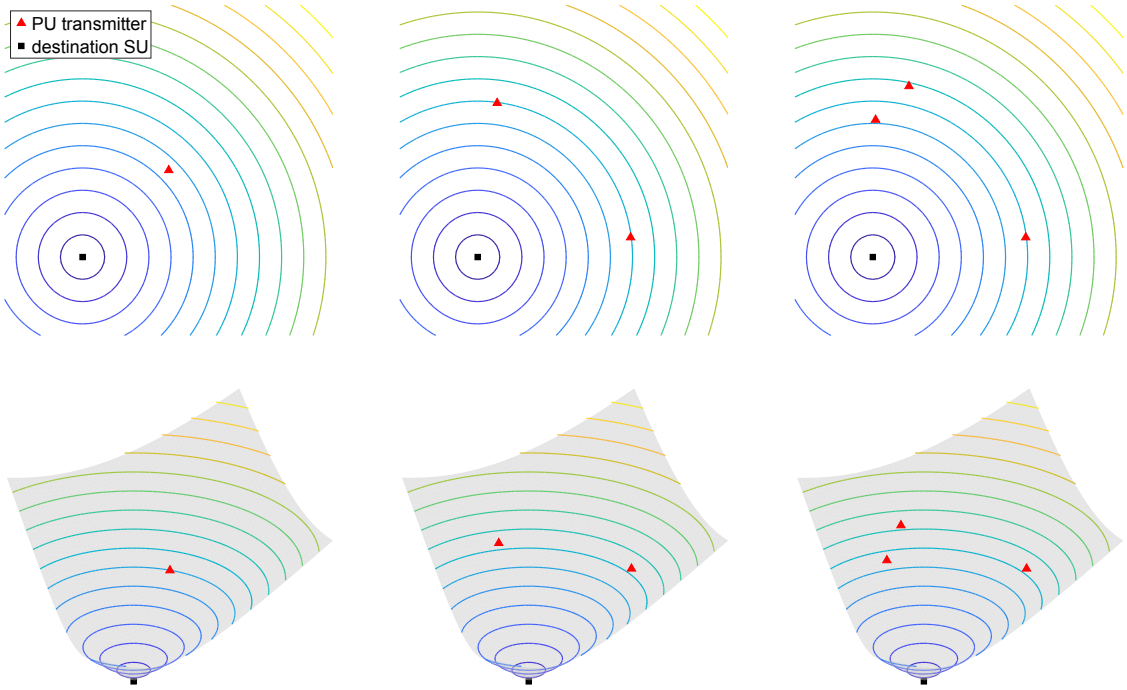
The chapter is concluded in section 6.3 with a summary of the results from both 6.1 and 6.2.

6.1 Visual Analysis of the Potential Fields

In the following, we will inspect the different potential fields (GPF, VFF and SSDOPF) defined by a fixed destination SU i located towards the southwest corner of a square-shaped area. The potential fields are shown in Fig. 13, 14 and 15. For each type of potential field, the number M of PUs, and consequently the number of licensed frequency bands, is altered from 1 to 3, with each M having predetermined positions for the corresponding PU transmitters. The positions of the PU transmitters are chosen such that the results would shed light on different aspects of the potential fields as intuitively as possible. For example, when $M = 3$, two of the PU transmitters are next to each other, with the third more isolated. We would expect packet routes (in the VFF and SSDOPF) to avoid the two co-located PUs more than they avoid the third isolated one. The potential fields are depicted within the square as both two- and three-dimensional elevation maps with yellow/high corresponding to high potential values and blue/low corresponding to low potential values. Recall that the scheduling will send packets from higher to lower potentials, and thus the negative gradient/direction of the slope of the potential field roughly corresponds to the paths along which packets travel. The destination is marked by a black square, and PUs by red triangles. Any adjustable parameter values needed to compute the potential fields are the same as in the network simulations discussed later.

Fig. 13 shows the resulting GPFs. As the potential in this case only depends on the distance to the destination, it remains unaffected when altering M. The negative gradient of the potential field, i.e. the direction of the slope in the picture, points directly towards the destination throughout the potential field.

The VFF is shown in Fig. 14. For $M = 1$, there is a single exponentially growing peak around the one PU transmitter. This results in paths that go around the PU transmitter. Close enough to the PU transmitter the gradient of the potential field may even point directly away from the destination when on the northeast side of the

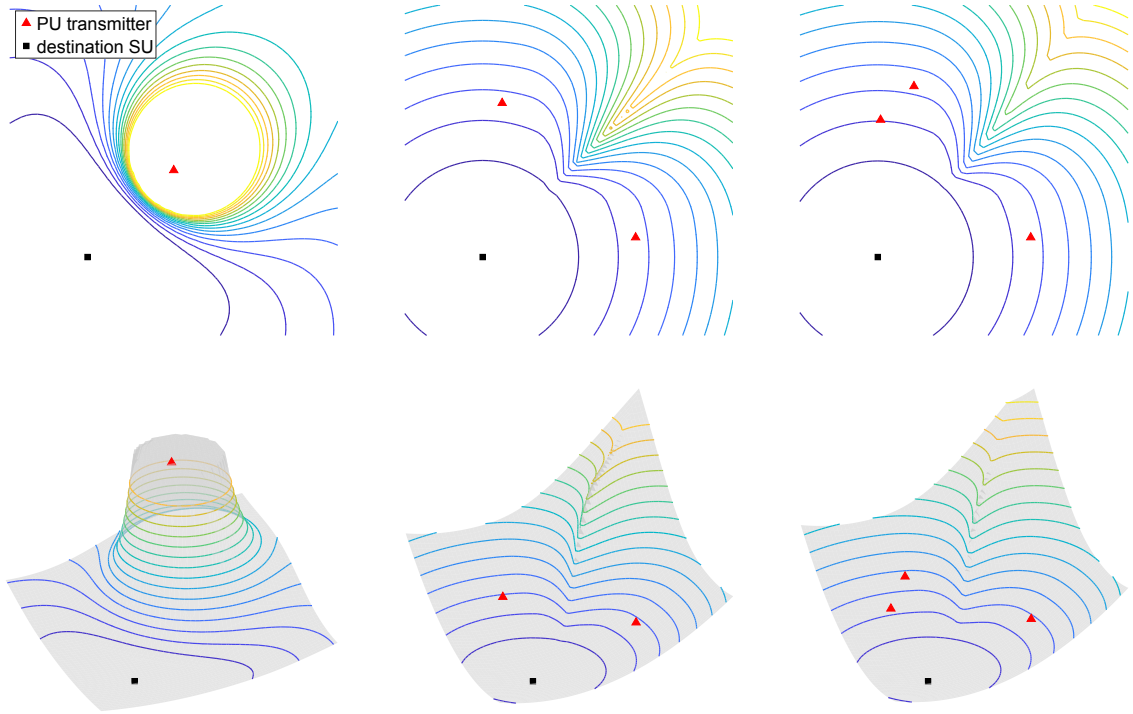


(a) $M = 1$. (b) $M = 2$. (c) $M = 3$.

Figure 13: GPF. The potential depends entirely on the distance to the destination (black square), while ignoring PUs (red triangles) and interferences caused by them.

PU transmitter. Thus, when too close to the PU transmitter, the VFF will prioritize sending packets away from the PU as opposed to getting closer to the destination. For $M = 2$, the peak is replaced by a ridge between the two PU transmitters. As a result, packets will avoid the diagonal connecting the northeast and southwest corners of the area where they would be close to both PU transmitters and instead follow routes that stay closer to one of the two PU transmitters. For $M = 3$, the potential field looks similar to the case $M = 2$. However, since there are now two PU transmitters close to each other on one side, the ridge is moved slightly towards these two PUs as the other side is more likely to have available spectrum.

Finally, Fig. 15 shows the SSDOPF. Because the SSDOPF is only defined at distinct locations, i.e. at the locations of SUs, we can no longer use curves to show different potential levels as was done for the GPF and VFF. Instead, the area depicted is now filled with a grid of $n \times n$ uniformly spaced SUs ($n = 20$ in this case), and the potential computed at each SU. In the $M = 1$ scenario, we get a plateau around the PU. Thus, packets avoid getting onto the plateau and close to the PU, but once on it, continue towards the destination instead of moving away from the PU as in the VFF. When $M = 2$, we get a ridge similar to the one in VFF, but much smaller. For $M = 3$, there is no longer any ridge and the potential field is instead starting to resemble the GPF. This suggests that the usefulness of computing the more complex SSDOPF as opposed to GPF may diminish as M grows.



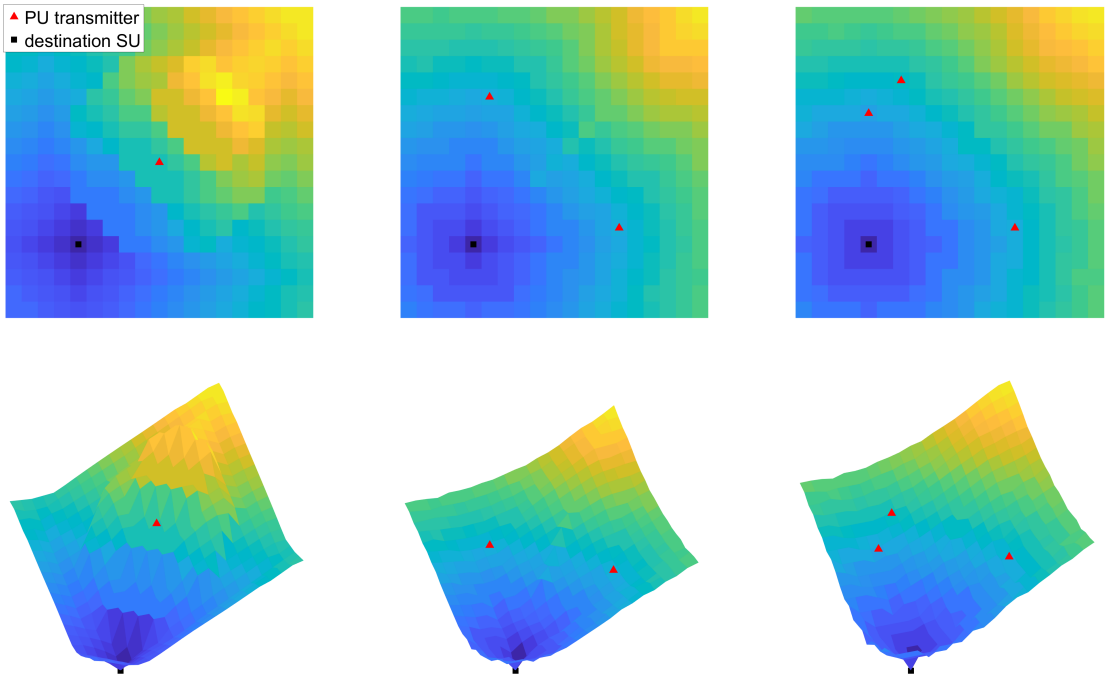
(a) $M = 1$. (b) $M = 2$. (c) $M = 3$.

Figure 14: VFF. The potential increases when moving closer to *all* PUs (red triangles), and decreases when approaching the destination (black square). Note that the potential grows exponentially when approaching the PU in (a) but that this peak has been omitted (cut off) in the picture.

6.2 Results of the Scheduling

Recall that the long-term goal of the scheduling is to minimize the end-to-end delays of packets. To measure the performance of the proposed algorithms, we therefore run network simulations where the proposed algorithms are applied over a large number of consecutive time slots, and the resulting end-to-end delays of packets over the course of the simulation recorded. Specifically, each network simulation is run using the following parameters and assumptions:

- The network consists of $N = 200$ SUs, located uniformly at random inside a square with side $x = 200$.
- The number M of frequency bands (and thus, PU transmitters) varies from 1 to 10. The PU transmitters are located uniformly at random within the same square as the SUs.
- Each PU transmitter has a corresponding receiver located uniformly at random inside its coverage region. Recall that these locations are not known to the SUs and hence they are only used to measure the violation rate of constraint (13).



(a) $M = 1$. (b) $M = 2$. (c) $M = 3$.

Figure 15: SSDOPF. The SSDOPF is similar to the VFF, but with less penalty for staying near PUs (red triangles) and more focus on reaching the destination (black square).

- The simulations are run using one of two different network topology settings: a *static* setting where SUs remain stationary throughout the simulation, and a *dynamic* setting where SUs move around and may connect/disconnect from the network at random. If not specified, the static setting is the one used.
- The SINR threshold is set to $\Gamma = 10$ (note that we use a linear scale here as per our definition (2) of the SINR).
- The path loss exponent is set to $\alpha = 4$ and the noise power at both SUs and PUs to $\sigma_{\text{SU}} = \sigma_{\text{PU}} = 10^{-4}$.
- SUs and PUs are given a coverage distance $d_{\text{C}} = x/4$, which defines the maximum transmit power of SUs according to $P_{\max} = \frac{\sigma_{\text{SU}}\Gamma}{(1+d_C)^{-\alpha}}$, and the actual transmit power used by the PU transmitters according to $P^m_{\texttt{TX}} = \frac{\sigma_{\text{PU}}\Gamma}{(1+d_{\text{C}})^{-\alpha}}$.
- The power control parameters are set to $\tau = 0.75$ and $\epsilon = 0.05$ unless stated otherwise.
- GPF is used as the potential field unless stated otherwise.
- The positive scaling factors in VFF are set to $\nu = 1$ and $\xi = 0.4 \cdot 10^5$.

- The exponent used to (de-)emphasize the priorities Q in the weight function is set to $\lambda = 0.4$.
- In the distributed scheduling, the number of rounds is set to $S = 1$.
- The simulations are run for a total of 100 time slots t, and contain 100 SU sessions z (= a data packet to be sent from a source SU to a destination SU) whose start time $t_z^{\text{start}} \in [1, 100]$, source SU u_z, and destination SU v_z are chosen uniformly at random. A session z is active (i.e. considered in the scheduling) during time slots $t \in [t_z^{\text{start}}, t_z^{\text{finish}}]$, where t_z^{finish} is the time slot during which the packet reached its destination v_z in the simulation. The end-to-end delay, henceforth simply *delay*, of a packet is defined to be the length, or number of time slots, of their active period, i.e. $\text{delay}(z) = (t_z^{\text{finish}} + 1) - t_z^{\text{start}}$.
- In each time slot, the SUs transmit according to the scheduling methods described in the previous chapters. Only transmissions with an SINR equal to or greater than the threshold are considered successful, while all other transmissions fail.

Note that any nondeterministic parameter values (e.g. the locations of SUs) are determined independently for each simulation run. After each simulation run, the following quantitative performance criteria are recorded:

1. The *mean* delay of all packets over the course of the simulation.
2. The *maximum* delay of any packet over the course of the simulation.
3. The violation rate of constraint (13), i.e. the number of failed PU transmissions divided by the total number of PU transmissions. We will refer to this as the *primary collision rate*.

In the following subsections we provide comparisons between different scheduling methods and study the impact of different parameter values for each scheduling approach. The comparisons are made via the resulting averages (over a total of 1000 simulation runs) of one or more of the above three quantities.

6.2.1 Overall Performance

In our first comparison, we will see how the proposed scheduling methods fare against a more simple random access based protocol. For the centralized scheduling method, two different values of the power control parameter τ are considered: the default $\tau = 0.75$, as well as $\tau = 0$. Recall that τ is used to adjust the transmit power between its lower and upper bounds via Eq. (48), and that the case $\tau = 0$ ($P_i^m = P_{i,\text{L}}^m$) corresponds to the FDS algorithm in [5], where SUs are prohibited from using the same frequency bands regardless of SINRs involved. This is similar to the conflict resolution in the pruning phase of the distributed scheduling. Hence, the centralized scheduling with $\tau = 0$ is more closely related to the distributed scheduling method, providing a more fair comparison between the centralized and distributed methods.

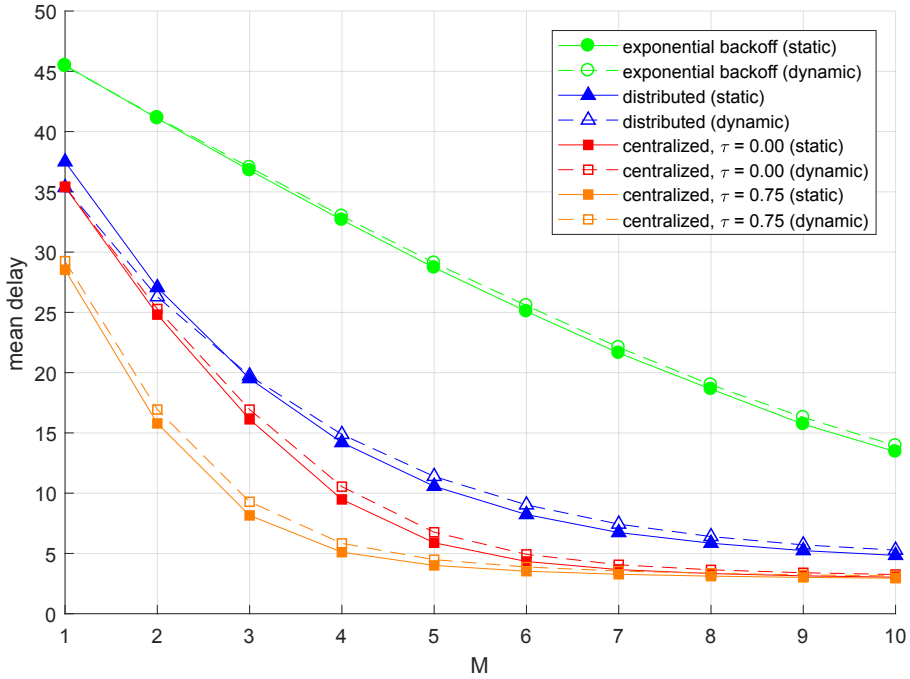
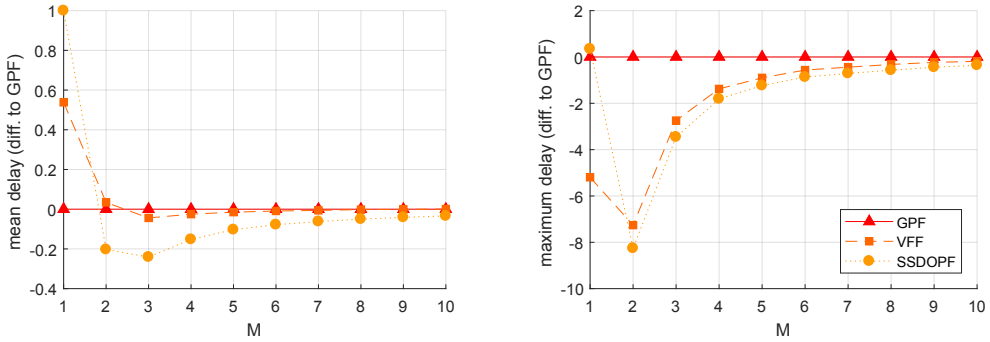


Figure 16: The mean delay of packets achieved by different scheduling methods, where the terms *static* and *dynamic* in brackets refer to the network topology. The proposed scheduling methods, both centralized and distributed, perform significantly better than the exponential backoff based random access protocol. Furthermore, their performances degrade only slightly in dynamic networks.

In the random access protocol used as a benchmark, SUs resolve conflicts by means of exponential backoff. Candidate transmitters will attempt to transmit in every time slot, but if the packet is not successfully transmitted, pick a random backoff time before they attempt to transmit again. Specifically, if a candidate transmitter has failed c consecutive transmissions, it will pick the next random wait time between 1 and $2^c - 1$ time slots uniformly at random. Recall that the scheduling problem in this case also includes finding approriate receivers, frequency bands and transmit powers of transmissions, which is not addressed by this random access protocol. Therefore, we also include the exchange phase and request phase (excluding the request messages) of the distributed scheduling method in the random access protocol to allow transmitters to make such decisions.

In addition to comparing different scheduling methods, the performances are studied in both static and dynamic network topologies. In the dynamic case, the radio environment is non-stationary and costly optimization makes less sense unless the dynamic behavior can be predicted reliably. The proposed methods involve little to no optimization, and hence we could justify this by showing that their performance degrades only minimally in dynamic network topologies.

The mean delays achieved by the different scheduling methods can be seen in Fig. 16. As the number M of frequency bands increases, the amount of available spectrum does too. Because the number of SU sessions, i.e. the traffic demand,



(a) (b)

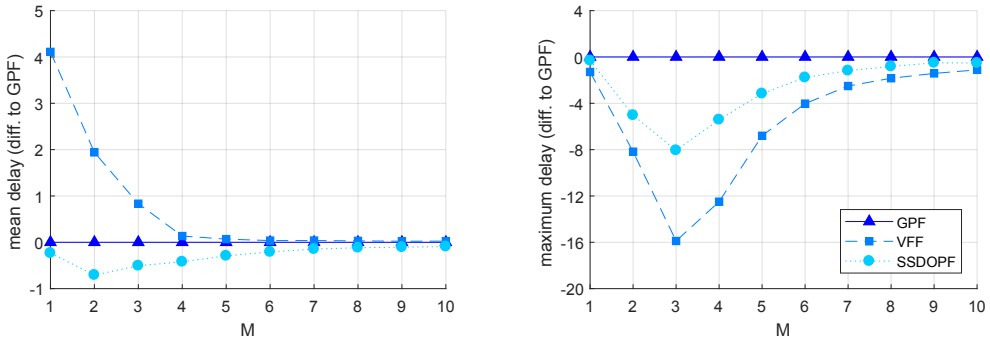
Figure 17: The difference in (a) mean and (b) maximum delay of packets achieved by different potential fields in comparison to the GPF when using centralized scheduling. The SSDOPF performs the best overall, with VFF mirroring the GPF in terms of mean delay, and the SSDOPF in terms of maximum delay. Both SSDOPF and VFF suffer relative to the GPF when $M = 1$.

remains constant, we therefore see that the performance of each method improves as M grows. The improvement continues until the traffic demands are exceeded by the available spectrum supplied, at which point additional spectrum no longer improves the performance. This saturation point is reached in the centralized scheduling around $M = 5$ for $\tau = 0.75$ and $M = 7$ for $\tau = 0$, while the distributed scheduling and random access protocol are yet to reach saturation by $M = 10$. Although the distributed scheduling is not quite able to match the performance of the centralized algorithm, even when $\tau = 0$, there are still significant gains compared to the random access protocol.

As expected, the loss in performance due to a dynamic network topology remains small for both the centralized and distributed scheduling methods. Curiously, the performance even improves for $M \leq 2$ in the latter case. Because the dynamicity in this case increases the randomness involved, this suggests that the distributed scheduling may benefit from randomizing more components of its protocol (in this case only the choice of channel is nondeterministic).

6.2.2 The Choice of Potential Field

In this subsection we will study how the different potential fields affect the performance of the scheduling. Here we consider both mean and maximum delays of packets. The former corresponds to the overall throughput of the network, while the latter shows how fair the scheduling is. The results this time are shown as differences compared to the GPF. That is, instead of the actual delays achieved by using each potential field, we show the *difference* in delay (mean or maximum) when using the same scheduling algorithm (centralized or distributed) but different potential fields (GPF, VFF, or SSDOPF, where GPF is the default one being compared to). Because the goal is to minimize delays, negative values mean better performance than the GPF and positive values worse performance.



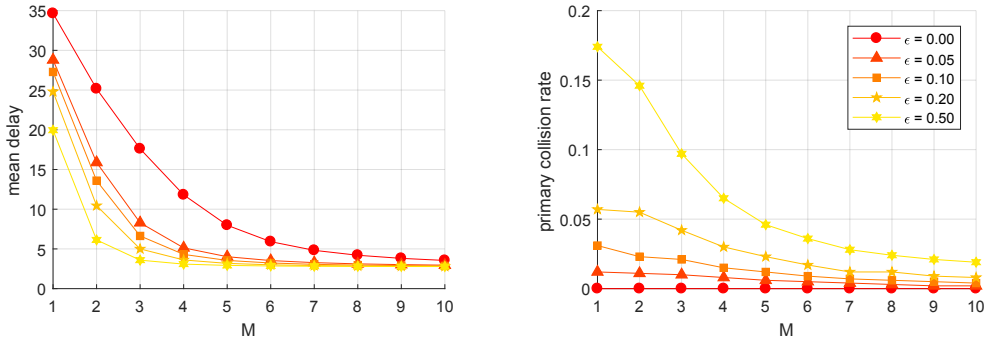
(a) (b)

Figure 18: The difference in (a) mean and (b) maximum delay of packets achieved by different potential fields in comparison to the GPF when using distributed scheduling. The SSDOPF performs better than the GPF both in terms of mean and maximum delay. The VFF performs worse than the GPF in terms of mean delay, but also the overall best in terms of maximum delay.

The results for the centralized scheduling can be seen in Fig. 17. First, let us take a look at the mean delay. Ignoring for a moment the case $M = 1$, we see that the SSDOPF predictably performs the best, whereas the VFF and GPF have almost equal performance with each other. When $M = 1$, both SSDOPF and VFF are noticeably worse than the GPF. There could be a number of reasons for this. Recall from Fig. 16 that the mean delays are the highest when $M = 1$, and thus the traffic congestion level, i.e. the number of concurrent SU sessions, is also at its highest. Because the VFF and SSDOPF only take PU interference into consideration (and not interference caused by SUs), they become less accurate when the amount of concurrent SU sessions increases. At $M = 1$, it appears that the number of concurrent SU sessions is so high that most interference will come from other SUs and not from the PUs, rendering the VFF and SSDOPF less useful.

Moving on to the maximum delay, we see that the SSDOPF again results in the overall best performance. The VFF is now much closer to the SSDOPF, with their performance curves following a very similar pattern. As with the mean delay, the performance of both the VFF and SSDOPF degrades when $M = 1$. This time, however, VFF remains better than GPF despite a relative decrease in performance.

In the distributed scheduling, the results of which are shown in Fig. 18, we notice a few differences compared to the centralized scheduling. The most noticeable difference comes from the VFF. It is now *worse* than the GPF in terms of mean delay, but also the overall best choice in terms of maximum delay. Although the SSDOPF appears to perform similarly as in the centralized case, there are two key differences. First, it now outperforms the GPF for *all* M, including $M = 1$. Second, the absolute differences in mean/maximum delay between it and the GPF are roughly twice as large as before, meaning its advantage over the GPF is even larger in the distributed scheduling.



(a) (b)

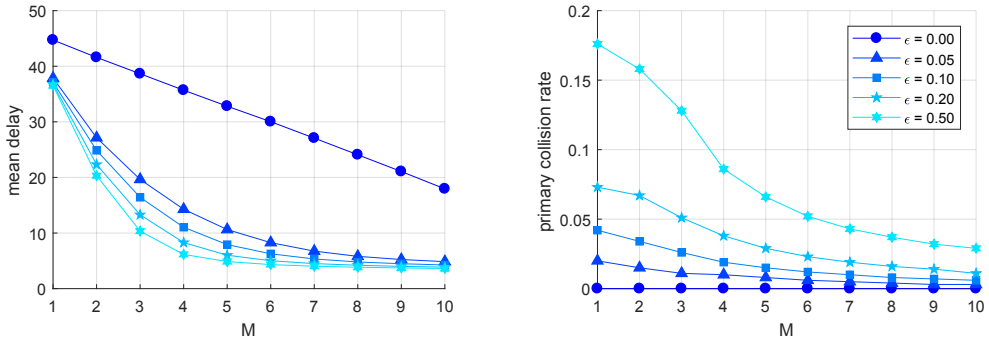
Figure 19: The (a) mean delay of packets and (b) primary collision rate when using different values of ϵ in the centralized scheduling. The performance in terms of mean delay improves noticeably when going from $\epsilon = 0$ to $\epsilon > 0$, with smaller improvements after that. The primary collision rate increases more or less linearly with increases in the value of ϵ.

6.2.3 Adjusting the Value of ϵ

Recall that the parameter ϵ is used when upper bounding the transmit powers of SUs according to (43). In particular, it controls at what distance $(1-\epsilon)d_{\mathrm{C}}^{m}$ the PU receiver is assumed to be from the PU transmitter in the worst case. If violation of constraint (13) (the PU SINR constraint) is strictly prohibited, we must set $\epsilon = 0$. However, in doing so, the transmit power upper bound is set to 0 in any frequency band m where the PU transmitter is transmitting. Thus, SUs will be unable to transmit in bands m with ongoing PU transmissions regardless of how far away from the PU transmitter they are. If we want a true underlay approach to spectrum utilization, we should therefore set $\epsilon > 0$. The higher the value of ϵ is, the less SUs need to upper bound their transmit powers and consequently we would expect the delays of packets to decrease but also a higher violation rate of constraint (13). In what follows, we will study this tradeoff between packet delays and the violation rate of constraint (13), referred to as the primary collision rate. Five different values of ϵ are tested: 0, 0.05, 0.1, 0.2 and 0.5.

Fig. 19 shows the results for the centralized scheduling. In terms of the mean delay, we notice that there is a noticeable gap between $\epsilon = 0$ and the four curves for $\epsilon > 0$. Thus, there is a clear advantage to gain by setting $\epsilon > 0$ however small the chosen value of ϵ may be. Meanwhile, the primary collision rate seems to increase linearly with the value of ϵ: the primary collision rate when setting $\epsilon = a\epsilon_0$ appears to be more or less a times as large as for ϵ_0 independently of the value of M. This makes it easy to adjust the value of ϵ if we are given some violation probability threshold. For instance, if the primary collision rate for a certain value of ϵ is twice as large as the maximum allowed value, we simply half the value of ϵ to get a violation probability that does not exceed the threshold while minimizing the delays of packets.

The same observations can be made for the distributed scheduling shown in Fig. 20. However, in this case the gap in terms of mean delay between $\epsilon = 0$ and $\epsilon > 0$



(a) (b)

Figure 20: The (a) mean delay of packets and (b) primary collision rate when using different values of ϵ in the distributed scheduling. The performance in terms of mean delay improves noticeably when going from $\epsilon = 0$ to $\epsilon > 0$, with smaller improvements after that. The primary collision rate increases more or less linearly with increases in the value of ϵ.

is even larger. In fact, comparing the curve of $\epsilon = 0$ to that of the random access protocol in Fig. 16, we notice that the two are quite similar. We then once again conclude that setting $\epsilon > 0$ is crucial if violation of constraint (13) is not strictly prohibited.

6.3 Conclusion

We conclude this chapter by summarizing the key observations from the visual analysis of the potential fields as well as the simulation studies:

- The VFF and SSDOPF have similar characteristics, but with the former seemingly more focused on avoiding PU interference as opposed to reaching the destination. As a result, the VFF provides more fairness in the scheduling, while the SSDOPF provides the better overall throughput.
- The proposed scheduling methods appear to work well in both static and dynamic networks.
- The VFF and SSDOPF work best when the amount of SU traffic is low compared to the amount of PU traffic.
- Setting the power control parameter ϵ to any non-zero positive value will significantly improve the performance of the proposed methods, especially in the distributed case.

7 Summary

In this thesis we have studied the problem of scheduling in multi-hop multi-band cognitive radio networks. Specifically, the problem consisted of both the routing of data packets as well as the allocation of available spectrum resources among the cognitive users, with the joint goal of minimizing packet delays in the network. To emphasize these two different aspects of the scheduling, we formulated the problem as separate long- and short-term problems. The long-term problem focused on the long-term routing of packets, while the short-term problem expressed the allocation of spectrum resources in the short term in the form of an **NP**-hard MINLP optimization problem.

Due to the time-varying nature of the spectrum opportunities, the key issue in the long-term problem was to find routes that did not rely on any particular network link being available at a particular time. To accomodate this, we proposed a novel potential field based modeling technique [5] (extended from work in [4]) that allowed us to abstract the routes away from specific paths in the network onto an underlying potential field. This way the particular links to be used at any given time could be chosen opportunistically based on the current spectrum availability by consulting the potential fields. Similarly, in the short-term problem, the time-varying nature of the spectrum opportunities encouraged us to develop algorithms of low complexity. This would ensure that the spectrum opportunities that had been identified at a particular time could be exploited before they were no longer available. Therefore, we developed a polynomial-time centralized greedy algorithm [5] that approximately solved the short-term problem, as well as a distributed version of that algorithm [6] that could be used in ad hoc networks. Neither method involved heavy optimization, making them particularly useful in networks with a high amount of unknowns involved.

Our experiments with the proposed solutions demonstrated a number of significant results. Firstly, the different potential fields used to solve the long-term problem all had differents strengths and weaknesses, with the SSDOPF favoring overall throughput and the VFF fairness. Furthermore, The VFF and SSDOPF worked best when the amount of SU data traffic, and consequently required amount of SU activity, was low relative to the amount of PU activity. We also confirmed that both centralized and distributed scheduling methods adapted well to more dynamic networks. Finally, the performance of the proposed methods could be improved significantly if we allowed for a small non-zero probability of causing PU transmissions to fail.

For future work, a number of questions yet remain unanswered in regards to the proposed solutions. Most significantly, due to the system model used in this thesis being somewhat unique, we have been unable to find good benchmarks to compare our solutions to. The random access protocol used as benchmark in our simulations was not necessarily suited to the task at hand and had to be augmented with parts from the proposed scheduling protocol, so the fact that the proposed solutions were able to outperform it may not be as significant as the simulations suggest. Furthermore, the short-term problem was presented as a MINLP, but this MINLP was never solved optimally. Thus, solving the MINLP optimally in addition

to solving it approximately would provide a possible benchmark for our algorithms. Finally, the proposed scheduling methods were designed for networks where the future is completely unknown. If this was not the case, and we had e.g. packet streams with fixed sources and destinations instead of the individual packets with random source and destination as in our simulations, could the proposed methods be extended to take advantage of this knowledge?

References

[1] T. S. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. N. Wong, J. K. Schulz, M. Samimi, and F. Gutierrez, "Millimeter wave mobile communications for 5g cellular: It will work!" *IEEE Access*, vol. 1, pp. 335–349, May 2013.

[2] FCC Spectrum Policy Task Force, "Report of the spectrum efficiency working group," November 2002.

[3] J. Mitola, "Cognitive radio: An integrated agent architecture for software defined radio," Ph.D. dissertation, KTH, Stockholm, Sweden, 2000.

[4] J. Oksanen, B. Kaufman, V. Koivunen, and H. V. Poor, "Robotics inspired opportunistic routing for cognitive radio using potential fields," *IEEE Trans. Cogn. Commun. Netw.*, vol. 1, no. 1, 2015.

[5] H. Hentila, J. Oksanen, and V. Koivunen, "Potential field based scheduling in cognitive radio networks," in *IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, July 2016.

[6] ——, "Distributed scheduling in multi-hop multi-band cognitive radio networks utilizing potential fields," in *IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, July 2017.

[7] Q. Zhao and B. M. Sadler, "A Survey of Dynamic Spectrum Access," *IEEE Signal Process. Mag.*, vol. 24, no. 3, pp. 79–89, May 2007.

[8] A. Pantelidou and A. Ephremides, "Scheduling in wireless networks," *Foundations and Trends® in Networking*, vol. 4, no. 4, pp. 421–511, April 2011.

[9] T. Yücek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 1, pp. 116–130, 2009.

[10] D. D. Ariananda, M. K. Lakshmanan, and H. Nikookar, "A survey on spectrum sensing techniques for cognitive radio," in *International Workshop on Cognitive Radio and Advanced Spectrum Management (CogART)*, May 2009, pp. 74–79.

[11] I. Cidon and M. Sidi, "Distributed assignment algorithms for multihop packet radio networks," *IEEE Transactions on Computers*, vol. 38, no. 10, pp. 1353–1361, October 1989.

[12] A. Ephremides and T. V. Truong, "Scheduling broadcasts in multihop radio networks," *IEEE Transactions on Communications*, vol. 38, no. 4, pp. 456–460, April 1990.

[13] M. Li, S. Salinas, P. Li, X. Huang, Y. Fang, and S. Glisic, "Optimal scheduling for multi-radio multi-channel multi-hop cognitive cellular networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 1, pp. 139–154, Jan 2015.

[14] D. Xue and E. Ekici, "Efficient distributed scheduling in cognitive radio networks in the many-channel regime," in *Modeling & Optimization in Mobile, Ad Hoc & Wireless Networks (WiOpt)*, May 2013, pp. 179–185.

[15] N. Zhang, W.-C. Liao, M. Hong, H. Farmanbar, and Z.-Q. Luo, "Traffic engineering for backhaul networks with wireless link," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[16] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, March 2000.

[17] M. H. Cheung, V. W. S. Wong, and R. Schober, "Sinr-based random access for cognitive radio: Distributed algorithm and coalitional game," *IEEE Transactions on Wireless Communications*, vol. 10, no. 11, pp. 3887–3897, November 2011.

[18] E. Bedeer, O. Amin, O. A. Dobre, M. H. Ahmed, and K. E. Baddour, "Energy-efficient power loading for ofdm-based cognitive radio systems with channel uncertainties," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 6, pp. 2672–2677, June 2015.

[19] Y. Shi, Y. T. Hou, S. Kompella, and H. D. Sherali, "Maximizing capacity in multihop cognitive radio networks under the sinr model," *IEEE Transactions on Mobile Computing*, vol. 10, no. 7, pp. 954–967, July 2011.

[20] G. I. Tsiropoulos, O. A. Dobre, M. H. Ahmed, and K. E. Baddour, "Radio resource allocation techniques for efficient spectrum access in cognitive radio networks," *IEEE Communication Surveys & Tutorials*, vol. 18, no. 1, pp. 824–847, 2016.

[21] M. Pan, C. Zhang, P. Li, and Y. Fang, "Joint routing and link scheduling for cognitive radio networks under uncertain spectrum supply," in *Proceedings of IEEE INFOCOM*, April 2011, pp. 2237–2245.

[22] Y. Le, X. Cheng, D. Chen, N. Zhang, T. Znati, M. A. Al-Rodhaan, and A. Al-Dhelaan, "Distributed back-pressure scheduling with opportunistic routing in cognitive radio networks," *EURASIP J. Wirel. Commun. Netw.*, vol. 2015, no. 1, pp. 1–14, 2015.

[23] D. Gözüpek and F. Alagöz, "Throughput and delay optimal scheduling in cognitive radio networks under interference temperature constraints," *Journal of Communications and Networks*, vol. 11, no. 2, pp. 147–155, Apr 2009.

[24] A. Goldsmith, S. A. Jafar, I. Maric, and S. Srinivasa, "Breaking spectrum gridlock with cognitive radios: An information theoretic perspective," *Proceedings of the IEEE*, vol. 97, no. 5, pp. 894–914, April 2009.

[25] R. Urgaonkar and M. J. Neely, "Opportunistic scheduling with reliability guarantees in cognitive radio networks," *IEEE Trans. Mobile Comput.*, vol. 8, no. 6, pp. 766–777, 2009.

[26] H. Zheng and L. Cao, "Device-centric spectrum management," in *IEEE International Symposium on Dynamic Spcetrum Access Networks (DySPAN)*, November 2005, pp. 56–65.

[27] D. I. Kim, L. B. Le, and E. Hossain, "Joint rate and power allocation for cognitive radios in dynamic spectrum access environment," *IEEE Transactions on Wireless Communications*, vol. 7, no. 12, pp. 5517–5527, December 2008.

[28] L. Akter and B. Natarajan, "Distributed approach for power and rate allocation to secondary users in cognitive radio networks," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 4, pp. 1526–1538, May 2011.

[29] C. Gao, S. Chu, and X. Wang, "Distributed scheduling in mimo empowered cognitive radio ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 7, pp. 1456–1468, July 2014.

[30] P. Cheng, Z. Zhang, H.-H. Chen, and P. Qiu, "Optimal distributed joint frequency, rate and power allocation in cognitive OFDMA systems," *IET Communications*, vol. 2, no. 6, pp. 815–826, Jul 2008.

[31] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936–1948, 1992.